

	Espacio reservado para el adhesivo de radicación
---	--

DIRECCIÓN DE NUEVAS CREACIONES
Procedimiento acelerado de exámenes de patente (PPH)

1	NÚMERO DE SOLICITUD ANTE LA SIC NC2017/0003311		
2	GLOBAL - PPH ☒ PAÍS AUSTRALIA	3	PCT – PPH ☐ PAÍS
4	PROSUR ☐ PAÍS	5	Alianza del Pacífico (AdP) ☐ PAÍS
6	OFICINA DE PRIMERA PRESENTACIÓN (OPP) ☐	7	OFICINA DE EXAMEN ANTERIOR (OEA) ☐
8	NÚMERO DE SOLICITUD O PUBLICACIÓN DE LA SOLICITUD (OPP) ☐ NÚMERO DE SOLICITUD EN OFICINA DE EXAMEN ANTERIOR (OEA): ☐ NÚMERO DE SOLICITUD PCT: ☒		1. PCT/SG2015/050317
9	☐ REPRESENTANTE LEGAL ☒ APODERADO		
APELLIDOS DAZA MONTALVO		NOMBRES CAROLINA MERCEDES	IDENTIFICACIÓN C.C. 66.783.790 T.P. 141563 CSJ
DIRECCIÓN Carrera 11 No. 86-53 Piso 3		No. TELÉFONO + 57 1 618 1088	
CIUDAD Bogotá		CORREO ELECTRÓNICO notificaciones@clarkem odet.com.co	
PAÍS COLOMBIA		No. RADICACIÓN O PROTOCOLO DE PODER GENERAL	
10	CONDICIONES PARA ACCEDER A ESTE PROCESO:		
10.1. DATOS DE PRIORIDAD ☒ SI ☐ NO			
(33) PAÍS DE ORIGEN 1. Estados Unidos de América		CÓDIGO PAÍS US	(31) NÚMERO 62/050,414 (32) FECHA (AAAA/MM/DD) 2014/09/15
10.2. DATOS SOLICITUD INTERNACIONAL PCT ☐ SI ☐ NO			
Solicitud Internacional No. PCT/SG2015/050317		Fecha	15/09/2015
Publicación Internacional No. WO/2016/043659		Fecha	24/03/2016
*El solicitante presenta una copia de la(s) reivindicación(es) patentable(s) u otorgable(s) de la solicitud de la Oficina de Primera Presentación que profirió el examen de patentabilidad, junto con una traducción al castellano. ☒ SI ☐ NO *El solicitante presenta los resultados de búsqueda de la Oficina de Examen Anterior, junto con una traducción al castellano. ☒ SI ☐ NO			
REQUISITOS PARA EL RECONOCIMIENTO DE LOS RESULTADOS DE EXÁMENES DE PATENTABILIDAD EFECTUADOS A SOLICITUDES EXTRANJERAS Para que la Superintendencia de Industria y Comercio pueda reconocer los resultados de exámenes de patentabilidad producidos por otras Oficinas de Propiedad Industrial se debe cumplir con las siguientes condiciones: (a) Al momento de solicitar el examen de patentabilidad, si el solicitante pide que se reconozcan exámenes de patentabilidad de otras oficinas extranjeras, deberá aportar junto con el pago de la tasa correspondiente, los siguientes anexos: (i) Copia de las resoluciones, fallos o decisiones que han sido relevantes para la determinación de la patentabilidad de la(s) solicitud(es) en la Oficina que haya proferido el examen de patentabilidad que contienen reivindicaciones patentables u otorgables que son la base de la petición, y si es del caso, su traducción al castellano. ☒ SI ☐ NO (ii) Copia de todas las reivindicaciones determinadas como patentables u otorgables en la Oficina que profirió el examen de patentabilidad y, si es del caso, su traducción al castellano. ☒ SI ☐ NO (iii) Copia de los antecedentes, información o documentos citados por el examinador de la Oficina de Primera Presentación del examen de patentabilidad, en las resoluciones, fallos o decisiones correspondientes, incluyendo la literatura no patente. Los documentos de patente deberán presentarse siempre y cuando no estén disponibles para la Superintendencia de Industria y Comercio mediante el simple acceso a bases de datos gratuitas. ☒ SI ☐ NO (iv) Una tabla de correspondencias de las reivindicaciones que han sido determinadas como patentables u otorgables en la solicitud de la Oficina que haya proferido el examen de patentabilidad. ☒ SI ☐ NO			
11	FIRMA DEL SOLICITANTE, DEL APODERADO O DEL REPRESENTANTE LEGAL Junto a cada firma, indicar el nombre del firmante y su calidad (si tal calidad no es obvia al leer el petitorio)		
 CAROLINA DAZA MONTALVO			

10 (iv)	TABLA DE CORRESPONDENCIA DE REIVINDICACIONES		
Reivindicaciones en la solicitud en Colombia (Modificaciones Voluntarias)		Reivindicaciones otorgables en la solicitud internacional No. PCT/SG2015/050317	Explicación sobre la suficiente correspondencia ó Comentarios que explican la correspondencia
1. Un método implementado por computadora para clasificar una imagen digital, el método comprende: obtener, de una computadora principal, datos de rasgos que corresponden a la imagen digital; determinar, mediante una unidad de procesamiento gráfico, una distancia semimétrica en función de una distribución binomial de Poisson entre los datos de rasgos y uno o más datos de rasgos de referencia almacenados en una memoria de la computadora principal; y clasificar la imagen digital usando la distancia semimétrica determinada.		1. Un método implementado por computadora para clasificar una imagen digital, el método comprende: obtener, de una computadora principal, datos de rasgos que corresponden a la imagen digital; determinar, mediante una unidad de procesamiento gráfico, una distancia semimétrica en función de una distribución binomial de Poisson entre los datos de rasgos y uno o más datos de rasgos de referencia almacenados en una memoria de la computadora principal; y clasificar la imagen digital usando la distancia semimétrica determinada.	<i>Igual</i>
2. El método de la reivindicación 1, donde la distancia semimétrica es un radio binominal de Poisson (PBR).		2. El método de la reivindicación 1, donde la distancia semimétrica es un radio binominal de Poisson (PBR).	<i>Igual</i>

3. El método de la reivindicación 1, donde clasificar la imagen digital comprende usar un clasificador de máquinas de vectores de soporte (SVM).	3. El método de la reivindicación 1, donde clasificar la imagen digital comprende usar un clasificador de máquinas de vectores de soporte (SVM).	<i>Igual</i>
4. El método de la reivindicación 1, donde clasificar la imagen digital comprende usar un clasificador de k vecino más cercano (kNN).	4. El método de la reivindicación 1, donde clasificar la imagen digital comprende usar un clasificador de k vecino más cercano (kNN).	<i>Igual</i>
5. El método de la reivindicación 4, donde el clasificador kNN es un clasificador de k vecinos más cercanos basado en el promedio local adaptativo (ALMkNN), donde el valor de los k vecinos más cercanos (k) se elige de manera adaptativa.	5. El método de la reivindicación 1, donde el clasificador kNN es un clasificador de k vecinos más cercanos basado en el promedio local adaptativo (ALMkNN), donde el valor de los k vecinos más cercanos (k) se elige de manera adaptativa.	<i>Similares, se cambió la dependencia para ahora depender de la reivindicación 4, y así poder darle claridad a la invención.</i>
6. El método de la reivindicación 5, donde el valor adaptativo de los k vecinos más cercanos no excede la raíz cuadrada del número de uno o más datos de características de referencia.	6. El método de la reivindicación 5, donde el valor adaptativo de los k vecinos más cercanos no excede la raíz cuadrada del número de uno o más datos de referencia.	<i>Similares, se cambió la expresión “uno o más datos de referencia” por “uno o más datos de características de referencia” para dar una mayor claridad a la invención.</i>
7. El método de la reivindicación 1, donde los datos de rasgo obtenidos y uno o más datos de rasgos de	7. El método de la reivindicación 1, donde los datos de rasgo obtenidos y uno o más datos de rasgos de	<i>Igual</i>

referencia comprenden datos de patrón binario local de concurrencia de invariancia rotacional en pares (PRICoLBP).	referencia comprenden datos de patrón binario local de concurrencia de invariancia rotacional en pares (PRICoLBP).	
8. El método de la reivindicación 1, donde los datos de rasgo obtenidos y uno o más datos de rasgos de referencia comprenden datos de histograma de gradientes orientados (HOG).	8. El método de la reivindicación 1, donde los datos de rasgo obtenidos y uno o más datos de rasgos de referencia comprenden datos de histograma de gradientes orientados (HOG).	<i>Igual</i>
9. El método de la reivindicación 1, donde los datos de rasgo obtenidos comprenden un vector de rasgo X de dimensión N tal que $X = (a_1 \dots a_N)$ y los datos de rasgo de referencia comprenden un vector de rasgo Y de dimensión N tal que $Y = (b_1 \dots b_N)$, y donde determinar la distancia semimétrica (PBR(X,Y)) comprende calcular:	9. El método de la reivindicación 1, donde los datos de rasgo obtenidos comprenden un vector de rasgo X de dimensión N tal que $X = (a_1 \dots a_N)$ y los datos de rasgo de referencia comprenden un vector de rasgo Y de dimensión N tal que $Y = (b_1 \dots b_N)$, y donde determinar la distancia semimétrica (PBR(X,Y)) comprende calcular:	<i>Igual</i>
$PBR(X,Y) = \frac{\sigma}{N - \mu} = \frac{\sqrt{\sum_{i=1}^N p_i(1 - p_i)}}{N - \sum_{i=1}^N p_i}$	$PBR(X,Y) = \frac{\sigma}{N - \mu} = \frac{\sqrt{\sum_{i=1}^N p_i(1 - p_i)}}{N - \sum_{i=1}^N p_i}$	

donde N es un entero mayor que 0, σ es la desviación estándar del vector X, μ es el promedio del vector X, y p_i es $a_i - b_i$.	donde N es un entero mayor que 0, σ es la desviación estándar del vector X, μ es el promedio del vector X, y p_i es $a_i - b_i$.	
10. El método de la reivindicación 1, donde la imagen digital comprende información que corresponde a una secuencia de ADN o ARN y los datos de rasgo obtenidos comprenden un vector X de proximidades de calidad de secuenciación para una primera muestra de ADN con una profundidad de secuenciación d_x tal que $X = (x_1 \dots x_{dx})$ y los datos de rasgo de referencia comprenden un vector Y de probabilidades de secuenciación para una muestra de ADN de referencia con una profundidad de secuenciación d_y tal que $Y = (y_1 \dots y_{dy})$, y donde determinar la distancia semimétrica (PBR_{seq}) comprende calcular:	10. El método de la reivindicación 1, donde la imagen digital comprende información que corresponde a una secuencia de ADN o ARN y los datos de rasgo obtenidos comprenden un vector X de proximidades de calidad de secuenciación para una primera muestra de ADN con una profundidad de secuenciación d_x tal que $X = (x_1 \dots x_{dx})$ y los datos de rasgo de referencia comprenden un vector Y de probabilidades de secuenciación para una muestra de ADN de referencia con una profundidad de secuenciación d_y tal que $Y = (y_1 \dots y_{dy})$, y donde determinar la distancia semimétrica (PBR_{seq}) comprende calcular:	<i>Igual</i>

$PBR_{seq}(X, Y) = \frac{\sigma_X \sigma_Y}{ \mu_X - \mu_Y },$ donde μ_x es un promedio del vector X, μ_y es un promedio del vector Y, σ_x es una desviación estándar para el vector X, y σ_y es una desviación estándar para el vector Y.	$PBR_{seq}(X, Y) = \frac{\sigma_X \sigma_Y}{ \mu_X - \mu_Y },$ donde μ_x es un promedio del vector X, μ_y es un promedio del vector Y, σ_x es una desviación estándar para el vector X, y σ_y es una desviación estándar para el vector Y.	
11. El método de la reivindicación 10, donde la clasificación de la imagen digital comprende: determinar si la distancia semimétrica (PBR _{seq}) es mayor que un valor de umbral y clasificar la secuencia de ADN o ARN como tumor o normal en función de la determinación de si la distancia semimétrica (PBR _{seq}) es mayor que el valor de umbral.	11. El método de la reivindicación 10, donde la clasificación de la imagen digital comprende: determinar si la distancia semimétrica (PBR _{seq}) es mayor que un valor de umbral y clasificar la secuencia de ADN o ARN como tumor o normal en función de la determinación de si la distancia semimétrica (PBR _{seq}) es mayor que el valor de umbral.	<i>Igual</i>
12. El método de la reivindicación 10, donde clasificar la imagen digital comprende identificar una variante rara	12. El método de la reivindicación 10, donde clasificar la imagen digital comprende identificar una variante rara en	<i>Igual</i>

en la secuencia de ADN o ARN.	la secuencia de ADN o ARN.	
13. El método de la reivindicación 1, que comprende además: determinar un dato de rasgo de referencia de coincidencia más cercana de uno o más datos de rasgo de referencia.	13. El método de la reivindicación 1, que comprende además: determinar un dato de rasgo de referencia de coincidencia más cercana de uno o más datos de rasgo de referencia.	<i>Igual</i>
14. El método de la reivindicación 13, que comprende además: identificar a una persona en función del dato de rasgo de referencia de coincidencia más cercana determinado, donde la imagen digital comprende al menos uno de: una oreja, un rostro, una huella dactilar y un iris.	14. El método de la reivindicación 13, que comprende además: identificar a una persona en función del dato de rasgo de referencia de coincidencia más cercana determinado, donde la imagen digital comprende al menos uno de: una oreja, un rostro, una huella dactilar y un iris.	<i>Igual</i>
15. Un sistema para clasificar una imagen digital que comprende: una computadora principal que comprende un procesador, donde la computadora principal está	15. Un sistema para clasificar una imagen digital que comprende: una computadora principal que comprende un procesador, donde la computadora principal está acoplada a una	<i>Igual</i>

acoplada a una memoria que comprende uno o más datos de rasgo de referencia; y una unidad de procesamiento gráfico (GPU) que comprende un procesador, donde la GPU está acoplada a la computadora principal y está configurada para: obtener, de la computadora principal, datos de rasgo que corresponden a la imagen digital; acceder, desde la memoria, a uno o más datos de rasgo de referencia; determinar una distancia semimétrica en función de una distribución binomial de Poisson entre los datos de rasgo y uno o más datos de rasgo de referencia; donde la computadora principal está configurada para: clasificar la imagen digital usando la distancia semimétrica determinada.	memoria que comprende uno o más datos de rasgo de referencia; y una unidad de procesamiento gráfico (GPU) que comprende un procesador, donde la GPU está acoplada a la computadora principal y está configurada para: obtener, de la computadora principal, datos de rasgo que corresponden a la imagen digital; acceder, desde la memoria, a uno o más datos de rasgo de referencia; determinar una distancia semimétrica en función de una distribución binomial de Poisson entre los datos de rasgo y uno o más datos de rasgo de referencia; donde la computadora principal está configurada para: clasificar la imagen digital usando la distancia semimétrica determinada.	
---	--	--

16. El sistema de la reivindicación 15, donde la distancia semimétrica es un radio binomial de Poisson (PBR).	16. El sistema de la reivindicación 15, donde la distancia semimétrica es un radio binomial de Poisson (PBR).	<i>Igual</i>
17. El sistema de la reivindicación 15, donde la computadora principal está configurada además para clasificar la imagen digital usando un clasificador de máquinas de vectores de soporte (SVM).	17. El sistema de la reivindicación 15, donde la computadora principal está configurada además para clasificar la imagen digital usando un clasificador de máquinas de vectores de soporte (SVM).	<i>Igual</i>
18. El sistema de la reivindicación 15, donde la computadora principal está configurada además para clasificar la imagen digital usando un clasificador de k vecino más cercano (kNN).	18. El sistema de la reivindicación 15, donde la computadora principal está configurada además para clasificar la imagen digital usando un clasificador de k vecino más cercano (kNN).	<i>Igual</i>
19. El sistema de la reivindicación 18, donde el clasificador kNN es un clasificador de k vecinos más cercanos basado en el promedio local adaptativo (ALMkNN), donde el valor de los k vecinos más cercanos (k) se elige de manera adaptativa.	19. El sistema de la reivindicación 18, donde el clasificador kNN es un clasificador de k vecinos más cercanos basado en el promedio local adaptativo (ALMkNN), donde el valor de los k vecinos más cercanos (k) se elige de manera adaptativa.	<i>Igual</i>

20. El sistema de la reivindicación 19, donde el valor adaptativo de los k vecinos más cercanos (k) no excede la raíz cuadrada del número de uno o más datos de características de referencia.	20. El sistema de la reivindicación 19, donde el valor adaptativo de los k vecinos más cercanos (k) no excede la raíz cuadrada del número de uno o más datos de referencia.	<i>Similares, se cambió la expresión “uno o más datos de referencia” por “uno o más datos de características de referencia” para dar una mayor claridad a la invención.</i>
21. El sistema de la reivindicación 15, donde los datos de rasgo y uno o más datos de rasgos de referencia comprenden datos de patrón binario local de concurrencia de invariancia rotacional en pares (PRICoLBP).	21. El sistema de la reivindicación 15, donde los datos de rasgo y uno o más datos de rasgos de referencia comprenden datos de patrón binario local de concurrencia de invariancia rotacional en pares (PRICoLBP).	<i>Igual</i>
22. El sistema de la reivindicación 15, donde los datos de rasgo obtenidos y uno o más datos de rasgos de referencia comprenden datos de histograma de gradientes orientados (HOG).	22. El sistema de la reivindicación 15, donde los datos de rasgo obtenidos y uno o más datos de rasgos de referencia comprenden datos de histograma de gradientes orientados (HOG).	<i>Igual</i>
23. El sistema de la reivindicación 15, donde los datos de rasgo comprenden un vector de rasgo X de dimensión N tal que $X = (a_1 \dots a_N)$ y los datos de rasgo de referencia comprenden un vector de rasgo Y de dimensión N tal que	23. El sistema de la reivindicación 15, donde los datos de rasgo comprenden un vector de rasgo X de dimensión N tal que $X = (a_1 \dots a_N)$ y los datos de rasgo de referencia comprenden un vector de rasgo Y de dimensión N tal que	<i>Igual</i>

Y = (b₁...b_N), y donde la GPU también está configurada para calcular: $PBR(X,Y) = \frac{\sigma}{N - \mu} = \frac{\sqrt{\sum_{i=1}^N p_i(1 - p_i)}}{N - \sum_{i=1}^N p_i}$ donde PBR(X,Y) es una distancia de radio binomial de Poisson (PBR) entre el vector X y el vector Y, N es un entero mayor que 0, σ es la desviación estándar del vector X; μ es el promedio del vector X, y p_i es a_i- b_i .	Y = (b₁...b_N), y donde la GPU también está configurada para calcular: $PBR(X,Y) = \frac{\sigma}{N - \mu} = \frac{\sqrt{\sum_{i=1}^N p_i(1 - p_i)}}{N - \sum_{i=1}^N p_i}$ donde PBR(X,Y) es una distancia de radio binomial de Poisson (PBR) entre el vector X y el vector Y, N es un entero mayor que 0, σ es la desviación estándar del vector X; μ es el promedio del vector X, y p_i es a_i- b_i .	
24. El sistema de la reivindicación 15, donde la imagen digital comprende información que corresponde a una secuencia de ADN o ARN y los datos de rasgo comprenden un vector X de proximidades de calidad de secuenciación para una primera muestra de ADN con una	24. El sistema de la reivindicación 15, donde la imagen digital comprende información que corresponde a una secuencia de ADN o ARN y los datos de rasgo comprenden un vector X de proximidades de calidad de secuenciación para una primera muestra de ADN con una	<i>Similares, se cambió la expresión “donde la GPU también está configurada para calcular y determinar la distancia semimétrica (PBRseq) que comprende calcular:” por “donde la GPU también está configurada para calcular:” para dar una mayor claridad a la invención.</i>

profundidad de secuenciación d_x tal que $X = (x_1 \dots x_{d_x})$ y los datos de rasgo de referencia comprenden un vector Y de probabilidades de secuenciación para una muestra de ADN de referencia con una profundidad de secuenciación d_y tal que $Y = (y_1 \dots y_{d_y})$, y donde la GPU también está configurada para calcular: $PBR_{seq}(X, Y) = \frac{\sigma_X \sigma_Y}{ \mu_X - \mu_Y },$ donde $PBR_{seq}(X, Y)$ es una distancia de radio binomial de Poisson (PBR) entre el vector X y el vector Y, μ_x es un promedio del vector X, μ_y es un promedio del vector Y, σ_x es una desviación estándar para el vector X, y σ_y es una desviación estándar para el vector Y.	profundidad de secuenciación d_x tal que $X = (x_1 \dots x_{d_x})$ y los datos de rasgo de referencia comprenden un vector Y de probabilidades de secuenciación para una muestra de ADN de referencia con una profundidad de secuenciación d_y tal que $Y = (y_1 \dots y_{d_y})$, y donde la GPU también está configurada para calcular y determinar la distancia semimétrica (PBR_{seq}) que comprende calcular: $PBR_{seq}(X, Y) = \frac{\sigma_X \sigma_Y}{ \mu_X - \mu_Y },$ donde $PBR_{seq}(X, Y)$ es una distancia de radio binomial de Poisson (PBR) entre el vector X y el vector Y, μ_x es un promedio del vector X, μ_y es un promedio del vector Y, σ_x es una desviación estándar para el vector X, y σ_y es una desviación estándar para el vector Y.	
--	--	--

25. El sistema de la reivindicación 24, donde la computadora principal también está configurada para: determinar si la distancia semimétrica (PBR_{seq}) es mayor que un valor de umbral y clasificar la secuencia de ADN o ARN como tumor o normal en función de la determinación de si la distancia semimétrica (PBR_{seq}) es mayor que el valor de umbral.	25. El sistema de la reivindicación 24, donde la computadora principal también está configurada para: determinar si la distancia semimétrica (PBR_{seq}) es mayor que un valor de umbral y clasificar la secuencia de ADN o ARN como tumor o normal en función de la determinación de si la distancia semimétrica (PBR_{seq}) es mayor que el valor de umbral.	<i>Igual</i>
26. El sistema de la reivindicación 24, donde la computadora principal también está configurada para: identificar una variante rara en la secuencia de ADN o ARN.	26. El sistema de la reivindicación 24, donde la computadora principal también está configurada para: identificar una variante rara en la secuencia de ADN o ARN.	<i>Igual</i>
27. El sistema de la reivindicación 15, donde la computadora principal también está configurada para: determinar datos de rasgo de referencia de coincidencia más cercana de uno o más datos de rasgo de referencia.	27. El sistema de la reivindicación 15, donde la computadora principal también está configurada para: determinar datos de rasgo de referencia de coincidencia más cercana de uno o más datos de rasgo de referencia.	<i>Igual</i>

28. El sistema de la reivindicación 27, donde la computadora principal también está configurada para: identificar a una persona en función de la determinación de los datos de rasgo de referencia de coincidencia más cercana, donde la imagen digital comprende al menos uno de: una oreja, un rostro, una huella dactilar y un iris.	28. El sistema de la reivindicación 27, donde la computadora principal también está configurada para: identificar a una persona en función de la determinación de los datos de rasgo de referencia de coincidencia más cercana, donde la imagen digital comprende al menos uno de: una oreja, un rostro, una huella dactilar y un iris.	<i>Igual</i>
--	--	--------------

ADVANCE E-MAIL

From the INTERNATIONAL BUREAU

PCT

NOTIFICATION CONCERNING
TRANSMITTAL OF COPY OF INTERNATIONAL
PRELIMINARY REPORT ON PATENTABILITY
(CHAPTER I OF THE PATENT COOPERATION
TREATY)

(PCT Rule 44bis.1(c))

To:

KINNAIRD, James Welsh
Marks & Clerk Singapore LLP,
Tanjong Pagar,
P O Box 636,
Singapore 910816
SINGAPOUR

Date of mailing (day/month/year)

30 March 2017 (30.03.2017)

Applicant's or agent's file reference

FP7739

IMPORTANT NOTICE

International application No.

PCT/SG2015/050317

International filing date (day/month/year)

15 September 2015 (15.09.2015)

Priority date (day/month/year)

15 September 2014 (15.09.2014)

Applicant

TEMASEK LIFE SCIENCES LABORATORY LIMITED

The International Bureau transmits herewith a copy of the international preliminary report on patentability (Chapter I of the Patent Cooperation Treaty)

The International Bureau of WIPO
34, chemin des Colombettes
1211 Geneva 20, Switzerland

Authorized officer

Kihwan Moon

Facsimile No. +41 22 338 82 70

e-mail: pct.team1@wipo.int

PATENT COOPERATION TREATY

PCT

INTERNATIONAL PRELIMINARY REPORT ON PATENTABILITY

(Chapter I of the Patent Cooperation Treaty)

(PCT Rule 44bis)

Applicant's or agent's file reference FP7739	FOR FURTHER ACTION	See item 4 below
International application No. PCT/SG2015/050317	International filing date (<i>day/month/year</i>) 15 September 2015 (15.09.2015)	Priority date (<i>day/month/year</i>) 15 September 2014 (15.09.2014)
International Patent Classification (8th edition unless older edition indicated) See relevant information in Form PCT/ISA/237		
Applicant TEMASEK LIFE SCIENCES LABORATORY LIMITED		

1.	This international preliminary report on patentability (Chapter I) is issued by the International Bureau on behalf of the International Searching Authority under Rule 44 bis.1(a).																								
2.	This REPORT consists of a total of 6 sheets, including this cover sheet. In the attached sheets, any reference to the written opinion of the International Searching Authority should be read as a reference to the international preliminary report on patentability (Chapter I) instead.																								
3.	This report contains indications relating to the following items: <table style="width: 100%; border: none;"> <tr> <td style="width: 5%; text-align: center;">☒</td> <td style="width: 30%;">Box No. I</td> <td style="width: 65%;">Basis of the report</td> </tr> <tr> <td style="text-align: center;">☐</td> <td>Box No. II</td> <td>Priority</td> </tr> <tr> <td style="text-align: center;">☐</td> <td>Box No. III</td> <td>Non-establishment of opinion with regard to novelty, inventive step and industrial applicability</td> </tr> <tr> <td style="text-align: center;">☐</td> <td>Box No. IV</td> <td>Lack of unity of invention</td> </tr> <tr> <td style="text-align: center;">☒</td> <td>Box No. V</td> <td>Reasoned statement under Article 35(2) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement</td> </tr> <tr> <td style="text-align: center;">☐</td> <td>Box No. VI</td> <td>Certain documents cited</td> </tr> <tr> <td style="text-align: center;">☐</td> <td>Box No. VII</td> <td>Certain defects in the international application</td> </tr> <tr> <td style="text-align: center;">☒</td> <td>Box No. VIII</td> <td>Certain observations on the international application</td> </tr> </table>	☒	Box No. I	Basis of the report	☐	Box No. II	Priority	☐	Box No. III	Non-establishment of opinion with regard to novelty, inventive step and industrial applicability	☐	Box No. IV	Lack of unity of invention	☒	Box No. V	Reasoned statement under Article 35(2) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement	☐	Box No. VI	Certain documents cited	☐	Box No. VII	Certain defects in the international application	☒	Box No. VIII	Certain observations on the international application
☒	Box No. I	Basis of the report																							
☐	Box No. II	Priority																							
☐	Box No. III	Non-establishment of opinion with regard to novelty, inventive step and industrial applicability																							
☐	Box No. IV	Lack of unity of invention																							
☒	Box No. V	Reasoned statement under Article 35(2) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement																							
☐	Box No. VI	Certain documents cited																							
☐	Box No. VII	Certain defects in the international application																							
☒	Box No. VIII	Certain observations on the international application																							
4.	The International Bureau will communicate this report to designated Offices in accordance with Rules 44bis.3(c) and 93bis.1 but not, except where the applicant makes an express request under Article 23(2), before the expiration of 30 months from the priority date (Rule 44bis .2).																								

The International Bureau of WIPO 34, chemin des Colombettes 1211 Geneva 20, Switzerland Facsimile No. +41 22 338 82 70	Date of issuance of this report 21 March 2017 (21.03.2017) Authorized officer Kihwan Moon e-mail: pct.team1@wipo.int
--	--

PATENT COOPERATION TREATY

From the:
INTERNATIONAL SEARCHING AUTHORITY

PCT

WRITTEN OPINION OF THE
INTERNATIONAL SEARCHING AUTHORITY

(PCT Rule 43bis.1)

To: KINNAIRD, James Welsh MARKS & CLERK SINGAPORE LLP Tanjong Pagar, P O Box 636 910816 Singapore Singapore		Date of mailing (<i>day/month/year</i>) 06 November 2015	
Applicant's or agent's file reference FP7739		FOR FURTHER ACTION See paragraph 2 below	
International application No. PCT/SG2015/050317	International filing date (<i>day/month/year</i>) 15 September 2015	Priority date (<i>day/month/year</i>) 15 September 2014	
International Patent Classification (IPC) or both national classification and IPC G06T 7/00 (2006.01) G06K 9/78 (2006.01) G06F 19/24 (2011.01)			
Applicant SWAMINATHAN, Muthukaruppan et al.			

1. This opinion contains indications relating to the following items:

- | | | |
|-------------------------------------|--------------|---|
| ☒ | Box No. I | Basis of the opinion |
| ☐ | Box No. II | Priority |
| ☐ | Box No. III | Non-establishment of opinion with regard to novelty, inventive step and industrial applicability |
| ☐ | Box No. IV | Lack of unity of invention |
| ☒ | Box No. V | Reasoned statement under Rule 43bis.1(a)(i) with regard to novelty, inventive step and industrial applicability; citations and explanations supporting such statement |
| ☐ | Box No. VI | Certain documents cited |
| ☐ | Box No. VII | Certain defects in the international application |
| ☒ | Box No. VIII | Certain observations on the international application |

2. FURTHER ACTION

If a demand for international preliminary examination is made, this opinion will be considered to be a written opinion of the International Preliminary Examining Authority ("IPEA") except that this does not apply where the applicant chooses an Authority other than this one to be the IPEA and the chosen IPEA has notified the International Bureau under Rule 66.1bis(b) that written opinions of this International Searching Authority will not be so considered.

If this opinion is, as provided above, considered to be a written opinion of the IPEA, the applicant is invited to submit to the IPEA a written reply together, where appropriate, with amendments, before the expiration of 3 months from the date of mailing of Form PCT/ISA/220 or before the expiration of 22 months from the priority date, whichever expires later.

For further options, see Form PCT/ISA/220.

Name and mailing address of the ISA AUSTRALIAN PATENT OFFICE PO BOX 200, WODEN ACT 2606, AUSTRALIA Email address: pct@ipaustalia.gov.au	Date of completion of this opinion 6 November 2015	Authorised Officer Peter Thong AUSTRALIAN PATENT OFFICE (ISO 9001 Quality Certified Service) Telephone No. +61 262832128
--	---	---

WRITTEN OPINION OF THE
INTERNATIONAL SEARCHING AUTHORITY

International application No.

PCT/SG2015/050317

Box No. I **Basis of this opinion**

1. With regard to the **language**, this opinion has been established on the basis of:
 - ☒ The international application in the language in which it was filed
 - ☐ A translation of the international application into, , which is the language of a translation furnished for the purposes of international search (under Rules 12.3(a) and 23.1(b)).
2. ☐ This opinion has been established taking into account **the rectification of an obvious mistake** authorized by or notified to this Authority under Rule 91 (Rule 43*bis*.1(a))
3. With regard to any **nucleotide and/or amino acid sequence** disclosed in the international application, this opinion has been established on the basis of a sequence listing filed or furnished:
 - a. (means)
 - ☐ on paper
 - ☐ in electronic form
 - b. (time)
 - ☐ in the international application as filed
 - ☐ together with the international application in electronic form
 - ☐ subsequently to this Authority for the purposes of search
4. ☐ In addition, in the case that more than one version or copy of a sequence listing has been filed or furnished, the required statements that the information in the subsequent or additional copies is identical to that in the application as filed or does not go beyond the application as filed, as appropriate, were furnished.
5. Additional comments:

WRITTEN OPINION OF THE INTERNATIONAL SEARCHING AUTHORITY		International Application No. PCT/SG2015/050317																		
Box No. V	Reasoned statement under Rule 43bis.1(a)(i) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement																			
1. Statement <table style="width: 100%; margin-top: 10px;"> <tr> <td style="width: 30%;">Novelty (N)</td> <td style="width: 40%;">Claims 1-28</td> <td style="width: 30%; text-align: right;">YES</td> </tr> <tr> <td></td> <td>Claims NONE</td> <td style="text-align: right;">NO</td> </tr> <tr> <td>Inventive step (IS)</td> <td>Claims 1-28</td> <td style="text-align: right;">YES</td> </tr> <tr> <td></td> <td>Claims NONE</td> <td style="text-align: right;">NO</td> </tr> <tr> <td>Industrial applicability (IA)</td> <td>Claims 1-28</td> <td style="text-align: right;">YES</td> </tr> <tr> <td></td> <td>Claims NONE</td> <td style="text-align: right;">NO</td> </tr> </table> 2. CITATIONS AND EXPLANATIONS: <u>CITATIONS</u> D1: OZDEMIR, B., "Structural Scene Analysis of Remotely Sensed Images using Graph Mining," Master's Thesis, Bilkent University, Ankara, Turkey, 20 July 2010. D2: PELE, O., "Distance Functions: Theory, Algorithms and Applications", Thesis submitted for the degree of "Doctor of Philosophy", Hebrew University, July 2011. D3: CHESTER, U. et al., "Machine Learning for Image Classification and Clustering Using a Universal Distance Measure", SISAP 2013. LNCS, vol. 8199, pages 59 to 72, 2013. D4: SRIVASTAVA, A. et al., "On Advances in Statistical Modeling of Natural Images", Journal of Mathematical Imaging and Vision 18, pages 17 to 33, 2003. <u>NOVELTY (N)</u> None of the citations discloses a computer-implemented method for classifying a digital image that involves obtaining, from a host computer, feature data corresponding to the digital image; determining, by a graphics processing unit, a semi-metric distance based on a Poisson-Binomial distribution between the feature data and one or more reference feature data stored in a memory of the host computer; and classifying the digital image using the determined semi-metric distance. None of the citations discloses a system for classifying a digital image that involves a GPU obtaining, from a host computer, feature data corresponding to the digital image, accessing, from a memory, one or more reference feature data and determining a semi-metric distance based on a Poisson-Binomial distribution between the feature data and the one or more reference feature data, which system also involves the host computer classifying the digital image using the determined semi-metric distance. Therefore, the subject matter of claims 1 to 28 is new and meets the requirements of Article 33 (2) of the PCT with regard to novelty.			Novelty (N)	Claims 1-28	YES		Claims NONE	NO	Inventive step (IS)	Claims 1-28	YES		Claims NONE	NO	Industrial applicability (IA)	Claims 1-28	YES		Claims NONE	NO
Novelty (N)	Claims 1-28	YES																		
	Claims NONE	NO																		
Inventive step (IS)	Claims 1-28	YES																		
	Claims NONE	NO																		
Industrial applicability (IA)	Claims 1-28	YES																		
	Claims NONE	NO																		

INVENTIVE STEP (IS)

None of the documents either individually or in combination obviously suggests a computer-implemented method for classifying a digital image that involves obtaining, from a host computer, feature data corresponding to the digital image; determining, by a graphics processing unit, a semi-metric distance based on a Poisson-Binomial distribution between the feature data and one or more reference feature data stored in a memory of the host computer; and classifying the digital image using the determined semi-metric distance.

None of the documents either individually or in combination obviously suggests a system for classifying a digital image that involves a GPU obtaining, from a host computer, feature data corresponding to the digital image, accessing, from a memory, one or more reference feature data and determining a semi-metric distance based on a Poisson-Binomial distribution between the feature data and the one or more reference feature data, which system also involves the host computer classifying the digital image using the determined semi-metric distance.

Therefore, the subject matter of claims 1 to 28 is not obvious and meets the requirements of Article 33(3) of the PCT with regard to inventive step.

INDUSTRIAL APPLICABILITY (IA)

The subject matter of claims 1 to 28 is considered to meet the requirements of Industrial Applicability under Article 33(4) of the PCT because it can be made by, or used in, industry.

WRITTEN OPINION OF THE INTERNATIONAL SEARCHING AUTHORITY Box No. VIII Certain observations in the international application	International Application No. PCT/SG2015/050317
The following observations on the clarity of the claims, description, and drawings or on the question whether the claims are fully supported by the description, are made: <u>Observations on the clarity of the claims</u> • Claim 5 (line 1), which is appended to claim 1, is not clear because there is no antecedence for "the kNN classifier". It is noted that a kNN classifier is first mentioned in claim 4. For the purposes of this opinion, it is assumed that claim 5 is appended to claim 4 just like claim 19 (which includes features that are similar to those of claim 5) is appended to claim 18 (which includes features that are similar to those of claim 4). • Claim 6 (lines 2 to 3), which is appended to claim 5, is not clear because there is no antecedence for "the one or more reference data". Claim 20 (lines 2 to 3), which is appended to claim 19, is also not clear for the same reason. • Claim 24 (lines 6 to 7) is not clear in relation to the phrase "wherein the GPU is further configured to calculate determining the semi-metric distance (PBR_{seq}) comprises calculating".	

TRATADO DE COOPERACIÓN DE PATENTES

PCT/SG2015/050317

CORREO ELECTRONICO ANTICIPADO De la OFICINA INTERNACIONAL

PCT NOTIFICACIÓN RELATIVA A TRANSMISIÓN DE COPIAS DE INFORME PRELIMINAR INTERNACIONAL SOBRE LA PATENTABILIDAD (CAPÍTULO 1 DEL TRATADO DE COOPERACIÓN DE PATENTES) (Regla 44bis.1(c) del PCT)		A: KINNAIRD, James Welsh Marks & Clerk Singapur LLP, Tanjong Pagar, PO Box 636, Singapur 910816 SINGAPOUR	
Fecha de envío (día /mes/año) 30 de marzo de 2017 (30.03.2017)		AVISO IMPORTANTE	
Referencia del archivo del Solicitante o Agente FP7739			
Solicitud internacional No. PCT / SG2015 / 050317	Fecha de Radicación internacional (día/mes/año) 15 de septiembre de 2015 (15.09.2015)	Fecha de precedencia (día/mes/año) 15 de septiembre de 2014 (15.09.2014)	
Solicitante TEMASEK LIFE SCIENCES LABORATORY LIMITED			

La Oficina Internacional transmite con la presente una copia del informe preliminar internacional sobre patentabilidad (Capítulo 1 del Tratado de Cooperación de Patentes).

La Oficina Intencional de la WIPO 34. chemin des Colombettes 1211 Ginebra 20. Suiza	Funcionario Autorizado Kihwan Moon
Facsímil No. +41 22 338 82 70	e-mail: pct.team1@wipo.int

TRATADO DE COOPERACIÓN DE PATENTES

PCT

INFORME PRELIMINAR INTERNACIONAL SOBRE PATENTABILIDAD
(Capítulo 1 del Tratado de Cooperación de Patentes)

(Regla 44bis del PCT)

Referencia del archivo del solicitante o agente FP7739		PARA ACCION FUTURA		Ver el punto 4 siguiente
Solicitud internacional N°. PCT / SG2015 / 050317		Fecha de presentación internacional (día/mes/año) 15 de septiembre de 2015 (15.09.2015)		Fecha de precedencia (día/mes/año) 15 septiembre 2014 (15.09.2014)
Clasificación Internacional de Patentes (8ª edición a menos que se indique una edición anterior) Ver información relevante en el Formulario PCT/ISA/237				
Solicitante TEMASEK LIFE SCIENCES LABORATORY LIMITED				

1.	Este informe preliminar internacional sobre patentabilidad (Capítulo 1) es publicado por la Oficina Internacional en nombre de la Autoridad de Búsqueda Internacional de conformidad con la Regla 44bis.1. (a).	
2.	El INFORME consta de un total de 6 hojas, incluyendo esta hoja de cubierta. En las hojas adjuntas, toda referencia a la opinión escrita de la Autoridad de Búsqueda Internacional se debe interpretar como una referencia al informe preliminar internacional sobre patentabilidad (capítulo 1.).	
3.	Este informe contiene indicaciones relativas a los siguientes puntos:	
	☒ Cuadro No. I	Bases del informe
	☐ Cuadro No. II	Precedencia
	☐ Cuadro No. III	No establecimiento de opiniones con respecto a la novedad, la actividad inventiva y la aplicabilidad industrial
	☐ Cuadro No. IV	Falta de unidad de la invención
	☒ Cuadro No. V	Declaración razonada de conformidad con el Artículo 35(2) en cuanto a la novedad, nivel inventivo o aplicabilidad industrial; citas y explicaciones que apoyan tal declaración
	☐ Cuadro No. VI	Ciertos documentos citados
	☐ Cuadro No. VII	Determinados defectos de la solicitud internacional
	☒ Cuadro No. VIII	Ciertas observaciones sobre la solicitud internacional
4.	La Oficina Internacional comunicará este informe a las Oficinas designadas de conformidad con las Reglas 44bis.3. (c) y 93bis.1 pero no, excepto cuando el solicitante formule una solicitud expresa en virtud del Artículo 23 (2), antes de la expiración de 30 meses a partir de la fecha de precedencia (Regla 44bis.2).	

		Fecha de emisión del presente informe 21 de marzo de 2017 (21.03.2017)
La Oficina Internacional de la WIPO 34, chemin des Colombettes 1211 Ginebra 20, Suiza Facsimile No. +41 22 338 82 70		Funcionario autorizado Kiwi Moon e-mail: pct.team1@wipo.int

TRATADO DE COOPERACIÓN DE PATENTES

De la:
AUTORIDAD DE BÚSQUEDA INTERNACIONAL

Para:			PCT OPINIÓN ESCRITA DE LA AUTORIDAD DE BÚSQUEDA INTERNACIONAL (Regla 43bis.1 del PCT)	
KINNAIRD, James Welsh MARKS & CLERK SINGAPORE LLP Tanjong Pagar, P O Box 636 910816 Singapore Singapore				Fecha de envío (día/mes/año) 06 de noviembre de 2015
Referencia del archivo del solicitante o del agente FP7739				PARA ACCION ADICIONAL Ver el párrafo 2 más adelante
Solicitud internacional Nº PCT / SG2015 / 050317		Fecha de presentación internacional (día/mes/año) 15 de septiembre de 2015	Fecha de precedencia (día/mes/año) 15 septiembre 2014	

Clasificación Internacional de Patentes (IPC) o tanto la clasificación nacional como la CIP G06T 7/00 (2006.01) G06K 9/78 (2006.01) G06F 19/24 (2011.01)
Solicitante SW AMINA THAN, Muthukaruppan et al.

1. Esta opinión contiene indicaciones relativas a los siguientes puntos:		
☒	Cuadro No. I	Bases de la opinión
☐	Cuadro No. II	Precedencia
☐	Cuadro No. III	No establecimiento de opiniones con respecto a la novedad, la actividad inventiva y la aplicabilidad industrial
☐	Cuadro No. IV	Falta de unidad de la invención
☒	Cuadro No. V	Declaración razonada en virtud de la Regla 43 bis.1(a) (i), en relación con la novedad, nivel inventivo y la aplicabilidad industrial; citas y explicaciones que apoyan tal declaración
☐	Cuadro No. VI	Ciertos documentos citados
☐	Cuadro No. VII	Determinados defectos de la solicitud internacional
☒	Cuadro No. VIII	Ciertas observaciones sobre la solicitud internacional
2. OTRA ACCIÓN		
Si se hace una solicitud de examen preliminar internacional, se considerará que esta opinión es una opinión escrita de la Autoridad Examinadora Preliminar Internacional ("IPEA"), excepto que esto no se aplica cuando el solicitante elige a una Autoridad distinta a ésta para ser la IPEA y la IPEA elegida ha notificado a la Oficina Internacional de conformidad con la Regla 66.1bis (b) que los dictámenes escritos de esta Autoridad de Búsqueda Internacional no serán considerados.		
Si este dictamen como se ha proporcionado anteriormente es considerado como una opinión escrita de la IPEA, se invita al solicitante a que presente a la IPEA una respuesta por escrito, junto con, cuando sea necesario con modificaciones, antes de la expiración de 3 meses a partir de la fecha de envío del formulario PCT/ISA/220 o antes de la expiración de 22 meses a partir de la fecha de precedencia, cualquiera que expire más tarde.		
Para más opciones, ver el formulario PCT/ISA/220.		
Nombre y dirección postal de la ISA	Fecha de terminación de esta opinión	Funcionario Autorizado
AUSTRALIAN PATENT OFFICE PO BOX 200, WODEN ACT 2606, AUSTRALIA	6 de noviembre de 2015	Peter Thong AUSTRALIAN PATENT OFFICE (Servicio de Calidad Certificado ISO 9001) Teléfono No. +61 262832128
Direccion de Email:pct@ipaustrialia.gov.au		

OPINIÓN ESCRITA DE LA AUTORIDAD
DE BÚSQUEDA INTERNACIONAL

Solicitud internacional No.
PCT/SG2015/050317

Cuadro Nº 1	Bases de esta opinión
-------------	-----------------------

1.

Con respecto al **Lenguaje**, este dictamen se ha establecido sobre la base de:

☒

La solicitud internacional en la lengua en que fue presentada

☐

Una traducción de la solicitud internacional a, que es el idioma de una traducción expedida a efectos de la búsqueda internacional (de conformidad con las Reglas 12.3. (a) y 23.1. (b).

2.

☐ Esta opinión se ha establecido teniendo en cuenta **la rectificación de un error obvio** autorizado por o notificado a esta Autoridad de conformidad con la Regla 91 (Regla **43bis**.1.(a)

3.

Con respecto a cualquier secuencia de nucleótidos y/o aminoácidos descritos en la solicitud internacional, esta opinión se ha establecido sobre la base de una lista de secuencias presentada o suministrada:

a.

(medio)

☐

en papel

☐

en formato electrónica

b.

segundo (hora)

☐

en la solicitud internacional presentada

☐

junto con la solicitud internacional en formato electrónico

☐

posteriormente a esta Autoridad para fines de búsqueda

4.

☐ Además, en el caso que se haya presentado o suministrado más de una versión o copia de una lista de secuencias, las declaraciones requeridas de la información en las copias subsiguientes o adicionales es idéntica a las que fueron presentadas o suministradas o no más allá de la solicitud presentada, según proceda.

5.

Comentarios adicionales:

OPINIÓN ESCRITA DE LA AUTORIDAD DE BÚSQUEDA INTERNACIONAL		Solicitud internacional No. PCT/SG2015/050317	
Cuadro Nº V Declaración razonada en virtud de la Regla 43bis.1.(a) (i) en lo que respecta a la novedad, nivel inventivo o aplicabilidad industrial; citas y explicaciones que apoyan tal declaración			
1. Estatuto			
Novedad (N)		Reivindicaciones 1-28 Reivindicaciones NINGUNO	SI NO
Etapas de la invención (IS)		Reivindicaciones 1-28 Reivindicaciones NINGUNO	SI NO
Aplicabilidad industrial (IA)		Reivindicaciones 1-28 Reivindicaciones NINGUNO	SI NO
2 CITAS Y EXPLICACIONES:			
CITAS D1: OZDEMIR, B., "Structural Scene Analysis of Remotely Sensed Images using Graph Mining", Master's Thesis, Bilkent University, Ankara, Turkey, 20 July 2010. D2: PELE, O., "Distance Functions: Theory, Algorithms and Applications", Thesis submitted for the degree "Doctor of Philosophy", Hebrew, University, July 2011. D3: CHESTER, U. et al., "Machine Learning for Image Classification and Clustering Using a Universal Distance Measure", SISAP 2013. LNCS, vol. 8199, páginas 59 a 72, 2013. D4: SRIVASTA VA, A. et al., "On Advances in Statistical Modeling of Natural Images", Journal of Mathematical Imaging and Vision 18, pages 17 a 33, 2003. NOVEDAD (N) Ninguna de las citas describe un método implementado por computador para clasificar una imagen digital que implica obtener, a partir de un computador central, datos de características correspondientes a la imagen digital; determinar, mediante una unidad de procesamiento de gráficos, una distancia semimétrica con base en una distribución Binomial Poisson entre los datos de característica y uno o más datos de característica de referencia almacenados en una memoria del computador principal; y clasificar la imagen digital usando la distancia semimétrica determinada. Ninguna de las citas describe un sistema para clasificar una imagen digital que implica obtener un GPU, a partir de un computador principal, datos de características correspondientes a la imagen digital, que acceden a partir de una memoria, uno o más datos de características de referencia y que determinan una distancia semimétrica con base en una distribución Poisson-Binomial entre los datos de características y uno o más datos de características de referencia, sistema que también implica que el computador central clasifique la imagen digital usando la distancia semimétrica determinada. Por lo tanto, la materia objeto de las reivindicaciones 1 a 28 es nueva y cumple los requisitos del Artículo 33 (2) del PCT con respecto a la novedad.			

NIVEL INVENTIVO (IS)

Ninguno de los documentos, ya sea individualmente o en combinación, sugiere obviamente un método implementado por un computador para clasificar una imagen digital que implica obtener, a partir de un computador central, datos de características correspondientes a la imagen digital; determinar, mediante una unidad de procesamiento de gráficos, una distancia semimétrica con base en una distribución Binomial Poisson entre los datos de característica y uno o más datos de características de referencia almacenados en una memoria del computador principal; y clasificar la imagen digital usando la distancia semimétrica determinada.

Ninguno de los documentos, individualmente o en combinación, sugiere obviamente un sistema para clasificar una imagen digital que implica obtener una GPU, a partir de un computador principal, datos de características correspondientes a la imagen digital, que acceden, desde una memoria, a uno o más datos de características de referencia y determinar una distancia semimétrica con base en una distribución Poisson-Binomial entre los datos de características y uno o más datos de características de referencia, sistema que también implica que el computador principal clasifique la imagen digital usando la distancia semimétrica determinada.

Por lo tanto, la materia objeto de las reivindicaciones 1 a 28 no es obvia y cumple los requisitos del artículo 33 (3) del PCT con respecto al nivel inventivo.

APLICABILIDAD INDUSTRIAL (Ia)

Se considera que la materia objeto de las reivindicaciones 1 a 28 cumple los requisitos de la Aplicabilidad Industrial conforme al Artículo 33 (4) del PCT porque puede ser hecha por, o usada en, la industria.

Se formulan las siguientes observaciones sobre la claridad de las reivindicaciones, la descripción, y los dibujos o sobre la pregunta si las reivindicaciones están plenamente respaldadas por la descripción:

Observaciones sobre la claridad de las reivindicaciones

- La reivindicación 5 (renglón 1), que se anexa a la reivindicación 1, no es clara porque no hay antecedentes para "el clasificador kNN". Se observa que un clasificador kNN se menciona por primera vez en la reivindicación 4. A los efectos de esta opinión, se supone que la reivindicación 5 se adjunta a la reivindicación 4 al igual que la reivindicación 19 (que incluye características similares a las de la reivindicación 5) se adjunta a la reivindicación 18 (que incluye características similares a las de la reivindicación 4).
- La reivindicación 6 (renglones 2 a 3), que se adjunta a la reivindicación 5, no está clara porque no hay antecedentes para "uno o más datos de referencia". La reivindicación 20 (renglones 2 a 3), que se adjuntan a la reivindicación 19, tampoco es clara por la misma razón,
- La Reivindicación 24 (renglones 6 a 7) no está clara en relación con la frase "en donde la GPU está configurada adicionalmente para calcular la determinación de la distancia semimétrica (PBRseq) que comprende el cálculo".

**REIVINDICACIONES PATENTABLES U OTORGABLES POR LA OFICINA
INTERNACIONAL AUSTRALIANA**

1. A computer-implemented method for classifying a digital image/the method comprising:
obtaining, from a host computer, feature data corresponding to the digital image;
determining, by a graphics processing unit, a semi-metric distance based on a Poisson-Binomial distribution between the feature data and one or more reference feature data stored in a memory of the host computer; and
classifying the digital image using the determined semi-metric distance.
2. The method of claim 1, wherein the semi-metric distance is a Poisson-Binomial Radius (PBR).
3. The method of claim 1, wherein classifying the digital image comprises using a support vector machines (SVM) classifier.
4. The method of claim 1, wherein classifying the digital image comprises using a k-nearest neighbor (kNN) classifier.
5. The method of claim 1, wherein the kNN classifier is an adaptive local mean-based k-nearest neighbors (ALMkNN) classifier, wherein the value of the k-nearest neighbors (k) is adaptively chosen.
6. The method of claim 5, wherein the adaptive value of the k-nearest neighbors does not exceed the square root of the number of the one or more reference data.
7. The method of claim 1, wherein the obtained feature data and the one or more reference feature data comprise Pairwise Rotation Invariant Co-occurrence Local Binary Pattern (PRICoLBP) data.
8. The method of claim 1, wherein the obtained feature data and the one or more reference feature data comprise Histogram of Oriented Gradients (HOG) data.

9. The method of claim 1, wherein the obtained feature data comprises an N-dimensional feature vector X such that $X = (a_1 \dots a_N)$ and the reference feature data comprises an N-dimensional feature vector Y such that $Y = (b_1 \dots b_N)$, and wherein the determining the semi-metric distance ($PBR(X,Y)$) comprises calculating:

$$PBR(X,Y) = \frac{\sigma}{N - \mu} = \frac{\sqrt{\sum_{i=1}^N p_i(1 - p_i)}}{N - \sum_{i=1}^N p_i}$$

wherein N is an integer greater than 0,

σ is the standard deviation of vector X,

μ is the mean of vector X, and

p_i is $|a_i - b_i|$.

10. The method of claim 1, wherein the digital image comprises information corresponding to a DNA or RNA sequence, and the obtained feature data comprises a vector X of sequencing quality proximities for a first DNA sample with a sequencing depth d_x such that $X = (x_1 \dots x_{d_x})$ and the reference feature data comprises a vector Y of sequencing probabilities for a reference DNA sample with a sequencing depth d_y such that $Y = (y_1 \dots y_{d_y})$, and wherein the determining the semi-metric distance (PBR_{seq}) comprises calculating:

$$PBR_{seq}(X,Y) = \frac{\sigma_X \sigma_Y}{|\mu_X - \mu_Y|}$$

wherein μ_x is a means for vector X,

μ_y is a means for vector Y,

σ_x is a standard deviation for vector X, and

σ_y is a standard deviation for vector Y.

11. The method of claim 10, wherein the classifying the digital image comprises: determining whether the semi-metric distance (PBR_{seq}) is greater than a threshold value, and

classifying the DNA or RNA sequence as being a tumor or normal based on the determining if the semi-metric distance (PBR_{seq}) is greater than the threshold value.

12. The method of claim 10, wherein classifying the digital image comprises identifying a rare variant in the DNA or RNA sequence.

13. The method of claim 1, further comprising:

determining a closest matching reference feature data of the one or more reference feature data.

14. The method of claim 13, further comprising:

identifying a person based on the determined closest matching reference feature data, wherein the digital image comprises at least one of: an ear, a face, a fingerprint, and an iris.

15. A system for classifying a digital image comprising:

a host computer comprising a processor, wherein the host computer is coupled to a memory comprising one or more reference feature data; and

a graphics processing unit (GPU) comprising a processor,

wherein the GPU is coupled to the host computer and is configured to:

obtain, from the host computer, feature data corresponding to the digital image;

access, from the memory, the one or more reference feature data;

determine a semi-metric distance based on a Poisson-Binomial distribution between the feature data and the one or more reference feature data; wherein the host computer is configured to:

classify the digital image using the determined semi-metric distance.

16. The system of claim 15, wherein the semi-metric distance is a Poisson- Binomial Radius (PBR).

17. The system of claim 15, wherein the host computer is further configured to classify the digital image using a support vector machines (SVM) classifier.

18. The system of claim 15, wherein the host computer is further configured to classify the digital image using a k-nearest neighbor (kNN) classifier.

19. The system of claim 18, wherein the kNN classifier is an adaptive local mean-based k-nearest neighbors (ALMkNN) classifier, wherein the value of the k-nearest neighbors (k) is adaptively chosen.

20. The system of claim 19, wherein the adaptive value of the k-nearest neighbors (k) does not exceed the square root of the number of the one or more reference data.

21. The system of claim 15, wherein the feature data and the one or more reference feature data comprise Pairwise Rotation Invariant Co-occurrence Local Binary Pattern (PRICoLBP) data.

22. The system of claim 15, wherein the obtained feature data and the one or more reference feature data comprise Histogram of Oriented Gradients (HOG) data.

23. The system of claim 15, wherein the feature data comprises an N- dimensional feature vector X such that $X = (a_1 \dots a_N)$ and the reference feature data comprises an N-dimensional feature vector Y such that $Y = (b_1 \dots b_N)$, and wherein the GPU is further configured calculate:

$$PBR(X,Y) = \frac{\sigma}{N - \mu} = \frac{\sqrt{\sum_{i=1}^N p_i(1 - p_i)}}{N - \sum_{i=1}^N p_i}$$

wherein $PBR(X,Y)$ is a Poisson-Binomial Radius (PBR) distance between the vector X and the vector Y,

N is an integer greater than 0,

σ is the standard deviation of vector X,

μ is the mean of vector X, and

P_i is $|a_i - b_i|$.

24. The system of claim 15, wherein the digital image comprises information corresponding to a DNA or RNA sequence, and the feature data comprises a vector X of sequencing quality proximities for a first DNA sample with a sequencing depth dx such that $X = (x_1 \dots x_{dy})$ and the reference feature data comprises a vector Y of sequencing probabilities for a reference DNA sample with a sequencing depth dy such that $Y = (y_1 \dots y_{dy})$, and wherein the GPU is further configured to calculate determining the semi-metric distance (PBR_{seq}) comprises calculating:

$$\text{PBR}_{\text{seq}}(X, Y) = \frac{\sigma_X \sigma_Y}{|\mu_X - \mu_Y|}$$

wherein $\text{PBR}_{\text{seq}}(X, Y)$ is a Poisson-Binomial Radius (PBR) distance between the vector X and the vector Y,

μ_X is a means for the vector X,

μ_Y is a means for the vector Y,

σ_X is a standard deviation for the vector X, and

σ_Y is a standard deviation for the vector Y.

25. The system of claim 24, wherein the host computer is further configured to: determine whether the semi-metric distance (PBR_{seq}) is greater than a threshold value, and

classify the DNA or RNA sequence as being a tumor or normal based on the determining if the semi-metric distance (PBR_{seq}) is greater than the threshold value.

26. The system of claim 24, wherein the host computer is further configured to: identify a rare variant in the DNA or RNA sequence.

27. The system of claim 15, wherein the host computer is further configured to: determine a closest matching reference feature data of the one or more reference feature data.

28. The system of claim 27, wherein the host computer is further configured to: identify a person based on the determined closest matching reference feature data, wherein the digital image comprises at least one of: an ear, a face, a fingerprint, and an iris.

**REIVINDICACIONES PATENTABLES U OTORGABLES POR LA
OFICINA INTERNACIONAL AUSTRALIANA**

1. Un método implementado por computadora para clasificar una imagen digital, el método comprende:

obtener, de una computadora principal, datos de rasgos que corresponden a la imagen digital;

determinar, mediante una unidad de procesamiento gráfico, una distancia semimétrica en función de una distribución binomial de Poisson entre los datos de rasgos y uno o más datos de rasgos de referencia almacenados en una memoria de la computadora principal; y

clasificar la imagen digital usando la distancia semimétrica determinada.

2. El método de la reivindicación 1, donde la distancia semimétrica es un radio binominal de Poisson (PBR).

3. El método de la reivindicación 1, donde clasificar la imagen digital comprende usar un clasificador de máquinas de vectores de soporte (SVM).

4. El método de la reivindicación 1, donde clasificar la imagen digital comprende usar un clasificador de k vecino más cercano (kNN).

5. El método de la reivindicación 1, donde el clasificador kNN es un clasificador de k vecinos más cercanos basado en el promedio local adaptativo (ALMkNN), donde el valor de los k vecinos más cercanos (k) se elige de manera adaptativa.

6. El método de la reivindicación 5, donde el valor adaptativo de los k vecinos más cercanos no excede la raíz cuadrada del número de uno o más datos de referencia.

7. El método de la reivindicación 1, donde los datos de rasgo obtenidos y uno o más datos de rasgos de referencia comprenden datos de patrón binario local de concurrencia de invariancia rotacional en pares (PRICoLBP).

8. El método de la reivindicación 1, donde los datos de rasgo obtenidos y uno o más datos de rasgos de referencia comprenden datos de histograma de gradientes orientados (HOG).

9. El método de la reivindicación 1, donde los datos de rasgo obtenidos comprenden un vector de rasgo X de dimensión N tal que $X = (a_1 \dots a_N)$ y los datos de rasgo de referencia comprenden un vector de rasgo Y de dimensión N tal que $Y = (b_1 \dots b_N)$, y donde

determinar la distancia semimétrica (PBR(X,Y)) comprende calcular:

$$PBR(X,Y) = \frac{\sigma}{N - \mu} = \frac{\sqrt{\sum_{i=1}^N p_i(1 - p_i)}}{N - \sum_{i=1}^N p_i}$$

donde N es un entero mayor que 0,

σ es la desviación estándar del vector X,

μ es el promedio del vector X, y

p_i es $|a_i - b_i|$.

10. El método de la reivindicación 1, donde la imagen digital comprende información que corresponde a una secuencia de ADN o ARN y los datos de rasgo obtenidos comprenden un vector X de proximidades de calidad de secuenciación para una primera muestra de ADN con una profundidad de secuenciación dx tal que $X = (x_1 \dots x_{dx})$ y los datos de rasgo de referencia comprenden un vector Y de probabilidades de secuenciación para una muestra de ADN de referencia con una profundidad de secuenciación dy tal que $Y = (y_1 \dots y_{dy})$, y donde determinar la distancia semimétrica (PBR_{seq}) comprende calcular:

$$PBR_{seq}(X,Y) = \frac{\sigma_X \sigma_Y}{|\mu_X - \mu_Y|},$$

donde μ_x es un promedio del vector X,

μ_y es un promedio del vector Y,

σ_x es una desviación estándar para el vector X, y

σ_y es una desviación estándar para el vector Y.

11. El método de la reivindicación 10, donde la clasificación de la imagen digital comprende:

determinar si la distancia semimétrica (PBR_{seq}) es mayor que un valor de umbral y

clasificar la secuencia de ADN o ARN como tumor o normal en función de la determinación de si la distancia semimétrica (PBR_{seq}) es mayor que el valor de umbral.

12. El método de la reivindicación 10, donde clasificar la imagen digital comprende identificar una variante rara en la secuencia de ADN o ARN.

13. El método de la reivindicación 1, que comprende además:

determinar un dato de rasgo de referencia de coincidencia más cercana de uno o más datos de rasgo de referencia.

14. El método de la reivindicación 13, que comprende además:

identificar a una persona en función del dato de rasgo de referencia de coincidencia más cercana determinado, donde la imagen digital comprende al menos uno de: una oreja, un rostro, una huella dactilar y un iris.

15. Un sistema para clasificar una imagen digital que comprende:

una computadora principal que comprende un procesador, donde la computadora principal está acoplada a una memoria que comprende uno o más datos de rasgo de referencia; y

una unidad de procesamiento gráfico (GPU) que comprende un procesador,

donde la GPU está acoplada a la computadora principal y está configurada para:

obtener, de la computadora principal, datos de rasgo que corresponden a la imagen digital;

acceder, desde la memoria, a uno o más datos de rasgo de referencia;

determinar una distancia semimétrica en función de una distribución binomial de Poisson entre los datos de rasgo y uno o más datos de rasgo de referencia;

donde la computadora principal está configurada para:

clasificar la imagen digital usando la distancia semimétrica determinada.

16. El sistema de la reivindicación 15, donde la distancia semimétrica es un radio binomial de Poisson (PBR).

17. El sistema de la reivindicación 15, donde la computadora principal está configurada además para clasificar la imagen digital usando un clasificador de máquinas de vectores de soporte (SVM).

18. El sistema de la reivindicación 15, donde la computadora principal está configurada además para clasificar la imagen digital usando un clasificador de k vecino más cercano (kNN).

19. El sistema de la reivindicación 18, donde el clasificador kNN es un clasificador de k vecinos más cercanos basado en el promedio local adaptativo (ALMkNN), donde el valor de los k vecinos más cercanos (k) se elige de manera adaptativa.

20. El sistema de la reivindicación 19, donde el valor adaptativo de los k vecinos más cercanos (k) no excede la raíz cuadrada del número de uno o más datos de referencia.

21. El sistema de la reivindicación 15, donde los datos de rasgo y uno o más datos de rasgos de referencia comprenden datos de patrón binario local de concurrencia de invariancia rotacional en pares (PRICoLBP).

22. El sistema de la reivindicación 15, donde los datos de rasgo obtenidos y uno o más datos de rasgos de referencia comprenden datos de histograma de gradientes orientados (HOG).

23. El sistema de la reivindicación 15, donde los datos de rasgo comprenden un vector de rasgo X de dimensión N tal que $X = (a_1...a_N)$ y los datos de rasgo de referencia comprenden un vector de rasgo Y de dimensión N tal que $Y = (b_1...b_N)$, y donde la GPU también está configurada para calcular:

$$PBR(X,Y) = \frac{\sigma}{N - \mu} = \frac{\sqrt{\sum_{i=1}^N p_i(1 - p_i)}}{N - \sum_{i=1}^N p_i}$$

donde $PBR(X,Y)$ es una distancia de radio binomial de Poisson (PBR) entre el vector X y el vector Y ,

N es un entero mayor que 0,

σ es la desviación estándar del vector X ;

μ es el promedio del vector X , y

p_i es $|a_i - b_i|$.

24. El sistema de la reivindicación 15, donde la imagen digital comprende información que corresponde a una secuencia de ADN o ARN y los datos de rasgo comprenden un vector X de proximidades de calidad de secuenciación para una primera muestra de ADN con una profundidad de secuenciación dx tal que $X = (x_1 \dots x_{dx})$ y los datos de rasgo de referencia comprenden un vector Y de probabilidades de secuenciación para una muestra de ADN de referencia con una profundidad de secuenciación dy tal que $Y = (y_1 \dots y_{dy})$, y donde la GPU también está configurada para calcular y determinar la distancia semimétrica (PBR_{seq}) que comprende calcular:

$$PBR_{seq}(X, Y) = \frac{\sigma_X \sigma_Y}{|\mu_X - \mu_Y|},$$

donde $PBR_{seq}(X,Y)$ es una distancia de radio binomial de Poisson (PBR) entre el vector X y el vector Y ,

μ_x es un promedio del vector X ,

μ_y es un promedio del vector Y ,

σ_x es una desviación estándar para el vector X , y

σ_y es una desviación estándar para el vector Y .

25. El sistema de la reivindicación 24, donde la computadora principal también está configurada para:

determinar si la distancia semimétrica (PBR_{seq}) es mayor que un valor de umbral y

clasificar la secuencia de ADN o ARN como tumor o normal en función de la determinación de si la distancia semimétrica (PBR_{seq}) es mayor que el valor de umbral.

26. El sistema de la reivindicación 24, donde la computadora principal también está configurada para:

identificar una variante rara en la secuencia de ADN o ARN.

27. El sistema de la reivindicación 15, donde la computadora principal también está configurada para:

determinar datos de rasgo de referencia de coincidencia más cercana de uno o más datos de rasgo de referencia.

28. El sistema de la reivindicación 27, donde la computadora principal también está configurada para:

identificar a una persona en función de la determinación de los datos de rasgo de referencia de coincidencia más cercana, donde la imagen digital comprende al menos uno de: una oreja, un rostro, una huella dactilar y un iris.

STRUCTURAL SCENE ANALYSIS OF REMOTELY SENSED IMAGES USING GRAPH MINING

A THESIS

SUBMITTED TO THE DEPARTMENT OF COMPUTER ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Bahadır Özdemir

July, 2010

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Selim Aksoy (Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. ıgdem Gündüz Demir

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Tolga Can

Approved for the Institute of Engineering and Science:

Prof. Dr. Levent Onural
Director of the Institute

ABSTRACT

STRUCTURAL SCENE ANALYSIS OF REMOTELY SENSED IMAGES USING GRAPH MINING

Bahadır Özdemir

M.S. in Computer Engineering

Supervisor: Assist. Prof. Dr. Selim Aksoy

July, 2010

The need for intelligent systems capable of automatic content extraction and classification in remote sensing image datasets, has been constantly increasing due to the advances in the satellite technology and the availability of detailed images with a wide coverage of the Earth. Increasing details in very high spatial resolution images obtained from new generation sensors have enabled new applications but also introduced new challenges for object recognition. Contextual information about the image structures has the potential of improving individual object detection. Therefore, identifying the image regions which are intrinsically heterogeneous is an alternative way for high-level understanding of the image content. These regions, also known as compound structures, are comprised of primitive objects of many diverse types. Popular representations such as the bag-of-words model use primitive object parts extracted using local operators but cannot capture their structure because of the lack of spatial information. Hence, the detection of compound structures necessitates new image representations that involve joint modeling of spectral, spatial and structural information.

We propose an image representation that combines the representational power of graphs with the efficiency of the bag-of-words representation. The proposed method has three parts. In the first part, every image in the dataset is transformed into a graph structure using the local image features and their spatial relationships. The transformation method first detects the local patches of interest using maximally stable extremal regions obtained by gray level thresholding. Next, these patches are quantized to form a codebook of local information and a graph is constructed for each image by representing the patches as the graph nodes and connecting them with edges obtained using Voronoi tessellations. Transforming images to graphs provides an abstraction level and the remaining operations

for the classification are made on graphs. The second part of the proposed method is a graph mining algorithm which finds a set of most important subgraphs for the classification of image graphs. The graph mining algorithm we propose first finds the frequent subgraphs for each class, then selects the most discriminative ones by quantifying the correlations between the subgraphs and the classes in terms of the within-class occurrence distributions of the subgraphs; and finally reduces the set size by selecting the most representative ones by considering the redundancy between the subgraphs. After mining the set of subgraphs, each image graph is represented by a histogram vector of this set where each component in the histogram stores the number of occurrences of a particular subgraph in the image. The subgraph histogram representation enables classifying the image graphs using statistical classifiers. The last part of the method involves model learning from labeled data. We use support vector machines (SVM) for classifying images into semantic scene types. In addition, the themes distributed among the images are discovered using the latent Dirichlet allocation (LDA) model trained on the same data. By this way, the images which have heterogeneous content from different scene types can be represented in terms of a theme distribution vector. This representation enables further classification of images by theme analysis.

The experiments using an Ikonos image of Antalya show the effectiveness of the proposed representation in classification of complex scene types. The SVM model achieved a promising classification accuracy on the images cut from the Antalya image for the eight high-level semantic classes. Furthermore, the LDA model discovered interesting themes in the whole satellite image.

Keywords: Graph-based scene analysis, graph mining, scene understanding, remote sensing image analysis.

ÖZET

UYDU GÖRÜNTÜLERİNİN ÇİZGE MADENCİLİĞİ İLE YAPISAL SAHNE ANALİZİ

Bahadır Özdemir
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Yöneticisi: Y. Doç. Dr. Selim Aksoy
Temmuz, 2010

Uydu teknolojisindeki gelişmeler ve Dünya'nın geniş bir yüzeyini kapsayan detaylı görüntülerin mevcut olması, uydu görüntülerinde otomatik içerik çıkarma ve sınıflandırma yapabilen akıllı sistemlere duyulan ihtiyacı her geçen gün arttırmaktadır. Yeni nesil sensörlerden alınan çok yüksek uzamsal çözünürlüklü görüntülerdeki artan detaylar yeni uygulamaları mümkün kılmakla birlikte temel nesnelerin sezimini zorlaştırmaktadır. Görüntü yapıları hakkındaki bağlamsal bilgiler birbirinden bağımsız nesnelerin sezimini geliştirme potansiyeline sahiptir. Bu nedenle, özünde heterojen olan görüntü bölgelerinin tanımlanması, görüntü içeriğini anlamak için alternatif bir yoldur. Bileşik yapılar olarak da bilinen bu bölgeler birçok farklı türdeki temel nesnelerden oluşmaktadır. Kelimeler-torbası gibi popüler gösterimler, yerel operatörler kullanılarak çıkarılan temel nesne parçalarını kullanır fakat mekansal bilgi eksikliği nedeniyle onların yapısını tutamaz. Dolayısıyla, bileşik yapıların sezimi spektral, uzaysal ve yapısal bilgilerin ortak modellenmesini içeren yeni görüntü gösterimlerini zorunlu kılar.

Biz, çizgelerin gösterim gücü ile kelimeler-torbası gösteriminin verimliliğini birleştiren bir görüntü gösterimi öneriyoruz. Önerilen yöntem üç bölümden oluşmaktadır. İlk bölümde, veri kümesindeki her bir görüntü yerel görüntü özellikleri ve onların uzamsal ilişkileri kullanılarak çizge yapısına dönüştürülür. Dönüştürme yöntemi ilk olarak gri seviye eşikleme ile elde edilen en kararlı uç bölgelerden, ilgili yerel yamaları tespit eder. Sonra, bu yamalar bir yerel bilgi çizelgesi oluşturmak için nicelendirilir, ve yamaları çizge düğümü gibi göstererek ve onları Voronoi mozaikinden elde edilen kenarlarla birleştirerek her bir görüntü için bir çizge inşa edilir. Görüntülerin çizgelere dönüştürülmesi bir soyutlama düzeyi sağlar ve sınıflandırma için geriye kalan işlemler çizgeler üzerinde yapılır. Önerilen yöntemin ikinci bölümü görüntü çizgelerinin sınıflandırılması

için en önemli altçizgelerin kümesini seçen bir çizge madenciliği algoritmasıdır. Önerdiğimiz çizge madenciliği algoritması ilk olarak her sınıf için sık görülen altçizgeleri bulur, sonra sınıf içinde görülme dağılımları açısından altçizgeler ve sınıflar arasındaki bağıntı miktarları ölçülerek en ayırt edici olanları seçer; ve son olarak altçizgeler arasındaki fazlalığı dikkate alarak en iyi temsil edenlerin seçmesiyle küme boyutunu küçültür. Altçizge kümesi madenciliğinden sonra her bir görüntü çizgesi, her bir bileşenin bu kümenin belli bir altçizgesinin görüntüde görülme sayısını tuttuğu bir histogram vektörü ile gösterilir. Altçizge histogram gösterimi görüntü çizgelerinin istatistiksel sınıflandırıcılar kullanılarak sınıflandırılmasını mümkün kılar. Yöntemin son bölümü etiketli verilerinden model öğrenilmesini içerir. Görüntülerin anlamsal sahne türlerine sınıflandırılması için destek vektör makineleri (DVM) kullanıyoruz. Ek olarak, görüntüler üzerine dağılan temalar, aynı veriler üzerinde öğretilen gizli Dirichlet tahsisi (GDT) modeli kullanılarak keşfedilir. Bu sayede, farklı sahne türlerinden heterojen bir içeriğe sahip görüntüler bir tema dağılım vektörü olarak gösterilebilirler. Bu gösterim tema analizi ile görüntülerin daha ileri düzeyde sınıflandırılmasını sağlar.

Antalya'nın bir Ikonos görüntüsü üzerindeki deneyler önerilen gösterimin karmaşık sahne türlerinin sınıflandırılmasındaki etkinliğini göstermektedir. DVM modeli Antalya görüntüsünden kesilen görüntülerde sekiz üst düzey anlamsal sınıf için umut verici sınıflandırma doğruluğu elde etti. Ayrıca, GDT modeli tüm uydu görüntüsünde ilginç temalar keşfetti.

Anahtar sözcükler: Çizge tabanlı sahne analizi, çizge madenciliği, sahne anlayışı, uydu görüntüsü analizi.

Acknowledgement

I would like to express my sincere thanks to my advisor, *Selim Aksoy*, for his guidance, suggestions and support throughout the development of this thesis. He introduced me to the world of research, and encouraged me to develop my own ideas for the problem while supporting each step with his knowledge and advice. Whenever I got stuck in details, he provided me a different viewpoint. Working with him has been a valuable experience for me.

I would like extend my thanks to the members of my thesis committee, *Çiğdem Gündüz Demir* and *Tolga Can*, for reviewing this thesis and their suggestions about improving this work.

My special thanks must be sent to *Fatoş Tünay Yarman-Vural* who introduced me to computer vision when I was an undergraduate student at the Middle East Technical University.

I would like to express my deepest gratitude to my family, always standing by me, for their endless support and understanding.

I am very grateful to all those with whom I was having nice days in EA226: *Fırat, Daniya, Sare* and *Aslı*. I am also grateful to *Çağlar* and *Gökhan* for their comments on the method and the scientific discussions.

Finally, I would like to thank TÜBİTAK BİDEB (The Scientific and Technological Research Council of Turkey) for their financial support during my master's studies. This work was also supported in part by the TÜBİTAK CAREER grant 104E074.

Bahadır Özdemir
20 July 2010, Ankara

Contents

1	Introduction	1
1.1	Overview	1
1.2	Problem Definition	2
1.3	Data Set	5
1.4	Summary of Contributions	6
1.5	Organization of the Thesis	9
2	Literature Review	10
2.1	Classification with Visual Words	10
2.2	Classification with Graph Representation	11
3	Transforming Images to Graphs	14
3.1	Finding Regions of Interest	14
3.1.1	Maximally Stable Extremal Regions	16
3.1.2	Types of Interest Regions	20
3.2	Feature Extraction	21

3.3	Graph Construction	27
3.3.1	Nodes and Labels	27
3.3.2	Spatial Relationships and Edges	28
4	Graph Mining	32
4.1	Foundations of Pattern Mining	35
4.2	Frequent Pattern Mining	36
4.3	Class Correlated Pattern Mining	37
4.3.1	Mathematical Modeling of Pattern Support	38
4.3.2	Correlated Patterns	42
4.4	Redundancy-Aware Top- k Patterns	52
4.5	Summary of the Mining Algorithm	55
4.6	Graph Patterns	59
5	Scene Classification	64
5.1	Subgraph Histogram Representation	64
5.2	Support Vector Machines	65
5.3	Latent Dirichlet Allocation	66
6	Experimental Results	71
6.1	Experimental Setup	71
6.1.1	Graph Construction Parameters	72

6.1.2	Graph Mining Parameters	72
6.1.3	Classifier Parameters	74
6.2	Classification Results	74
7	Conclusions and Future Work	88
7.1	Conclusions	88
7.2	Future Work	90

List of Figures

1.1	Overall flowchart of the algorithm	4
1.2	An Ikonos image of Antalya, and some compound structures of interest are zoomed in. The classes are (in clockwise order): Sparse residential areas, orchards, greenhouses, fields, forests, dense residential areas with small buildings, dense residential areas with trees, and dense residential areas with large buildings.	7
3.1	Steps of transforming images to graphs	15
3.2	A given input image dark and bright MSERs, and ellipses fitted to them for parameters $\Omega = (\Delta, a_-, a_+, v_+, d_+) = (10, 60, 5000, 0.4, 1)$	19
3.3	Ellipses fitted to MSER groups stable_dark , stable_bright , unstable_dark and unstable_bright are drawn with green, red, yellow and cyan, respectively on different scene types for parameter sets $\Omega_{\text{high}} = (10, 60, 5000, 0.4, 1)$ and $\Omega_{\text{low}} = (5, 35, 1000, 4, 1)$	22
3.4	Satellite image of same region is given in (a) panchromatic and (d) visible multispectral bands. In (b) and (e), a given MSER is drawn with yellow and ellipse fitted to this MSER is drawn with green. Expanded ellipses at squared Mahalanobis distance $r_1^2 = 5$ and $r_2^2 = 20$ are drawn with red and cyan, respectively. In (c) and (f), pixels in R_{in} and R_{out} are shown for different bands.	23

3.5	Results of morphological operations on images from three different classes. Images from top to down are in the order: original images, images closed by disk with radii 2, images closed by disk with radii 7, images opened by disk with radii 2 and images opened by disk with radii 7.	25
3.6	A sample ellipse and its eigenvectors e_1 and e_2 are shown, corresponding eigenvalues are λ_1 and λ_2 , respectively. Major and minor diameters are also shown.	26
3.7	The problem of discovering neighboring node pairs in the Voronoi tessellation is shown in (a) and solution to this problem using external nodes is seen in (b). Corresponding graphs are given in (c) and (d), respectively.	30
3.8	Graph construction steps. The color and shape of a node in (d) represent its label after k -means clustering.	31
4.1	Steps of graph mining algorithm	34
4.2	Poisson distributions with four different expected values.	39
4.3	A sample histogram of a dataset with 100 elements and fitting mixtures of 3 Poisson distributions to this histogram are shown in blue and red, respectively.	42
4.4	The procedure for positive and negative distance computation is illustrated for four classes. The interest class is the second one and the distances are computed as $p = \text{EMD}(P_2, P_{\text{ref}})$ and $n = \text{EMD}(P_3, P_{\text{ref}})$	47
4.5	The correlation function $\gamma(p, n)$	48
4.6	Plot of a convex function f	51
4.7	Two sample redundant graph patterns	53

4.8	The pn space showing the search regions for the first two steps of the algorithm. The shaded area (union of dark and light gray) represent the domain region of $\mathcal{F}_c$ and dark gray area represents the domain region of $\mathcal{R}_c$	58
4.9	An example for overlapping embeddings	60
4.10	In (a) The embeddings of the subgraph in Figure 4.9(a); in (b) the corresponding overlap graph.	62
4.11	Images from top to down are original images from three different classes, image graphs for 36 labels, embeddings of sample subgraphs found by the mining algorithm and the sample subgraphs where the color and shape of a node represents its label.	63
5.1	Graphical model representation of LDA. The boxes are <i>plates</i> representing replicates. The outer plate represents image graphs, while the inner plate represents the repeated choice of themes and subgraphs within an image graph [7].	68
5.2	Graphical model representation of the variational distribution used to approximate the posterior in LDA [7].	69
6.1	Three clusters of stable dark MSERs are drawn with different colors at ellipse centers for $N_\ell = 36$. Yellow, green and magenta points are concentrated on dense residential areas with large buildings, dense residential areas with small buildings and orchards, respectively.	76
6.2	Four clusters of different type MSERs are drawn with different colors at ellipse centers for $N_\ell = 36$. Yellow, green, cyan and magenta points are concentrated on sea, forests, stream bed/clouds and dense residential areas with trees, respectively.	77

- 6.3 Plot of classification accuracy of the graph mining algorithm for five different number of labels over the number of subgraphs per class. The lines are drawn by averaging the accuracy values for the parameters $N_\theta \in \{200, 500, 800\}$ 79
- 6.4 Plot of classification accuracy of the graph mining algorithm for three different N_θ values over the number of subgraphs per class. The lines are drawn by averaging the accuracy values for the parameters $N_\ell \in \{18, 26, 36, 54, 72\}$ 81
- 6.5 The confusion matrix of the graph mining algorithm using the parameters $N_\ell = 36, N_\theta = 200$ and $N_s = 9$. Class names are given in short: *sparse* and *dense* are used for sparse and dense residential areas, respectively. Also, *large* and *small* mean large and small buildings, respectively. 83
- 6.6 The confusion matrix of the bag-of-words model for 26 labels. Class names are given in short: *sparse* and *dense* are used for sparse and dense residential areas, respectively. Also, *large* and *small* mean large and small buildings, respectively. 83
- 6.7 Sample images from the dataset. The images at the left are correctly classified by the graph mining algorithm while the images at right-hand side are misclassified using the parameters $N_\ell = 36, N_\theta = 200$ and $N_s = 9$. The image classes from top to down are in the order: dense residential areas with large buildings, dense residential areas with small buildings, dense residential areas with trees, sparse residential areas, greenhouses, orchards, forests and fields. 84
- 6.8 The classification of all tiles except sea using the SVM learned from the training set for the parameters $N_\ell = 36, N_\theta = 200$ and $N_s = 9$. Each color represents a unique class. 85

- 6.9 Every tile is labeled by a unique color which indicates the corresponding theme that dominates the other themes in that tile. The theme distributions are inferred from the LDA model for 12 themes. The subgraph set is the one mined in the previous experiments for the best parameters. 86
- 6.10 The most dominating 6 themes are shown, found by the LDA model trained for 16 themes. The intensity of red color represents the probability of the theme in an individual tile. 87

List of Tables

3.1	Ten basic features extracted from four bands and two regions. . .	23
6.1	The number of images in the training and testing datasets for each class. Class names are in the text.	75
6.2	The classification accuracy of the graph mining algorithm, in percentage (%), for all parameter sets in the experiments.	78
6.3	Classification accuracy of the bag-of-word model and the mining algorithm, in percentage terms, for different number of words/labels.	82

List of Algorithms

1	<code>k-means++</code> Algorithm, [3]	28
2	Greedy Algorithm for MMS, [45]	56
3	Pattern Mining Algorithm	57

Chapter 1

Introduction

Never use epigraphs, they kill the mystery in the work!

“THE BLACK BOOK” – ORHAN PAMUK

1.1 Overview

The amount of high-resolution satellite images is constantly increasing every day. Huge amount of information leads the requirement of automatic processing of remote sensing data by intelligent systems. Such systems usually perform image content extraction, classification and content-based retrieval in several application areas such as agriculture, ecology and urban planning. Very high resolution images become available by the advances in the satellite technology and processing of such images becomes feasible by the increasing computing power with the help of improvements in processor technology and parallel processing. This availability has enabled the study of multi-modal, multi-spectral, multi-resolution and multi-temporal data sets for monitoring purposes such as urban land use monitoring and management, geometric information system (GIS) and mapping, environmental change, site suitability, agricultural and ecological studies [2]. However, it also makes the problem of developing such intelligent systems more challenging because of the increased complexity.

Increasing details in very high spatial resolution images obtained from new generation sensors have been the main cause of the rising popularity of object-based approaches against traditional pixel-based approaches. Object-based approaches are aiming to identify primitive objects such as buildings and roads. Unfortunately, most algorithms cannot manage to detect such small objects in a very detailed image because segmentation algorithms usually fail to produce homogeneous regions corresponding to primitive structures. Contextual information about the image structures has the potential of improving individual object detection. Consequently, finding compound structures that correspond to high-level structures such as residential areas, forests, agricultural areas has become an alternative in image classification and high-level partitioning in the recent years because compound structures enable high-level understanding of image regions which are intrinsically heterogeneous [47]. Compound structures can be detected using local image features extracted from output of a segmentation algorithm or from interest points/regions. However, the detection of objects in such a detailed image is a difficult task. Therefore, some methods use textural analysis in lower resolution for detection of compound structures [42] or for detection/segmentation in high spatial resolution [19, 39]. In this thesis, we focus on representation of images by local image features with their spatial relationships and processing this representation model to detect compound structures in high spatial resolution.

1.2 Problem Definition

Pattern classification algorithms usually use one of the two traditional pattern recognition approaches: Statistical pattern recognition and syntactical/structural pattern recognition. Statistical approach uses feature vectors for object representation and generative or discriminative methods for modeling patterns in a vector space. The main advantage of this approach is available powerful algorithmic tools. On the other hand, structural approach uses strings or graphs for object representation. The main advantage of structural approach is the higher representation power and variable representation size. Both approaches have been used for detecting compound structures and image classification.

One of the statistical methods used for image classification is the *bag-of-words model*, which was originally developed for document analysis, adapted for images in [28]. Histogram of visual words obtained using a codebook constructed by quantizing local image patches has been a very popular representation for image classification in the recent years. This representation has been shown to give successful results for different image sets; however, a commonly accepted drawback is its disregarding of the spatial relationships among the individual patches as these relationships become crucial as contextual information for the understanding of complex scenes.

Structural approach used in image classification is aiming to represent images by graphs. Graphs provide powerful models where the nodes can store the local content and the edges can encode the spatial information. However, their use for image classification has been limited due to difficulties in translating complex image content to graph representation and inefficiencies in comparison of these graphs for classification. For example, the graph edit distance works well for matching relatively small graphs [37] but it can become quite restrictive for very detailed image content with a large number of nodes and edges.

We propose an intermediate representation that combines the representational power of graphs with the efficiency of the bag-of-words representation. The proposed method has three stages: transforming images into a graph representation, selecting the best subgraphs using a graph mining algorithm, and learning a model for each class to be used for classification. Figure 1.1 shows the overall flowchart of the algorithm.

Transforming images to graphs provides an abstraction level for images. Remaining operations for classification are made on graphs. Therefore, graphs transformed from images should contain sufficient information about the image content and spatial relationships. We describe a method for transforming the scene content and the associated spatial information of that scene into graph data. The method, which will be described in detail in Chapter 3, produces promising results on an Ikonos image of Antalya, Turkey (see Chapter 6).

The proposed approach represents each graph with a histogram of subgraphs

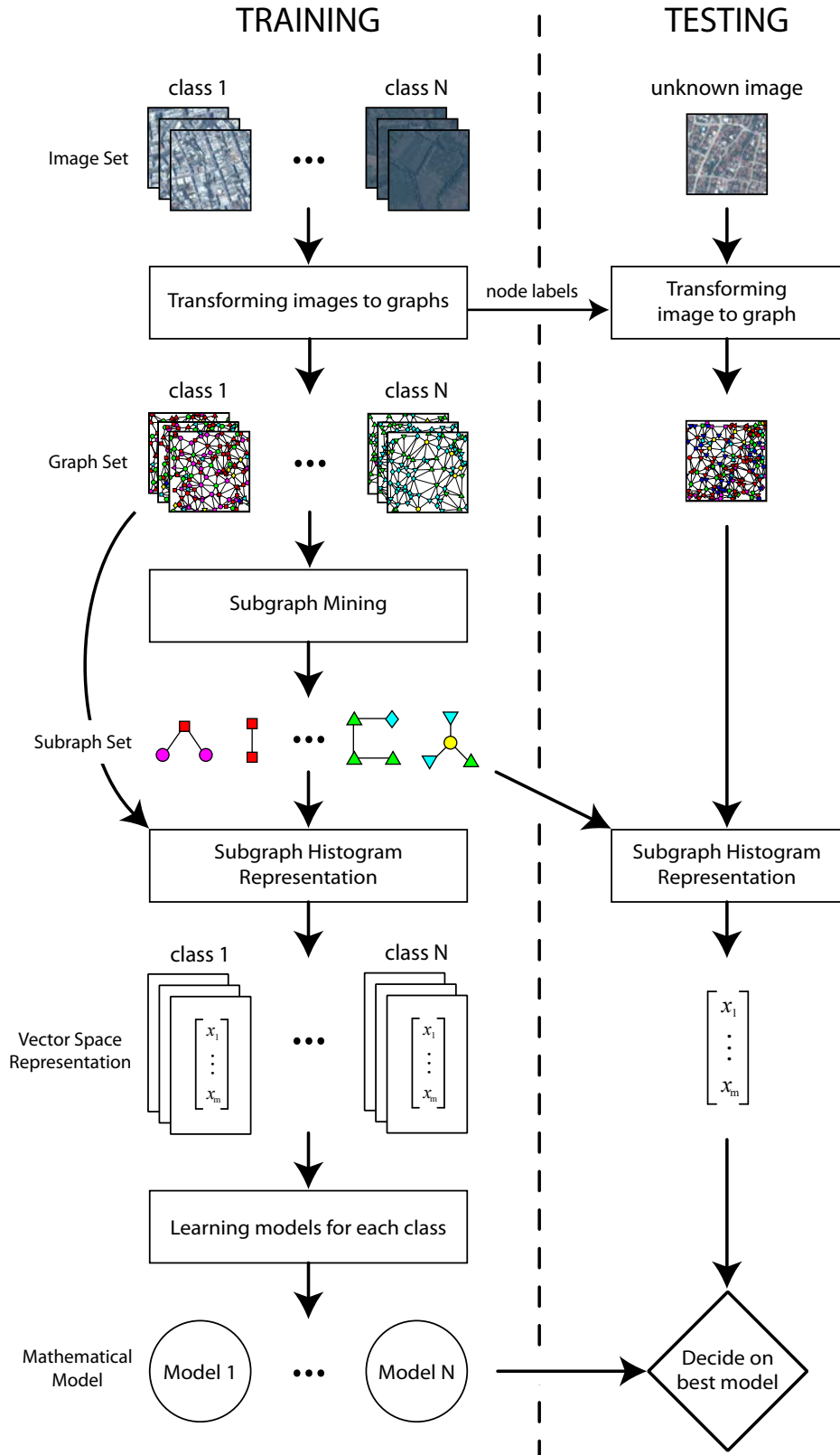


Figure 1.1: Overall flowchart of the algorithm

selected by a graph mining algorithm where the subgraphs encode the local patches and their spatial arrangements. The subgraphs are used to avoid the need of identifying a fixed arbitrary complexity (in terms of the number of nodes) and to require that they have a certain amount of support in different images in the data set. Partitioning remote sensing data into tiles usually produces images which contain heterogeneous regions of different classes. Some compound structures are naturally found near other structures. For example, orchards and greenhouses are usually detected near villages. Therefore, subgraphs selected by the algorithm should handle heterogeneous within-class content in an image set. A subgraph should also correspond to a structure particular to that class for classification purposes. Consequently, we propose a graph mining algorithm, where details can be found in Chapter 4, which tries to find a set of most important subgraphs considering frequency, correlation with classes and redundancy. Each image graph is represented by a histogram vector of this set in order to benefit from the advantages of statistical pattern recognition approach.

Finally, images represented by histogram vectors are classified in the vector space by traditional statistical classifiers. We employ *support vector machines* (SVM) for classifying images. In addition, topics/themes are discovered using latent probabilistic models such as *latent Dirichlet allocation* (LDA) that can be used for further classification of images for heterogeneous content. We show that good results for classification of images cut from large satellite scenes can be obtained for eight high-level semantic classes using support vector machines together with subgraph selection.

1.3 Data Set

The experiments are performed on an Ikonos image of Antalya, Turkey, consisting of a 12511×14204 pixel panchromatic band with 1m spatial resolution and four 3128×3551 pixel multi-spectral bands with 4m spatial resolution. In the experiments we use the panchromatic band and the pan-sharpened multi-spectral image produced by an image fusion method from visible multi-spectral bands and the

panchromatic band. The produced image approximates 1m spatial resolution in visible bands. We use the Antalya image because of its diverse content including several types of complex high-level structures such as dense and sparse residential areas with large and small buildings as well as fields and forests. The whole image was partitioned into 250×250 pixel tiles and these images were grouped into eight semantic classes, namely, (a) dense residential areas with large buildings, (b) dense residential areas with small buildings, (c) dense residential areas with trees, (d) sparse residential areas, (e) greenhouses, (f) orchards, (g) forests, and (h) fields. Only relatively homogeneous tiles, totally 585 images, are used in model learning and classification. The image and sample regions from every class are demonstrated in Figure 1.2.

1.4 Summary of Contributions

In this thesis, the goal is to correctly classify a given unknown image according to the models learned from training data for each class. Our framework for this aim has three parts and each part contains significant contributions.

The main contribution in the first part is a graph representation method for images. Although graphs offer higher representation power, their usage in computer vision has been below their usage in other fields. The primary reason for this issue is that images are not intrinsically in graph structure such as chemical compounds, program flows and social/computer networks. These data types come with their intrinsic graph structures and are perfectly suitable for structural approaches. The problem with graph representation of images is the difficulty of transforming image contents to graph structure. Most of the methods which construct graphs from images has used image segmentation algorithms so far [32, 23, 1, 22, 4]. In such methods, the regions in the output of segmentation usually correspond to graph nodes with labels determined by the features extracted from these regions whereas the edges encode the relationships between the regions. Unfortunately, precise segmentation of high spatial resolution satellite images as in Figure 1.2 is quite hard to obtain and this affects the performance

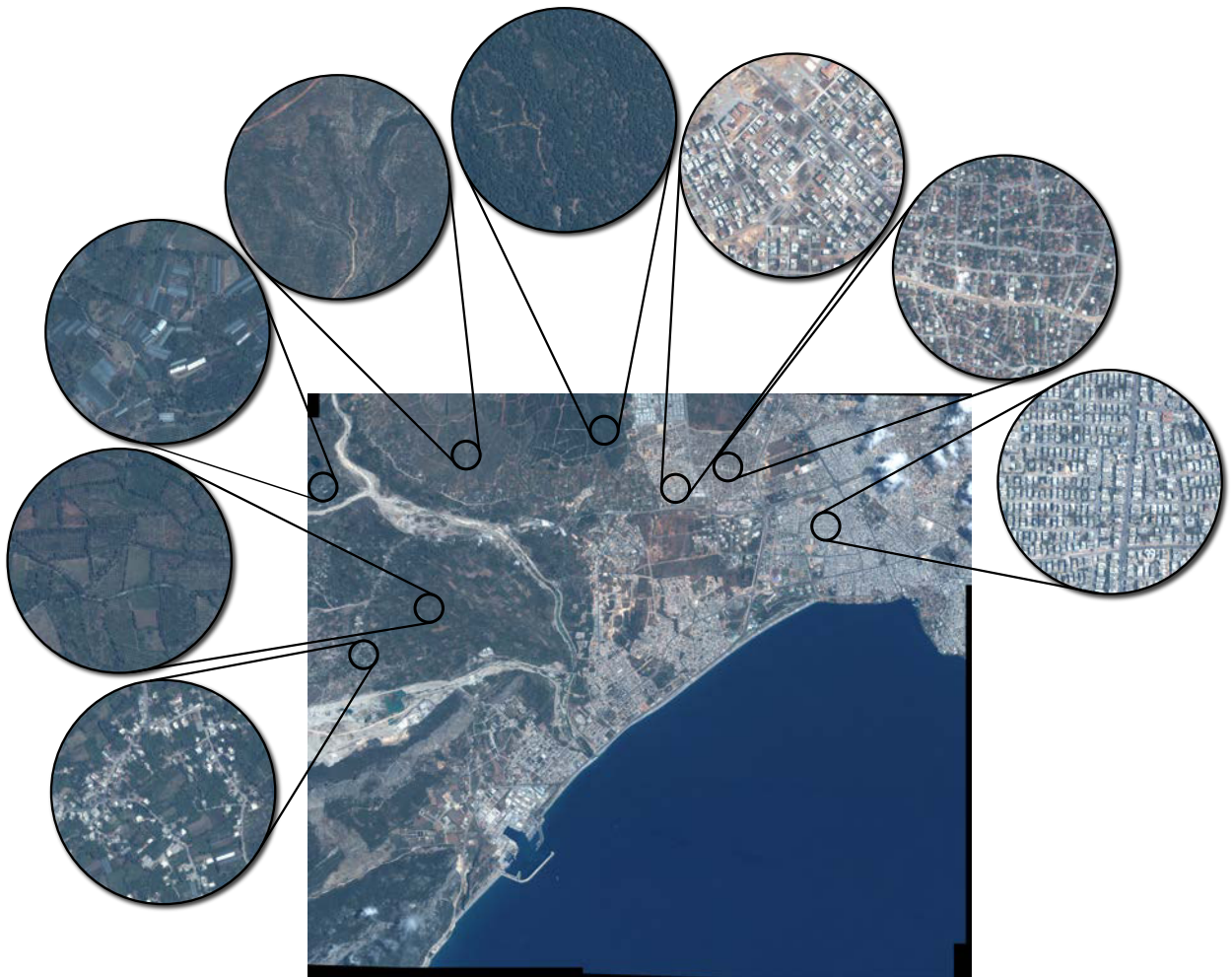


Figure 1.2: An Ikonos image of Antalya, and some compound structures of interest are zoomed in. The classes are (in clockwise order): Sparse residential areas, orchards, greenhouses, fields, forests, dense residential areas with small buildings, dense residential areas with trees, and dense residential areas with large buildings.

of graph representation negatively. Alternatively, we use regions of interests and their spatial relationships to transform image content into graph representation. Identifying only important regions in an image instead of whole image can supply sufficient information about the image content. First, local patches of interest are detected using *maximally stable extremal regions* obtained by gray level thresholding. We extract several features from these regions and their surroundings for better understanding of the regions. Next, these patches are quantized to form a codebook of local information, and a graph for each image is constructed by representing these patches as the graph nodes. The spatial relationships between the patches are identified using Voronoi tessellation and neighboring nodes are connected with edges. The abstraction level provided by the graph representation enables us to apply the same classification method on images coming from different sources like another satellite with different spatial resolution. For example, a QuickBird image can be classified in graph representation by a system trained on graphs constructed from Ikonos images as long as the node labels are compatible.

The second part proposes a graph mining algorithm to select the most important subgraphs for classification of graphs transformed from images. The mining algorithm we propose is a combination of three graph mining algorithms connected in series; in other words the output of one algorithm is the input of another one. The first algorithm seeks subgraphs frequently seen in a graph set. We use one of the popular algorithms in the graph mining literature for this purpose. Frequency criterion ensures the importance of subgraphs in the graph set. The most important contribution of this part is the second mining algorithm for finding correlated subgraphs which are frequently found in only one class of graphs and not in others. The available algorithms in the literature for correlated graph mining use a simple support definition which ignores the frequency of subgraph in a single graph and represents the support of a subgraph in a single graph as a binary relation of existence or absence [10, 34, 33]. We propose a novel algorithm where the frequency of subgraphs in a single graph are considered in the calculation of subgraph correlation (details are in Section 4.3). This method enhances classification performance considerably when images of a class cannot be fully homogeneous such as greenhouses seen in Figure 1.2. In such cases, this

method seeks subgraphs which are common among examples of that class, i.e. particular to that class. Final mining algorithm removes redundant subgraphs to avoid curse of dimensionality and selects the most significant subgraphs. The second and third mining algorithms work like a filter. They allow some subgraphs to pass to the next algorithm if they satisfy the criteria of the algorithms. The final set of subgraphs satisfying all criteria is used for representing a graph as a histogram vector where each component of the vector is the frequency of the corresponding subgraph in the given graph.

The third and last part is the classification of images using their vector representations by traditional classifiers like support vector machines. Addition to this, we use latent Dirichlet allocation to discover topics (themes) and their distribution in the image. This an important contribution because finding a homogeneous tile of a satellite image becomes harder when the tile size increases. Experimental results of the proposed methods are given in Chapter 6.

1.5 Organization of the Thesis

The rest of the thesis is organized as follows. Chapter 2 presents an overview of related works in the literature. Chapter 3 introduces the method of transforming an image into graph representation. In Chapter 4, we first give a brief introduction to graph mining and then describe our graph mining algorithm. Chapter 5 explains learning models used for classification. Experimental results are given in Chapter 6, and Chapter 7 provides conclusions and future work.

Chapter 2

Literature Review

*The knowledge and learning that we have, is, at most,
but little compared with that of which we are ignorant.*

PLATO

In this chapter, we give the review of the previous studies on image classification using the bag-of-words model or the graph representation. The methods are divided into two sections according to their image representation. In the first section, we describe some image classification methods which are based on the bag-of-words model but also consider the spatial information of visual words. The second section describes the graph representation of images in the literature and their applications to image classification and retrieval.

2.1 Classification with Visual Words

The visual word concept is introduced in [28] as an image patch represented by a codeword from a large vocabulary of codewords. The vocabulary called codebook is formed by quantizing the image patches. Hence, an image is represented with a histogram of visual words. This analogy enables the usage of generative

probabilistic models of text corpora such pLSI and LDA in computer vision applications. These probabilistic models are based on the bag-of-words assumption [7], the exchangeability of visual words, that the location of patches in an image can be neglected. According to a recent survey [24], the bag-of-words model has been extended by weighting scheme, stop word removal, feature selection, spatial information and visual bi-gram. In relation to our study, we describe the extension methods which are using the spatial information and/or bi-gram of the visual words.

In [26], Lazebnik *et al.* add geometric correspondences to visual words by partitioning the image into increasingly sub-regions and computing the histograms of local features found inside each subregion. In [29], Li *et al.* propose the contextual bag-of-words representation to model two kinds of typical contextual relations between local patches, i.e., a semantic conceptual relation and a spatial neighboring relation. For the semantic conceptual relation, visual words are grouped on multiple semantic levels with respect to the similarity of the class distribution induced by the patches. To explore the spatial neighboring relation, the algorithm uses the visual n-gram approach. According to Yuan *et al.* [46], the clustering of primitive visual features tends to result in synonymous and polysemous visual words that bring large uncertainties and ambiguities in the representation. To overcome these problems, they propose a method which generates a higher-level lexicon, i.e. visual phrase lexicon, where a visual phrase is a meaningful spatially co-occurrent pattern of visual words. The method employs several data mining techniques and pattern summarization, with modifications to fit the image data.

2.2 Classification with Graph Representation

In this section, we give some previous works which use graph structure for image representation especially for classification and indexing/retrieval. An attributed relational graph (ARG) is a graph with attributes (also called labels or weights) on its nodes and/or edges. In computer vision applications, they are usually created from the output of a segmentation algorithm where each segment is denoted

by a node, and the edges are used to reflect the adjacent relations among the segments. In [23] ARGs are used to find the common pattern of the input images by finding the maximal common subgraph in the ARGs. In [1], Aksoy described a hierarchical approach for the content modeling and retrieval of satellite images using ARGs that combine region class information and spatial arrangements. The retrieval operation uses the graph edit distance [32] as the dissimilarity measure between two ARGs. Harchaoui and Bach propose *graph kernels* for supervised classification of image graphs constructed in a similar way from the morphological segmentation of images [21]. Another graph type used for image representation is *hypergraphs* where each edge is a subset of the set of nodes for modeling the higher-order relations between nodes [5]. Bunke *et al.* use hypergraphs to represent fingerprint images and classify those graphs using a hypergraph matching algorithm [11]. Unlike previous methods which construct image graph from the output of segmentation, in [20] Gao *et al.* construct graphs from corner points and Delaunay triangulation for the images of real world objects in black background. They cluster and classify image graphs by computing the graph edit distance between pairwise graphs.

Some methods transform the graphs constructed from images into feature vector and classify images in the vector space by statistical algorithms. These algorithm can be divided into two groups. In the first group of algorithm, each graph is transformed into a vector such that each of the components corresponds to the distance of the input graph to a predefined reference graph set. The studies [37] and [12] employ this approach for the datasets of symbol/letter images and fingerprint images using the Lipschitz Embedding [9] and the dissimilarity space representation [36], respectively. In the second group of algorithms, each graph is represented by a frequency vector of a subgraph set where the i th component is the number of occurrences of the i th subgraph in the input graph. The subgraph set is found by a graph mining algorithm for some criteria like frequency. A set of subgraphs found by the frequent subgraph mining of region-adjacency graphs is used for image indexing [22] and for clustering document images [4]. In [35], Nowozin *et al.* use weighted substructure mining which is combination of graph

mining and the boosting algorithm in order to classify images. In graph construction, each interest point is represented by one vertex and its descriptor becomes the corresponding vertex label and all vertices are connected by undirected edges with labels determined by the distance between two interest points.

Chapter 3

Transforming Images to Graphs

*One morning, when Gregor Samsa woke from troubled dreams,
he found himself transformed in his bed into a horrible vermin.*

“THE METAMORPHOSIS” – FRANZ KAFKA

The first step of the algorithm is transforming every image to a graph structure as seen in Figure 1.1. Local image features and the relationships between them are encoded in the graph representation. In this chapter, we focus on this transformation process. Figure 3.1 shows the details for a sample image. First, local patches of interest in an image are detected using maximally stable extremal regions (MSER) obtained by gray level thresholding. Next, these patches are quantized to form a codebook of local information, and a graph for each image is constructed by representing these patches as the graph nodes and connecting them with edges obtained using Voronoi tessellations. The details of each step are explained in the following sections.

3.1 Finding Regions of Interest

The maximally stable extremal regions enable us to model local image content without the need for a precise segmentation that can be quite hard for high spatial

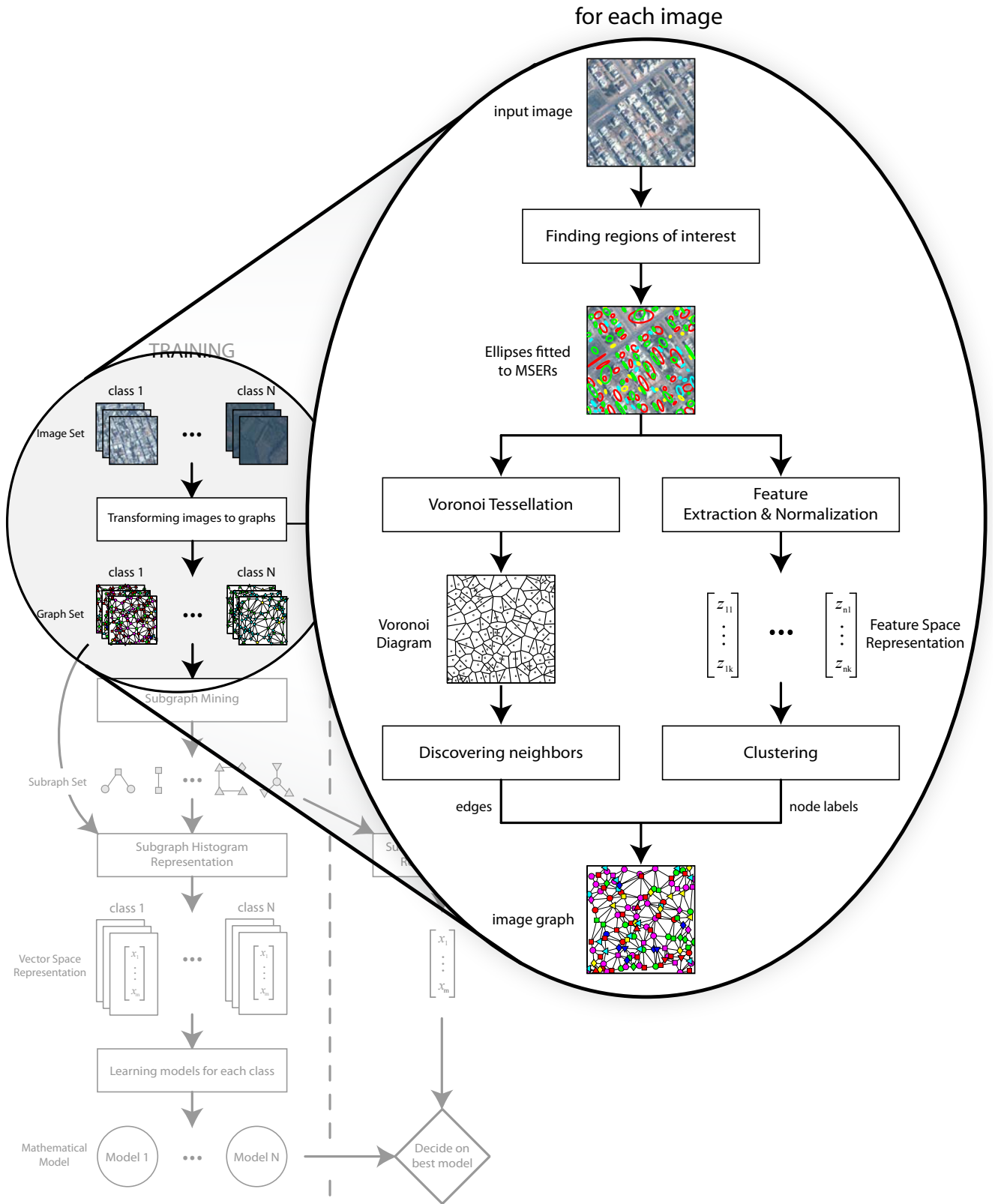


Figure 3.1: Steps of transforming images to graphs

resolution satellite images. In the following Section 3.1.1 the MSER algorithm is briefly described. The effects of MSER parameters for detecting regions of interest and different types of regions used in the algorithm are explained in Section 3.1.2.

3.1.1 Maximally Stable Extremal Regions

In this section, we introduce the *Maximally Stable Extremal Regions* (MSER), a new type of image elements proposed by Matas *et al.* in [31]. The regions are selected according to their extremal property of the intensity function in the region and on its outer boundary. The formal definition of the MSER concept and the necessary auxiliary definitions are given below.

Definition 3.1 (Maximally Stable Extremal Regions, [31]).

Image I is a mapping $I : D \subset \mathbb{Z}^2 \rightarrow S$. Extremal regions are well defined on images if:

1. S is totally ordered, i.e. reflexive, antisymmetric and transitive binary relation $\leq$ exists. Extremal regions can be defined on $S = \{0, 1, \dots, 255\}$ or real-valued images ($S = \mathbb{R}$).
2. An adjacency relation $A \subset D \times D$ is defined. For example, 4-neighborhoods are used; $p, q \in D$ are adjacent (pAq) iff $\sum_{i=1}^d |p_i - q_i| \leq 1$.

Region Q is a contiguous subset of D , i.e. for each $p, q \in Q$ there is a sequence $p, a_1, a_2, \dots, a_n, q$ and $pAa_1, a_iAa_{i+1}, a_nAq$.

(Outer) Region Boundary $\partial Q = \{q \in D \setminus Q \mid \exists p \in Q : qAp\}$, i.e. the boundary ∂Q of Q is the set of pixels adjacent to at least one pixel of Q but not belonging to Q .

Extremal Region $Q \subset D$ is a region such that either for all $p \in Q, q \in \partial Q : I(p) > I(q)$ (maximum intensity region) or $p \in Q, q \in \partial Q : I(p) < I(q)$ (minimum intensity region).

Maximally Stable Extremal Region Let $Q_1, \dots, Q_{i-1}, Q_i, \dots$ be a sequence of nested extremal regions, i.e. $Q_i \subset Q_{i+1}$. Extremal region Q_{i*} is maximally stable iff $q(i) = |Q_{i+\Delta} \setminus Q_{i-\Delta}| / |Q_i|$ has a local minimum at $i*$. $\Delta \in S$ is a parameter of the method.

The MSER algorithm is similar to the *watershed algorithm* except their outputs. In watershed computation, we deal with only the thresholds where regions merge, so resultant regions are highly unstable. In MSER detection, we seek a range of thresholds where the size of regions are effectively unchanged. Since every extremal region is a connected component of a thresholded image, all possible thresholds are applied to image and the stability of extremal regions are evaluated to find MSERs.

As given in the formal definition 3.1 the intensity of extremal regions can be less or greater than its boundary. We prefer calling dark MSER and bright MSER for minimum intensity MSER and maximum intensity MSER, respectively. The algorithm is generally implemented to detect dark MSERs and the intensity of input image is inverted to detect bright MSERs.

In our study, we use the VLFeat implementation of the MSER algorithm [43]. This implementation provides a rotation-invariant region descriptor and additional parameters which offer extra control over selection of MSERs. These parameters are related to area, variation (stability) and the diversity of extremal regions.

Let Q_i be an extremal region at the threshold level i . The following tests are performed for every MSER:

- *Area*: exclude too small or too big MSERs, $a_- \leq |Q_i| \leq a_+$.
- *Variation*: exclude too unstable MSERs, $v(Q_i) < v_+$ where VLFeat implementation differently uses stability score as $v(Q_i) = |Q_{i+\Delta} \setminus Q_i| / |Q_i|$.
- *Diversity*: remove duplicated MSERs, for any MSER Q_i find the parent

MSER Q_j and check if $|Q_j \setminus Q_i| / |Q_j| < d_+$ where Q_j is the parent of Q_i iff $Q_i \subset Q_j$ for $i \leq j \leq i + \Delta$.

We denote MSER parameter set as $\Omega = (\Delta, a_-, a_+, v_+, d_+)$. These parameters are used to eliminate less important extremal regions, i.e. too small or too big regions. The stability criterion is adjusted by parameters both Δ and v_+ . The graph representation should encode both local image features and their spatial relationships correctly. Therefore, regions of interests should not share any pixel like in segmentation to transform planar relationships between regions. However, multiple thresholds may yield stable extremal regions for some parts of the image and the output is nested subset regions [31]. In this study, we always set $d_+ = 0$ to prevent overlapping extremal regions (actually one covers another).

Ellipsoids

MSERs have arbitrary shapes as seen in Figures 3.2(b) and 3.2(c) for given input image in Figure 3.2(a). Therefore, many implementations return extremal regions as a set of ellipsoids fitted to actual regions. Ellipsoids are represented with two parameters: mean vector and covariance matrix of the pixels composing the region. The parameters $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ of extremal region Q are computed as

$$\boldsymbol{\mu} = \frac{1}{|Q|} \sum_{\mathbf{x} \in Q} \mathbf{x}, \quad \boldsymbol{\Sigma} = \frac{1}{|Q|} \sum_{\mathbf{x} \in Q} (\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^\top \quad (3.1)$$

where the pixel coordinate $\mathbf{x} = (x_1, \dots, x_n)^\top$ uses the standard index order and ranges. The MSER algorithm can also be applied to volumetric images; however, in this study we only deal with 2D grayscale images ($n = 2$). Thus, $\boldsymbol{\mu}$ has two components and $\boldsymbol{\Sigma}$ has three independent components because covariance matrix is a symmetric positive definite matrix. Ellipses fitted to MSERs in Figures 3.2(b) and 3.2(c) are drawn in Figures 3.2(d) and 3.2(e), respectively. The ellipses are drawn at $(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) = 1$. *

*The quantity $r^2 = (\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})$ is called the squared Mahalanobis distance from $\mathbf{x}$ to $\boldsymbol{\mu}$.



(a) input image



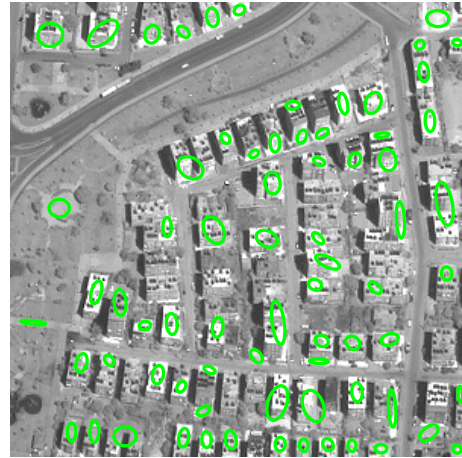
(b) dark MSERs



(c) bright MSERs



(d) ellipses fitted to dark MSERs



(e) ellipses fitted to bright MSERs

Figure 3.2: A given input image dark and bright MSERs, and ellipses fitted to them for parameters $\Omega = (\Delta, a_-, a_+, v_+, d_+) = (10, 60, 5000, 0.4, 1)$.

3.1.2 Types of Interest Regions

To handle all regions of interest by a single global parameter set is hard to obtain for an image set including different complex scene types. For example, extremal regions observed in urban areas are usually highly stable while such an observation in fields is less possible. We define two parameter sets with different stability criteria, Ω_{high} and Ω_{low} , to detect extremal regions such as in both urban areas and fields. In addition, it allows us to group extremal regions according to their stability scores. Applying the MSER algorithm with these parameters on both the intensity image (for dark MSERs) and on the inverted image (for bright MSER) results in four different region groups as:

- Highly stable dark MSERs (`stable_dark`)
- Highly stable bright MSERs (`stable_bright`)
- Less stable dark MSERs (`unstable_dark`)
- Less stable bright MSERs (`unstable_bright`)

Due to the definition of MSER, less stable MSERs cover highly stable ones. Therefore, we use restrictions on less stable ones. The set definitions of these four groups are given by

$$\text{stable_dark}(I) = \{R \mid R \subset I \wedge R \text{ is an MSER satisfying } \Omega_{\text{high}}\}, \quad (3.2)$$

$$\text{stable_bright}(I) = \{R \mid R \subset \bar{I} \wedge R \text{ is an MSER satisfying } \Omega_{\text{high}}\} \quad (3.3)$$

where $\bar{I}$ denotes the intensity inverted image of I . Similarly, less stable ones are defined as

$$\begin{aligned} \text{unstable_dark}(I) = \{R \mid R \subset I \wedge R \text{ is an MSER satisfying } \Omega_{\text{low}}, \\ \wedge \forall R' \in \text{stable_dark}(I) : R \cap R' = \emptyset\}, \end{aligned} \quad (3.4)$$

$$\begin{aligned} \text{unstable_bright}(I) = \{R \mid R \subset \bar{I} \wedge R \text{ is an MSER satisfying } \Omega_{\text{low}} \\ \wedge \forall R' \in \text{stable_bright}(I) : R \cap R' = \emptyset\} \end{aligned} \quad (3.5)$$

Figure 3.3 shows these four groups of MSERs for three different scene types. As seen in the figure, stable MSERs are observed especially on buildings and their shadows while unstable ones are seen everywhere like random sampling.

3.2 Feature Extraction

We extract several features from MSERs to identify the location where they are observed. Interest regions become more discriminative with their surroundings. The size of ellipses fitted to MSERs are expanded before extracting features from these regions. This method is proposed by Sivic *et al.* in [40]. We group the pixels inside expanded ellipses into two sets. The first set represents the MSER region and consists of pixels near to ellipse center whereas the other group containing outer pixels represents the surroundings of the MSER. As mentioned previously, each MSER is represented with two parameters $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. We denote the inner and outer groups of pixels as R_{in} and R_{out} , respectively. Image I is defined on $D \subset \mathbb{Z}^2$, then two groups are defined by

$$R_{in} = \{\mathbf{x} \in D \mid (\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}) \leq r_1^2\}, \quad (3.6)$$

$$R_{out} = \{\mathbf{x} \in D \mid r_1^2 < (\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}) \leq r_2^2\} \quad (3.7)$$

where every $\mathbf{x}$ represents a single pixel coordinate. For a given MSER, expanded ellipses and the pixels in regions R_{in} , R_{out} are shown on both panchromatic and multispectral bands in Figure 3.4.

We extract 17 rotation-invariant features from each MSER. Exactly 10 of them are basic features such as mean and standard deviation extracted from both R_{in} and R_{out} . Table 3.1 shows these basic 10 features.

The other 7 features are computed from the union group, $R_{all} = R_{in} \cup R_{out}$. These are 4 granulometry features, area and aspect ratio of ellipse, and moment of inertia.

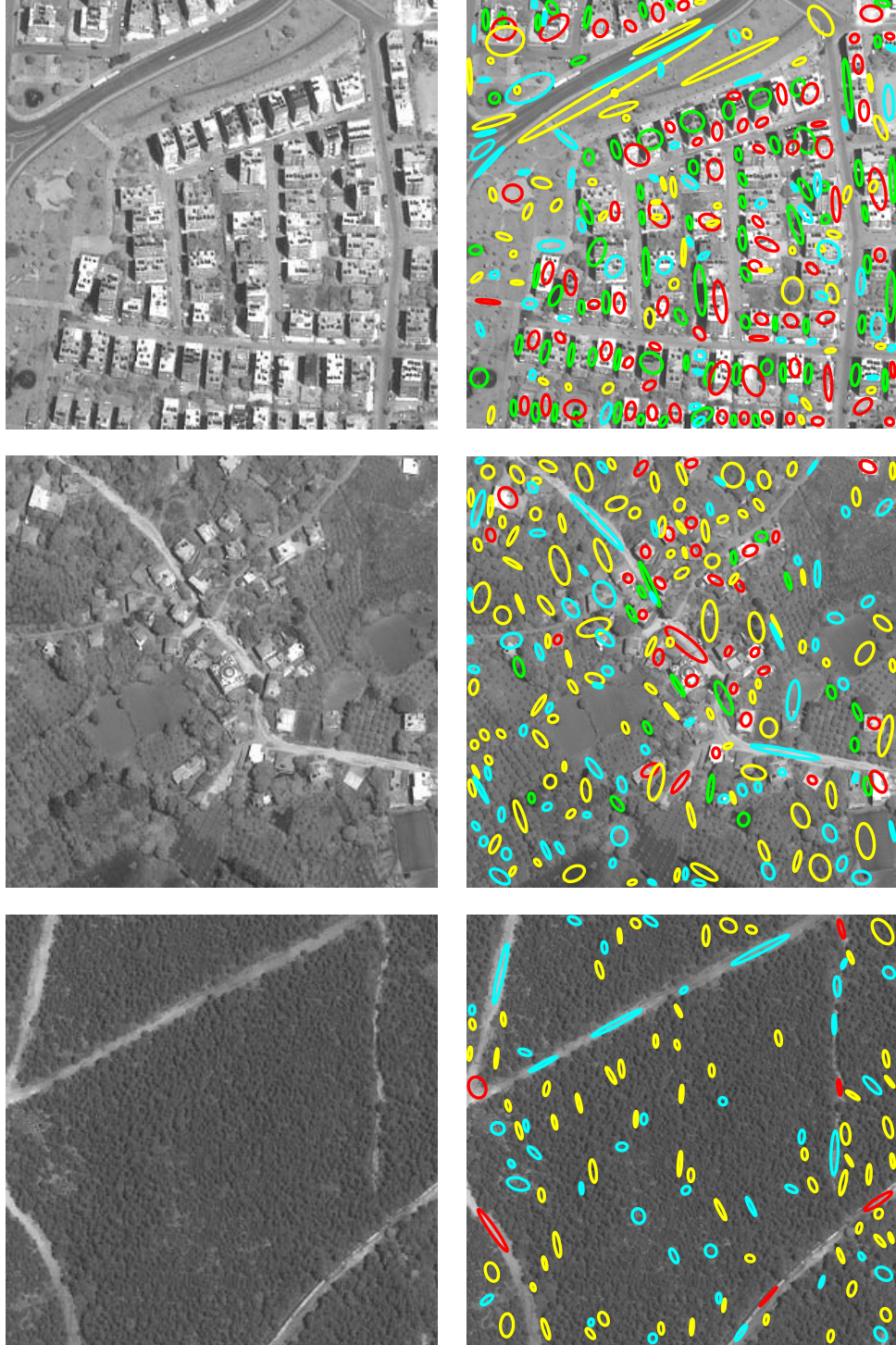


Figure 3.3: Ellipses fitted to MSER groups `stable_dark`, `stable_bright`, `unstable_dark` and `unstable_bright` are drawn with green, red, yellow and cyan, respectively on different scene types for parameter sets $\Omega_{\text{high}} = (10, 60, 5000, 0.4, 1)$ and $\Omega_{\text{low}} = (5, 35, 1000, 4, 1)$.

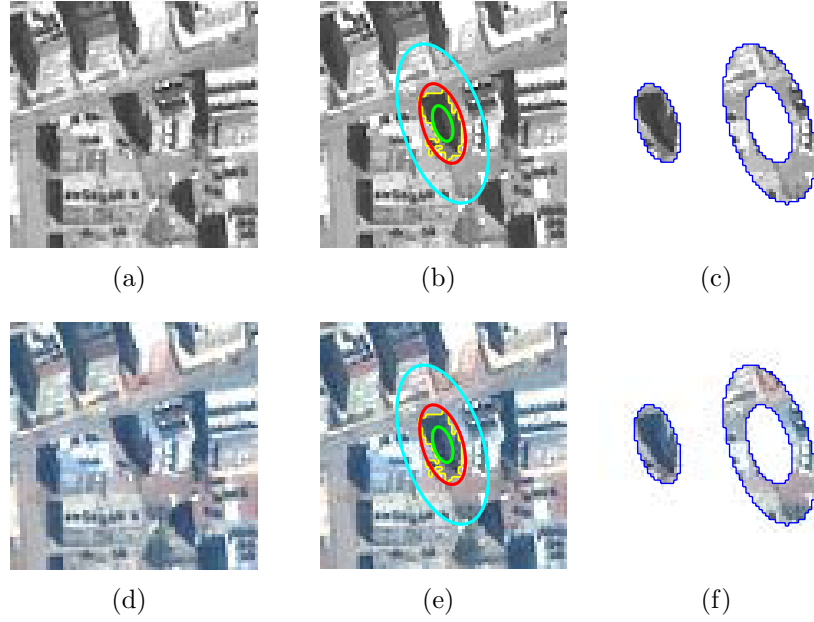


Figure 3.4: Satellite image of same region is given in (a) panchromatic and (d) visible multispectral bands. In (b) and (e), a given MSER is drawn with yellow and ellipse fitted to this MSER is drawn with green. Expanded ellipses at squared Mahalanobis distance $r_1^2 = 5$ and $r_2^2 = 20$ are drawn with red and cyan, respectively. In (c) and (f), pixels in R_{in} and R_{out} are shown for different bands.

Table 3.1: Ten basic features extracted from four bands and two regions.

			Region	
			R_{in}	R_{out}
Image	Panchromatic band		mean, standard deviation	mean, standard deviation
	Multispectral bands	Red band	mean	mean
		Green band	mean	mean
		Blue band	mean	mean

Granulometry

Granulometry is a technique to analyze the size and shape of granular materials. The idea is based on sieving a sample through various sized and shaped sieves [44]. A collection of grains is analyzed by sieving through sieves with increasing mesh size while measuring the mass retained by each sieve [41].

The concept of granulometry is extended for images by considering them as grains and applying morphological opening and closing with a family of structuring elements with increasing sizes [41]. Morphological opening provides information about image contents which are brighter than their neighborhoods and in contrast closing operation gives information about regions darker than their neighborhoods. Size information of these structures are obtained from the size of structuring element used in the morphological operation. Besides the information gained from standard deviation, granulometry produces useful information about the arrangement of objects in the expanded ellipse region.

We use only two sizes of structuring element, a disk with radii 2 and 7. They are employed to detect smaller and bigger structures in the image, respectively. The granulometry features are extracted from the region R_{all} in panchromatic band using morphological opening and closing, resulting in 4 granulometry features. Let ψ denote the structuring element, we compute the granulometry feature, Φ , known as normalized size distribution as

$$\Phi(I, \psi) = \frac{\sum_{\mathbf{x} \in R_{all}} (I \circ \psi)(\mathbf{x})}{\sum_{\mathbf{x} \in R_{all}} I(\mathbf{x})} \quad (3.8)$$

where $\circ$ denotes morphological opening; for morphological closing features it should be replaced by $\bullet$ denoting morphological closing.

Figure 3.5 shows results of morphological opening and closing with disk structuring element with radii 2 and 7 on sample images from three different classes. As shown in the figure, the urban area image is affected from morphological operations the most and the forest image is affected the least.

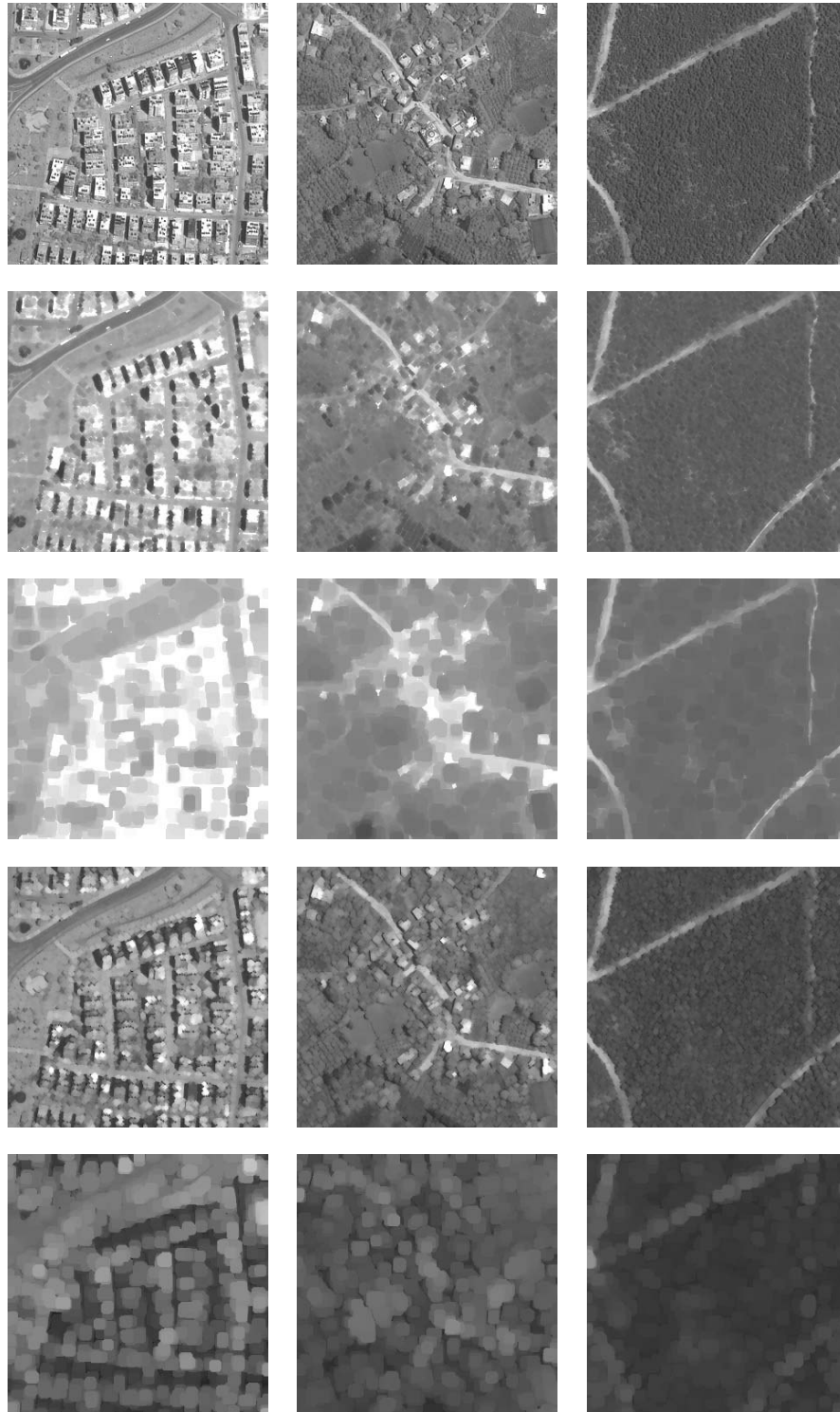


Figure 3.5: Results of morphological operations on images from three different classes. Images from top to down are in the order: original images, images closed by disk with radii 2, images closed by disk with radii 7, images opened by disk with radii 2 and images opened by disk with radii 7.

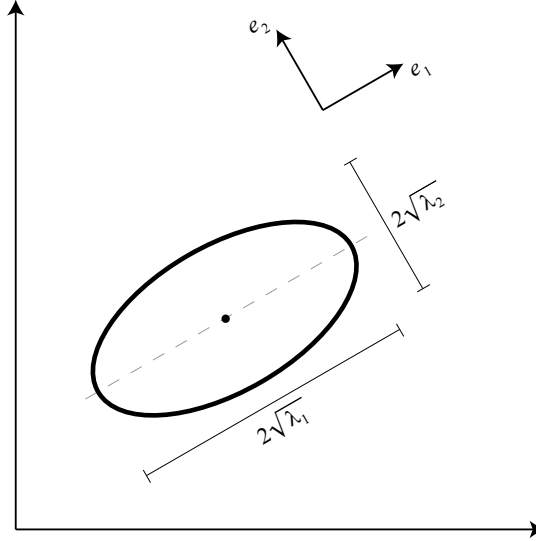


Figure 3.6: A sample ellipse and its eigenvectors e_1 and e_2 are shown, corresponding eigenvalues are λ_1 and λ_2 , respectively. Major and minor diameters are also shown.

Moment of Inertia

Another feature computed from R_{all} is the moment of inertia. It provides useful information about intensity distribution in the expanded region with respect to the distance to ellipse center. The level of intensity change between the MSER and its surrounding can be identified with this feature. The formula is given below

$$M_I = \frac{\sum_{\mathbf{x} \in R_{all}} I(\mathbf{x}) \cdot (\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) / r_2^2}{\sum_{\mathbf{x} \in R_{all}} I(\mathbf{x})}. \quad (3.9)$$

The value of M_I is in the range $[0, 1]$ due to division by r_2^2 in the numerator.

Area and Aspect Ratio of Ellipse

The last two features are the area and aspect ratio of ellipse. These features give information about the shape of MSER. These features are calculated using the eigenvalues of $\boldsymbol{\Sigma}$. Figure 3.6 shows a sample ellipse, its eigenvectors and eigenvalues. Let λ_1 and λ_2 be the eigenvalues of $\boldsymbol{\Sigma}$ in descending order. The area of the ellipse is equal to $\pi\sqrt{\lambda_1\lambda_2}$ and the aspect ratio is equal to $\sqrt{\lambda_1/\lambda_2}$.

3.3 Graph Construction

We have tried to extract local image features thus far. As the next step, we discretize the features extracted from MSERs in order to construct a codebook. By this way, each MSER will be a visual word from the codebook. Image representation by visual words is called the *bag of words* representation [28]. However, this method ignores the relationships between visual words. Instead, we propose a graph representation which encapsulates local image features as well as the spatial information of the scene.

The definition of a labeled graph is given below and the graph construction steps are described in the following subsections.

Definition 3.2 (Labeled graph, [18]).

A labeled or attributed graph is a triplet $G = (V, E, \ell)$, where V is the set of vertices, $E \subseteq V \times V - \{(v, v) \mid v \in V\}$ is the set of edges, and $\ell : V \cup E \rightarrow \Gamma$ is a function that assigns labels from the set Γ to nodes and edges.

3.3.1 Nodes and Labels

Now, we have 17-dimensional feature vectors for each MSER in 4 different groups. These are discretized using *k-means clustering* separately for each group. We employ the *k-means++ algorithm* proposed by Arthur and Vassilvitskii in [3] owing to its better seed initialization. It can be seen in Algorithm 1. Each MSER corresponds to a graph node where its label is determined from the output of the *k-means* algorithm. In other words, the set of vertices V is the union of four region groups and the labeling function ℓ is a mapping from MSERs to the output of the clustering algorithm performed for every region group. The parameter of the clustering algorithm, number of clusters k , has a major effect on the performance of image classification. This effect will be discussed in Chapters 6. The algorithm is applied to each region group, so the parameter set for the number of labels is denoted by $\Upsilon = (k_{sd}, k_{sb}, k_{ud}, k_{ub})$ where the initials of the region groups are used as the indexes of the parameters. We normalize each feature to zero mean

and unit variance before applying the k -means algorithm. Cluster centers and normalization parameters are also used in the testing stage. For an unknown image, the labels of graph nodes are assigned according to the closest cluster center to the feature vector after the normalization.

Algorithm 1 `k-means++` Algorithm, [3]

Input: Set of data points, $\mathcal{X}$

Number of clusters, k

Output: Clusters of data points, C

- 1: Choose an initial center c_1 uniformly at random from $\mathcal{X}$.
 - 2: Choose the next center c_i , selecting $c_i = x' \in \mathcal{X}$ with probability $\frac{D(x')^2}{\sum_{x \in \mathcal{X}} D(x)^2}$ where $D(x)$ denotes the shortest distance from a data point x to the closest center we have already chosen.
 - 3: Repeat Step 2 until we have chosen a total of k centers.
 - 4: Proceed as with the standard k -means algorithm.
-

3.3.2 Spatial Relationships and Edges

The final step of graph construction is to connect every neighboring node pair with an undirected edge. To do so, we locate the nodes in V at ellipse centers. We can determine whether given two nodes are neighbors or not by computing the Euclidean distance between the nodes and comparing it to a threshold. However, such a threshold is scale dependent [17] and cannot be automatically set for different scenes because the density of nodes in different types of scenes differs. In addition, a global threshold defined for all scene types creates more complex graphs for the images in which large number of nodes are found such as urban areas and it may produce unconnected nodes for the images with fewer number of nodes such as fields. To handle such problems we use the Voronoi tessellation where the nodes correspond to the cell centroids. The nodes whose cells are neighbors (sharing an edge) in the Voronoi tessellation are considered as neighbor nodes and are connected by undirected edges. In other words, the set of edges can be given by

$$E = \{(u, v) \mid u, v \in V \wedge u \text{ and } v \text{ are neighbors in the Voronoi diagram}\} \quad (3.10)$$

and the labeling function ℓ assigns the same trivial label to every edge, means we ignore edge labels.

The Voronoi tessellation successfully partitions the image region; however, some cell pairs which are not neighboring inside the image region may become neighboring outside the image region as in Figure 3.7(a). The graph constructed from this tessellation includes unnecessary edges between some outer nodes that can be seen in Figure 3.7(c). Our solution to this problem is to construct graph from whole remote sensing image and then to cut this graph into tiles (see Figures 3.7(b) and 3.7(d)).

All steps of graph construction are shown in Figure 3.8. This process is applied to every image in both training and testing stages. As a result, we produce a set of graphs which encode image content appropriately for each image and this set provides an abstraction level for new images.

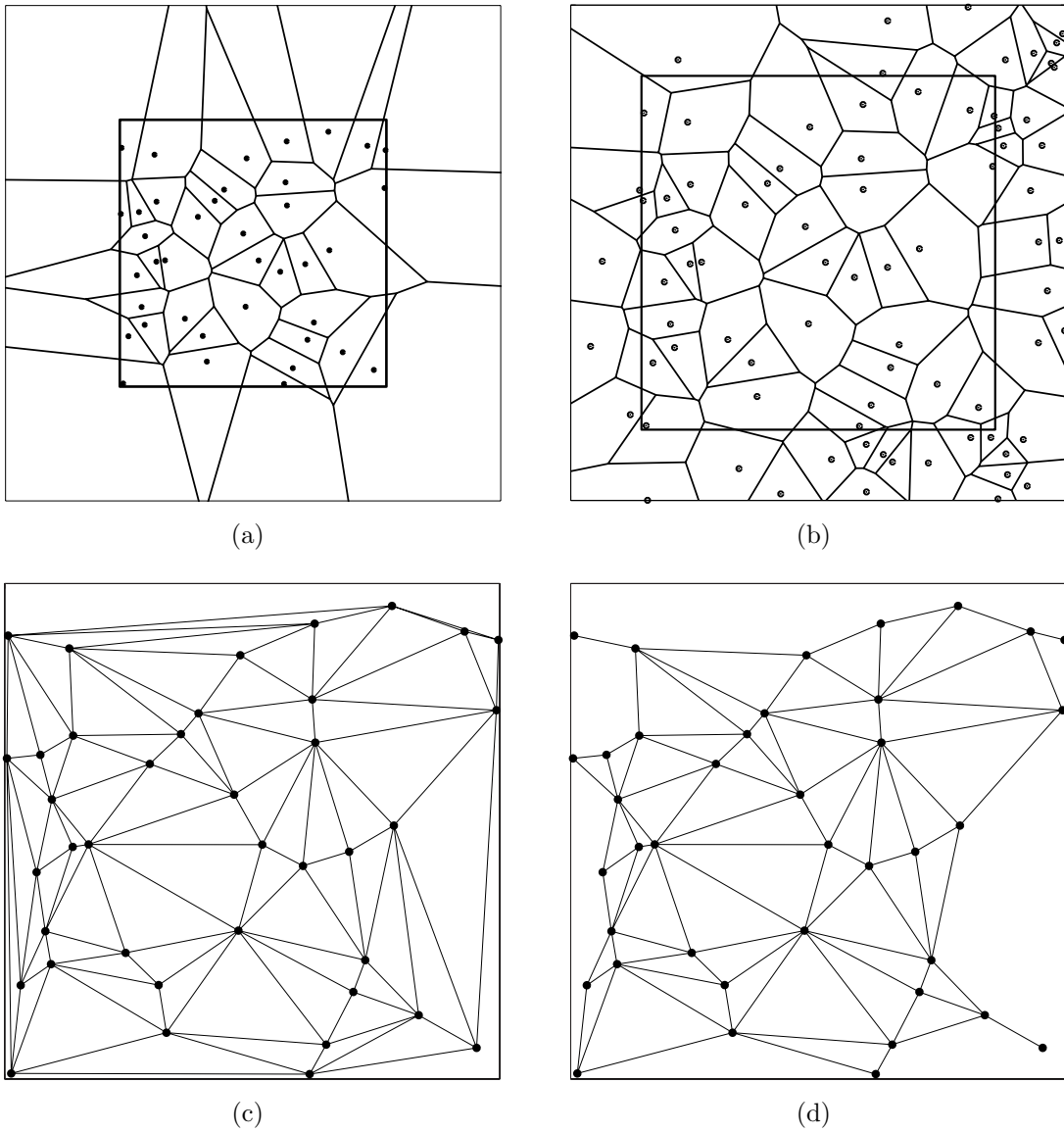


Figure 3.7: The problem of discovering neighboring node pairs in the Voronoi tessellation is shown in (a) and solution to this problem using external nodes is seen in (b). Corresponding graphs are given in (c) and (d), respectively.

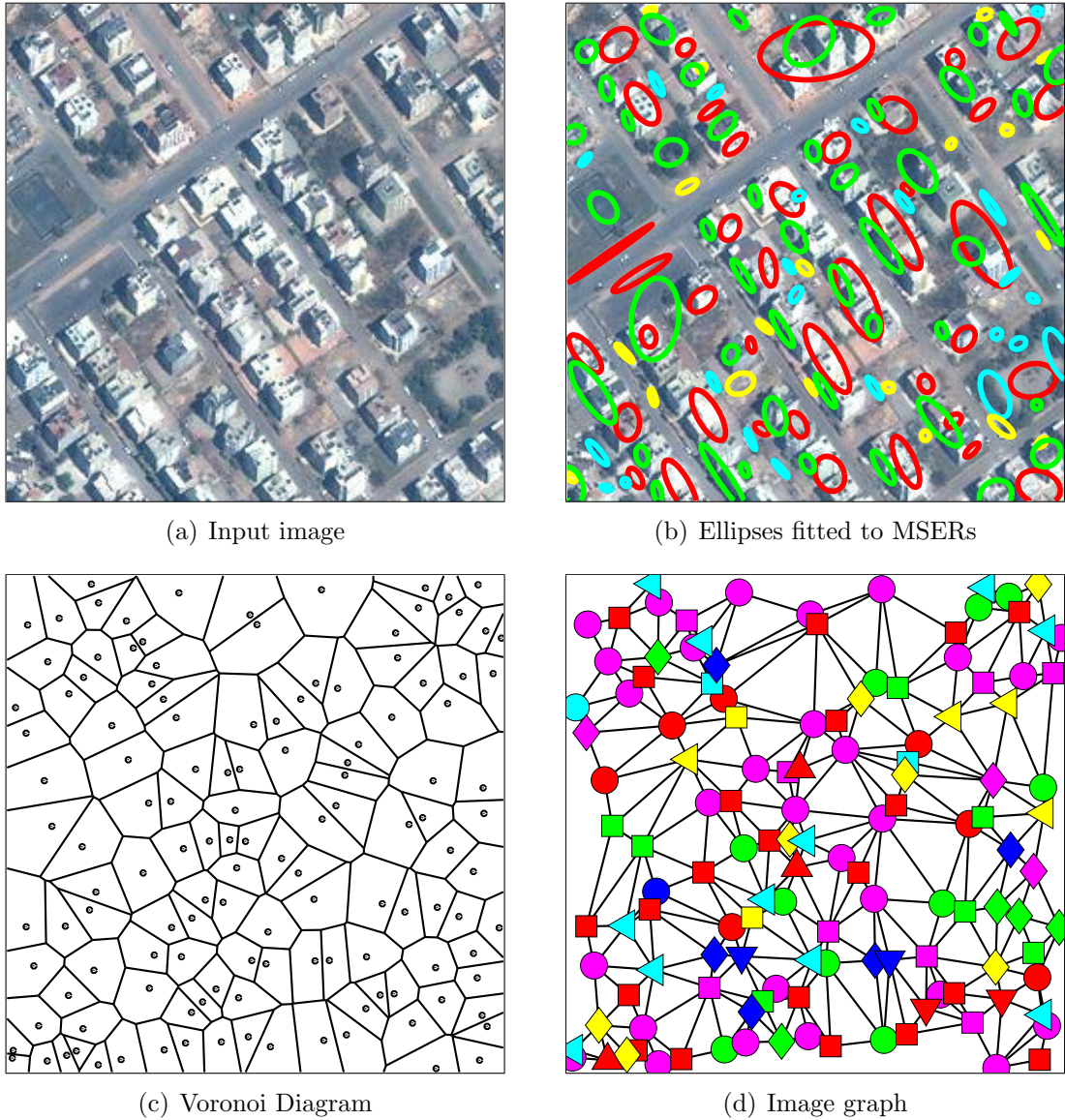


Figure 3.8: Graph construction steps. The color and shape of a node in (d) represent its label after k -means clustering.

Chapter 4

Graph Mining

11:15, restate my assumptions:

- 1. Mathematics is the language of nature.*
- 2. Everything around us can be represented and understood through numbers.*
- 3. If you graph these numbers, patterns emerge.*

Therefore, there are patterns everywhere in nature.

MAXIMILLIAN COHEN – FROM THE MOVIE π

At the end of previous chapter we manage to represent every image with a graph. Graphs are powerful in representing image content; however, their use for image classification has been limited due to inefficiencies in comparisons of these graphs for classification. All algorithmic tools for feature-based object representations can be available for graphs if they are embedded in vector spaces. For example, the *dissimilarity representation* [36] developed by Pekalska converts an input graph to feature vector with respect to a set of graph patterns called *prototypes*. The i th element of this vector is equal to the *graph edit distance* [37] between the input graph and the i th prototype. This method works quite well for matching relatively small graphs but it can become quite restrictive for very detailed image content with a large number of nodes and edges such as the graph in Figure 3.8(d). Furthermore, graph edit distance produces unreliable results

when the number of edit operations are too large and it is inefficient due to high computational complexity. Another graph embedding method is representing a graph as a *frequency vector* (histogram vector) for a given set of subgraphs [16]. The i th element of this vector is equal to the number of times (frequency) that the i th subgraph occurs in the input graph. The difficult part of this approach is to determine the subgraph set. For image classification, such a subgraph set should contain

1. Frequent graph patterns,
2. Discriminative graph patterns, and
3. Graph patterns having low redundancy.

The first criterion ensures that the subgraphs in the set can also be found in an unknown image graph. The second criterion guarantees the performance of classifiers, and the final criterion avoids redundancy that leads to the curse of dimensionality. To find the set satisfying these criteria, we propose a graph mining algorithm that first discovers frequent subgraphs from the image graph set, then discriminative subgraphs in the set are selected and finally redundant ones are removed from the set. We employ two methods from the literature for the first and third criteria, and develop a novel algorithm for mining discriminative patterns. The flowchart of the algorithm is displayed in Figure 4.1.

In this study we are dealing with image graphs but the subgraph-graph relation is analogous with term-document and symbol-string relations. Hence, the histogram vector method can also be extended for these relations, i.e. in the field of information retrieval. Therefore, we will use the term *pattern* to generalize subgraphs/terms/symbols in this chapter until Section 4.6. We first explain our data mining method in the following sections, then we specialize the method for graph mining.

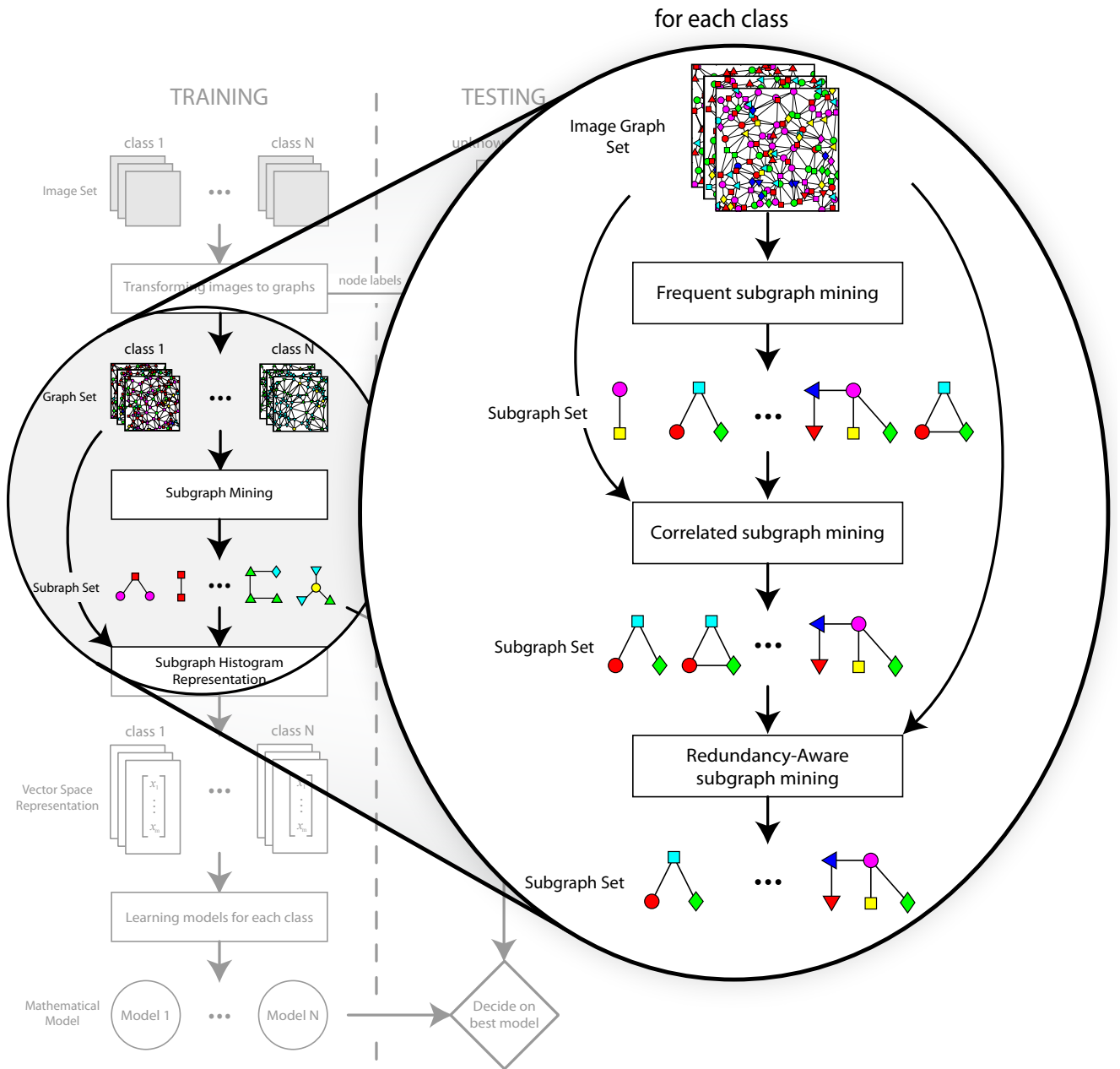


Figure 4.1: Steps of graph mining algorithm

4.1 Foundations of Pattern Mining

Before the details of the algorithm, we would like to give some background information about pattern mining. In this chapter we use a similar notation to Bringmann's in [10] and the definitions in this section are mainly taken from that study.

A definition of the task of finding all potentially interesting patterns is given by Mannila and Toivonen [30]. The result of a data mining task is defined as a theory depending on three parameters: a *pattern language* $\mathcal{L}$, a *dataset* $\mathcal{D}$, and a *selection predicate* ϕ .

Definition 4.1 (Theory of ϕ with respect to $\mathcal{L}$ and $\mathcal{D}$, [30]).

Assume a dataset $\mathcal{D}$, a pattern language $\mathcal{L}$ for expressing properties or defining subgroups of the data, and a selection predicate ϕ are given. The predicate ϕ is used for evaluating whether a pattern $\pi \in \mathcal{L}$ defines a potentially interesting subclass of $\mathcal{D}$. The task of finding the theory of $\mathcal{D}$ with respect to $\mathcal{L}$ and ϕ is defined as

$$Th(\mathcal{L}, \mathcal{D}, \phi) = \{\pi \in \mathcal{L} \mid \phi(\pi, \mathcal{D}) \text{ is true}\} \quad (4.1)$$

In our problem, the selection predicate ϕ is true if the pattern π is frequent, discriminative and not redundant for the dataset $\mathcal{D}$. We continue our definitions with the *matching function*. Many graph mining researchers define matching function as whether given subgraph occurs in example graph or not as in [10]. However, our study requires the number of times that a pattern occurs in an example (Details will be given in the following sections). Therefore, we define the matching function differently as follows.

Definition 4.2 (Matching Function).

Assume a pattern language $\mathcal{L}$, a dataset $\mathcal{D}$, and an evaluation predicate φ is given. The number of *valid occurrences* of pattern π in $x \in \mathcal{D}$ is defined as $\text{match} : \mathcal{L} \times \mathcal{D} \rightarrow \mathbb{Z}_+^0$ such that

$$\text{match}(\pi, x, \varphi) = |\{h \mid h(\pi) \subseteq x \wedge \varphi(h, \pi, x) \text{ is true}\}| \quad (4.2)$$

where h is called a mapping of pattern π into example x .

We use the terms *valid occurrences* and *mapping* in this definition instead of simply saying the number of occurrences of π in x because the occurrence of graph patterns in other graphs needs additional evaluations than term or symbol patterns. We will describe some evaluation predicates for graph patterns in Section 4.6; until there omitting the parameter φ from $\text{match}(\pi, x, \varphi)$ for simplicity, we have $\text{match}(\pi, x)$ as an equivalent to the former.

The frequency vector described in the introductory paragraph of this chapter is called propositionalization and defined below.

Definition 4.3 (Propositionalization, [10]).

Given a set of n patterns $\mathcal{S} = \{\pi_1, \dots, \pi_n\}$, we define the feature vector of an example x as

$$\vec{f}_{\mathcal{S}}(x) = (\text{match}(\pi_1, x), \dots, \text{match}(\pi_n, x))^{\top}. \quad (4.3)$$

Total number of valid occurrences of a pattern in a dataset is called *support* of that pattern. Again, we drop the evaluation predicate φ for the support definition.

Definition 4.4 (Support).

Given a pattern language $\mathcal{L}$ and a dataset $\mathcal{D}$, support of a pattern π in $\mathcal{D}$ is defined as

$$\text{supp}(\pi, \mathcal{D}) = \sum_{x \in \mathcal{D}} \text{match}(\pi, x). \quad (4.4)$$

And, our last definition in this section is *frequency*.

Definition 4.5 (Frequency).

Given a pattern language $\mathcal{L}$, a dataset $\mathcal{D}$, the frequency of a pattern π in $\mathcal{D}$ is defined as

$$\text{freq}(\pi, \mathcal{D}) = \frac{\text{supp}(\pi, \mathcal{D})}{|\mathcal{D}|}. \quad (4.5)$$

4.2 Frequent Pattern Mining

Our graph mining algorithm starts with discovering frequent patterns in the dataset. Frequent patterns have broad application areas such as association

rule mining, indexing, clustering and classification [14]. We are interested in the usefulness of frequent patterns in classification. Frequent pattern mining was extensively studied in the data mining community and numerous algorithms have been proposed in domains of different pattern types.

Assume dataset $\mathcal{D}$ is a set of examples where each example is labeled by one class in a domain of classes $\mathcal{C}$. The set of examples labeled by the class c is denoted by $\mathcal{D}_c$. In notation, we can say $\mathcal{D} = \bigcup_{i \in \mathcal{C}} \mathcal{D}_i$. The problem of frequent pattern discovery for class c can be formulated as finding all patterns generated by the pattern language $\mathcal{L}$, whose support in dataset $\mathcal{D}_c$ is greater than a threshold θ_c^{supp} . The set of all frequent patterns for class c is

$$\mathcal{F}_c = \{\pi \in \mathcal{L} \mid \text{supp}(\pi, \mathcal{D}_c) \geq \theta_c^{\text{supp}}\}. \quad (4.6)$$

Assume we try to find frequent patterns in $\mathcal{D}_c$. Let examples labeled by class c be the set of positive examples, denoted by $\mathcal{D}_+$ and the set of all other examples labeled by other classes be the set of negative examples, denoted by $\mathcal{D}_-$. Some frequent pattern mining applications limit the support of frequent patterns in negative set. In set definition, it is given by

$$\mathcal{F}_c = \{\pi \in \mathcal{L} \mid \text{supp}(\pi, \mathcal{D}_+) \geq \theta_+^{\text{supp}} \wedge \text{supp}(\pi, \mathcal{D}_-) \leq \theta_-^{\text{supp}}\}. \quad (4.7)$$

Since the size and property of a dataset varies among classes we can set different thresholds for each class. In our study, we mine the set of frequent patterns for each class, then they will be used as input to the next step which is correlated pattern mining.

4.3 Class Correlated Pattern Mining

Our second objective for the selected patterns is being discriminating for classification. Assume a dataset $\mathcal{D}$ labeled with three classes such as $\mathcal{D}_1$, $\mathcal{D}_2$ and $\mathcal{D}_3$. A discriminative pattern is expected to be found in examples of one class and not in the other classes. Such a relation between a pattern and a class is called

correlation. Two types of correlation have been defined: Positive and negative correlation. Assume we focus on finding correlated patterns of the first class. A pattern observed in only the first class, $\mathcal{D}_1$, is called positively correlated pattern with the first class or shortly *class-correlated pattern*. In contrast, a negatively correlated pattern is found in all classes except the first class (in this case $\mathcal{D}_2$ and $\mathcal{D}_3$). Both positive and negative correlation can be useful for classification. For example, an unseen example can be assigned to the first class if it includes positively correlated patterns with the first class. On the other hand, observation of negatively correlated patterns with the first class in an unseen example indicates that the example does not belong to the first class. Previous works on class-correlated patterns in [10, 33] only involve two-class datasets and the study in [34] handles multi-class correlation. However, all previous methods on correlated pattern mining base on binary matching function. Hence, more than one occurrences of a pattern in an example have the same effect with a single occurrence on correlation measure. We propose a novel technique to mine multi-class correlated patterns, in which the number of occurrences are taken into consideration. Unlike previous works, we are only dealing with positively correlated patterns and do not consider negatively correlated patterns. The following subsection explains the mathematical modeling of patterns in examples and the second subsection gives details of mining correlated patterns according to the model.

4.3.1 Mathematical Modeling of Pattern Support

We start our algorithm by deriving the probability that a pattern π occurs k times in an example. Let x be a document containing N_t terms and the probability that a randomly selected term t from x is an instance of the pattern π be a *Bernoulli distribution* with $\Pr\{T = \pi\} = p$ and $\Pr\{T \neq \pi\} = 1 - \Pr\{T = \pi\} = 1 - p$. Then, the probability that x contains π for k times is a *Binomial distribution* with

$$\Pr\{K = k\} = \binom{N_t}{k} p^k (1 - p)^{N_t - k} \quad \text{for } k = 0, 1, 2, \dots, N_t. \quad (4.8)$$

We now define the expected number of occurrences as $\lambda = pN_t$. We generalize

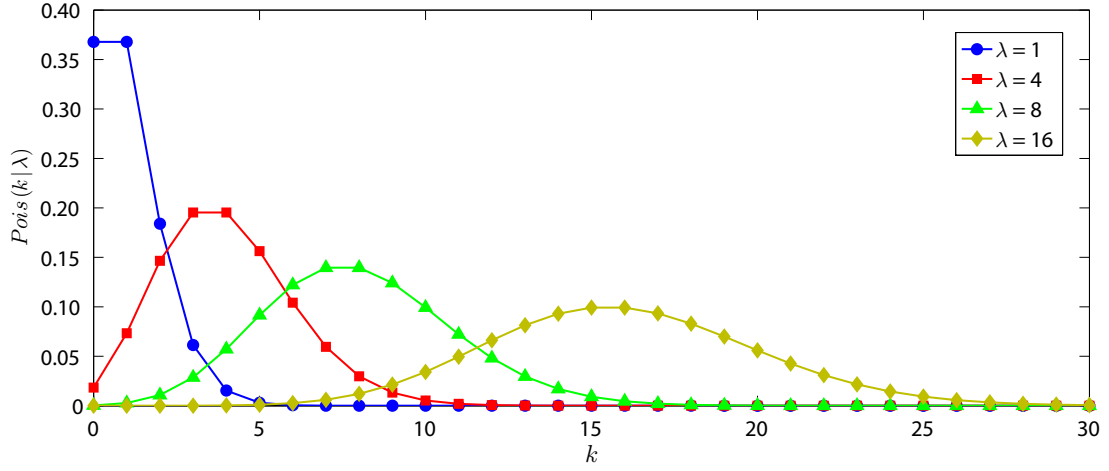


Figure 4.2: Poisson distributions with four different expected values.

this probability by assuming that the size of document is unbounded, then

$$\Pr\{K = k\} = \lim_{N_t \rightarrow \infty} \binom{N_t}{k} p^k (1-p)^{N_t-k} = \frac{\lambda^k e^{-\lambda}}{k!} \quad \text{for } k \in \mathbb{Z}_+^0. \quad (4.9)$$

The limiting case of Binomial distribution is known as *Poisson distribution* and is denoted by $Pois(k | \lambda)$. Poisson distributions with different expected values are shown in Figure 4.2.

We model the term frequency in a document as a Poisson distribution. However, Church and Gale claim in [15] that term rates vary from author-to-author, topic-to-topic, document-to-document, section-to-section, and paragraph-to-paragraph (and theme-to-theme if we consider images). They propose *Poisson mixture* to capture much of this heterogeneous structure by allowing the Poisson parameter λ to vary over documents. It is subject to a density function ϑ aimed to capture dependencies on hidden variables such author, topic, etc.

Definition 4.6 (Poisson Mixtures, [15]).

Given the density function ϑ to capture dependencies on hidden variables, the general form of a Poisson mixture is

$$p(k) = \int_0^\infty \vartheta(\lambda) Pois(k | \lambda) d\lambda \quad (4.10)$$

where ϑ density function should integrate to 1. That is, $\int_0^\infty \vartheta(\lambda) d\lambda = 1$.

In this study we will use a mixture of finite number of Poisson distributions. Therefore, we use coefficients α_j instead of ϑ function:

$$p(k) = \sum_{j=1}^m \alpha_j \text{Pois}(k | \lambda_j) \quad (4.11)$$

where $\sum_{j=1}^m \alpha_j = 1$ such that $\alpha_j \geq 0$ for $j = 1, \dots, m$.

The next step of modeling is parameter estimation for Poisson mixtures from a dataset. The *Expectation-Maximization(EM) Algorithm* can be used for the solution to the maximum-likelihood parameter estimation problem. The EM algorithm is exhaustively described in [6] for Gaussian mixture model and we will extend this study for Poisson mixture model.

We have a density function $p(k | \Theta)$ governed by the set of parameters $\Theta = (\alpha_1, \dots, \alpha_m, \lambda_1, \dots, \lambda_m)$ and a dataset of size n , supposedly drawn independently from this distribution, i.e. $\mathcal{K} = \{k_1, \dots, k_n\}$. The resulting density for samples is

$$p(\mathcal{K} | \Theta) = \prod_{i=1}^n p(k_i | \Theta) = L(\Theta | \mathcal{K}) \quad (4.12)$$

where $L(\Theta | \mathcal{K})$ is called the likelihood function. The goal of EM algorithm is to find Θ^* iteratively where

$$\Theta^* = \arg \max_{\Theta} L(\Theta | \mathcal{K}). \quad (4.13)$$

We assume that a complete data set exists as a combination of the observed but incomplete data $\mathcal{K}$ and the missing data $\mathcal{Z}$. The EM algorithm seeks to find the maximum likelihood estimate (MLE) of the marginal likelihood by iteratively applying the following two steps:

Expectation step: The expected value of the log likelihood function is calculated with respect to the unknown data $\mathcal{Z}$ given the observed data $\mathcal{K}$ and the current parameter estimates $\Theta^{(i-1)}$.

$$Q(\Theta, \Theta^{(i-1)}) = \mathbb{E}[\log p(\mathcal{K}, \mathcal{Z} | \Theta) | \mathcal{K}, \Theta^{(i-1)}]. \quad (4.14)$$

Maximization step: The expectation can be maximized by finding optimum values for the new parameters Θ as

$$\Theta^{(i)} = \arg \max_{\Theta} Q(\Theta, \Theta^{(i-1)}). \quad (4.15)$$

We can maximize Q with respect to the two sets of parameters α_j and λ_j , independently. Fortunately, the estimates for these parameters are same as Gaussian mixture model (estimate for λ of Poisson distribution is the same with estimate for μ of Gaussian distribution).

The estimate for α_j can be computed as

$$\hat{\alpha}_j = \frac{1}{n} \sum_{i=1}^n p(j | k_i, \Theta^{(g)}) \quad (4.16)$$

where

$$p(j | k_i, \Theta^{(g)}) = \frac{\alpha_j^{(g)} \text{Pois}(k_i | \lambda_j^{(g)})}{\sum_{t=1}^m \alpha_t^{(g)} \text{Pois}(k_i | \lambda_t^{(g)})}. \quad (4.17)$$

Equating the partial derivative of $Q(\Theta, \Theta^{(g)})$ with respect to λ_j to zero gives

$$\hat{\lambda}_j = \frac{\sum_{i=1}^n p(j | k_i, \Theta^{(g)}) k_i}{\sum_{i=1}^n p(j | k_i, \Theta^{(g)})}. \quad (4.18)$$

These steps continue until the change in log-likelihood between two iterations is less than a threshold or the number of iterations reaches a limit.

Corollary 4.1 (Relationship between weighted average and sample mean).

After each iteration of the EM algorithm, the weighted average equals to the sample mean:

$$\sum_{j=1}^m \hat{\alpha}_j \hat{\lambda}_j = \frac{1}{n} \sum_{i=1}^n k_i. \quad (4.19)$$

Proof. Putting (4.16) and (4.18) into (4.19) gives

$$\begin{aligned} \sum_{j=1}^m \hat{\alpha}_j \hat{\lambda}_j &= \sum_{j=1}^m \left(\frac{\sum_{i=1}^n p(j | k_i, \Theta^{(g)})}{n} \times \frac{\sum_{i=1}^n p(j | k_i, \Theta^{(g)}) k_i}{\sum_{i=1}^n p(j | k_i, \Theta^{(g)})} \right) \\ &= \frac{1}{n} \sum_{i=1}^n k_i \sum_{j=1}^m p(j | k_i, \Theta^{(g)}) = \frac{1}{n} \sum_{i=1}^n k_i. \end{aligned} \quad (4.20)$$

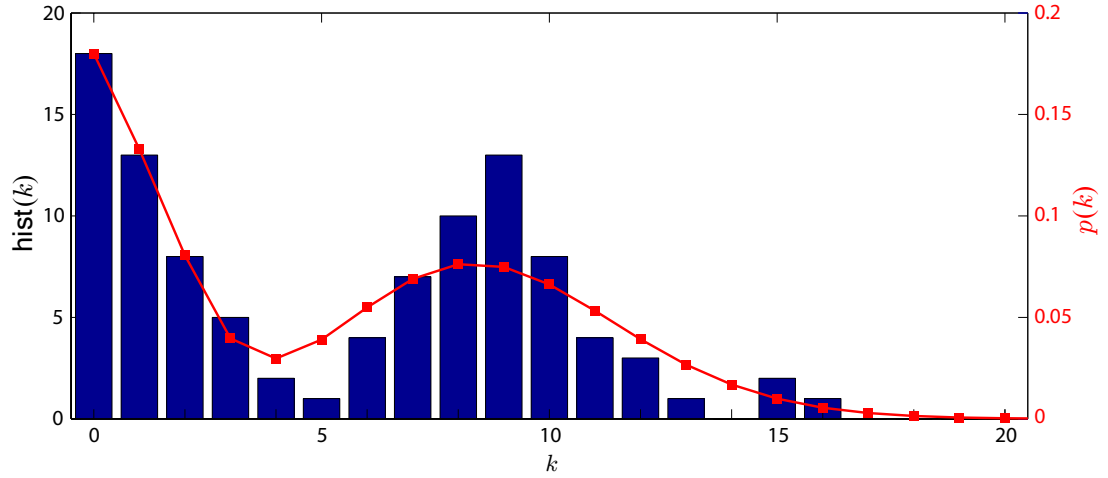


Figure 4.3: A sample histogram of a dataset with 100 elements and fitting mixtures of 3 Poisson distributions to this histogram are shown in blue and red, respectively.

This shows the equality between weighted average and sample mean. $\square$

A sample histogram and the Poisson mixture trained on this data can be seen in Figure 4.3.

Returning to our problem, we employ Poisson mixture to model the probability of the term frequency of pattern π in a document randomly selected from the corpus $\mathcal{D}$. Thus, the dataset used in the EM algorithm is $\mathcal{K} = \{k_1, \dots, k_n\} = \{\text{match}(\pi, x_1), \dots, \text{match}(\pi, x_n)\}$ for $\mathcal{D} = \{x_1, \dots, x_n\}$.

4.3.2 Correlated Patterns

In this section we give details of correlated-pattern mining according the probability calculations drawn in the previous section. A pattern π is called correlated with a class $c \in \mathcal{C}$ if that pattern is frequently observed in $\mathcal{D}_c$ while it is seldom seen in $\mathcal{D}_{c'}$ for all $c' \in \mathcal{C}$ such that $c' \neq c$. Using the support of a pattern in correlation measure may result in wrong classification of unknown documents because the support in a dataset does not give any frequency information of the pattern in individual documents. For example, a pattern may occur frequently

in some documents of a dataset while not occurring in other documents. Therefore, frequency analysis of a pattern should be done using the frequency in each document as in the previous section.

Assume a pattern π which does not occur in a dataset $\mathcal{D}$, in notational form $\text{supp}(\pi, \mathcal{D}) = 0$, then training the EM algorithm on such a case produces the following density function p_{ref} , which we call the *reference distribution*,

$$p_{\text{ref}}(k) = \begin{cases} 1 & \text{if } k = 0 \\ 0 & \text{otherwise.} \end{cases} \quad (4.21)$$

As understood from its name we employ this density function for comparison with other distributions in correlation measurement. Let $p_c(k | \pi, \mathcal{D}_c)$ denote the Poisson mixture distribution trained for pattern π on dataset $\mathcal{D}_c$. We compute the distance (dissimilarity) between density functions $p_c(k | \pi, \mathcal{D}_c)$ and $p_{\text{ref}}(k)$ using the *Earth mover's distance* technique.

In terms of positive correlation between pattern π and class c , the distance between $p_c(k | \pi, \mathcal{D}_c)$ and $p_{\text{ref}}(k)$ should be as large as possible while the distance between $p_{c'}(k | \pi, \mathcal{D}_{c'})$ and $p_{\text{ref}}(k)$ should be as small as possible for all $c' \in \mathcal{C}$ such that $c' \neq c$.

Earth mover's distance is proposed by [38] to measure the dissimilarity between not only probability distributions but also histograms and clusters. Levina and Bickel [27] showed that Earth mover's distance is conceptually equivalent to *Mallow distance* on probability distributions but we continue calling it the Earth mover's distance in this study.

Definition 4.7 (Earth mover's distance, [38]).

Let $Q = \{(q_1, w_{q_1}), \dots, (q_r, w_{q_r})\}$ be the signature with r clusters, where q_i is the representative of the i th data cluster and w_{q_i} is the number of points in the cluster. Let $Q' = \{(q'_1, w_{q'_1}), \dots, (q'_s, w_{q'_s})\}$ be the second signature with s clusters; and $\mathbf{D} = [d_{ij}]$ is the ground distance matrix where d_{ij} is some measure of distance between clusters of q_i and q'_j . Earth mover's distance is computed by solving the optimization problem of finding a flow $\mathbf{F} = [f_{ij}]$, with f_{ij} the flow from q_i to q'_j ,

that minimizes the overall work

$$\text{Minimize} \quad \text{WORK}(Q, Q', \mathbf{F}) = \sum_{i=1}^r \sum_{j=1}^s d_{ij} f_{ij} \quad (4.22)$$

$$\text{Subject to} \quad f_{ij} \geq 0, \quad 1 \leq i \leq r, \quad 1 \leq j \leq s, \quad (4.23)$$

$$\sum_{j=1}^s f_{ij} \leq w_{q_i}, \quad 1 \leq i \leq r, \quad (4.24)$$

$$\sum_{i=1}^r f_{ij} \leq w_{q'_j}, \quad 1 \leq j \leq s, \quad (4.25)$$

$$\sum_{i=1}^r \sum_{j=1}^s f_{ij} = \min \left(\sum_{i=1}^r w_{q_i}, \sum_{j=1}^s w_{q'_j} \right). \quad (4.26)$$

Once the optimal flow f_{ij}^* is found, the Earth Mover's distance between Q and Q' is defined as

$$\text{EMD}(Q, Q') = \frac{\sum_{i=1}^r \sum_{j=1}^s f_{ij}^* d_{ij}}{\sum_{i=1}^r \sum_{j=1}^s f_{ij}^*}. \quad (4.27)$$

—

According to (4.24) and (4.25), we can transform a discrete distribution $p(k)$ defined in $\mathbb{Z}_+^0$ to the signature form by

$$P = \{(k, p(k)) \mid k \in \mathbb{Z}_+^0 \wedge p(k) > 0\}. \quad (4.28)$$

The signature size of a Poisson mixture may be infinite, or the signature may be reduced to finite elements if the condition in (4.28) is changed to $p(k) > \epsilon$ for a small number ϵ .

From (4.28), the signature form of the distribution $p_{\text{ref}}(k)$ becomes $P_{\text{ref}} = \{(q'_1, w_{q'_1})\} = \{(0, 1)\}$. Let P_c denote the signature form of probability distribution $p_c(k \mid \pi, \mathcal{D}_c)$, then the optimal flow from P_c to P_{ref} is

$$f_{ij}^* = \begin{cases} p_c(k_i \mid \pi, \mathcal{D}_c) & \text{if } j = 1 \\ 0 & \text{otherwise.} \end{cases} \quad (4.29)$$

Constraint (4.26) is actually equal to $\sum_{i=1}^r \sum_{j=1}^s f_{ij} = 1$ for probability distributions and this constraint forces all earth to flow from every q_i to the only element of P_{ref} , which is q'_1 . Finally, we need to define a ground distance function, $d : \mathbb{Z}^2 \rightarrow \mathbb{R}_+^0$, for the ground distance matrix $\mathbf{D} = [d_{ij}] = [d(q_i, q'_j)]$. Then, the Earth mover's distance is computed as

$$\begin{aligned}
 \text{EMD}(P_c, P_{\text{ref}}) &= \frac{\sum_{i=1}^r \sum_{j=1}^s f_{ij}^* d_{ij}}{\sum_{i=1}^r \sum_{j=1}^s f_{ij}^*} \\
 &= \sum_{i=1}^r p_c(k_i | \pi, \mathcal{D}_c) d(k_i, 0) \\
 &= \sum_{i=1}^r p_c(k_i | \pi, \mathcal{D}_c) d'(k_i) \\
 &= \mathbb{E}[d'(K) | \pi, \mathcal{D}_c]
 \end{aligned} \tag{4.30}$$

where $d'(k) = d(k, 0)$ and K is a random variable $K \in \{k_1, \dots, k_r\}$ or more generally $K \sim \mathbb{Z}_+^0$. One possible definition of the distance function for two distributions is $d(i, j) = 1 - e^{-\xi|i-j|}$ where ξ is a regulation constant. Note that $d(i, j) \in [0, 1]$, and

$$d'(i) = 1 - e^{-\xi i} \quad \text{for } i \geq 0. \tag{4.31}$$

Accordingly, the Earth mover's distance is in the same range $[0, 1]$ due to definition of distance function. One reason why we choose this nonlinear function is to prevent outliers in the dataset from dominating the Earth mover's distance. The regulation constant ξ is used for this purpose.

Assume we try to find correlated patterns of class c . Each frequent pattern $\pi \in \mathcal{F}_c$ mined in the previous section should be tested by a correlation measure γ based on the Earth mover's distance. The correlation measure that we define has two parameters: positive distance p_π^c and negative distance n_π^c . The positive distance is the Earth mover's distance between P_c and P_{ref} for dataset $\mathcal{D}_c$. On the other hand, the negative distance is computed from datasets $\mathcal{D}_{c'}$ for all $c' \in \mathcal{C}$ such that $c' \neq c$. In terms of classification, the class which will cause confusion with class c is the one with the maximum Earth mover's distance between $P_{c'}$ and P_{ref} from all $c' \in \mathcal{C}$ such that $c' \neq c$. Thus, the positive and negative distance

used in correlation measure γ is given by

$$p_\pi^c = \mathbb{E}[d'(K) \mid \pi, \mathcal{D}_c], \quad n_\pi^c = \max_{c' \neq c, c' \in \mathcal{C}} \mathbb{E}[d'(K) \mid \pi, \mathcal{D}_{c'}]. \quad (4.32)$$

The computation procedure of positive and negative distances are illustrated in Figure 4.4 for four classes. As shown in the figure, for each class the probability distribution is computed by the EM algorithm and the Earth mover's distance is computed between this distribution and the reference distribution. Then, the distance computed for the interest class c is assigned p_π^c and the greatest distance computed for the other classes is assigned to n_π^c .

We use p_π^c and n_π^c as parameters to measure the correlation between the pattern π and the class c . The correlation measures commonly used in the literatures are chi-square (χ^2) test and information gain, which are computed from a contingency table. Instead, we derive a correlation function γ from the χ^2 test as follows without indices:

$$\gamma(p, n) = \frac{(p - n)|p - n|}{(p + n)(2 - p - n)}. \quad (4.33)$$

The range of γ function is $[-1, 1]$. Positive values of γ for pattern π imply positive correlation and negative values indicate negative correlation. The correlation function γ can be seen in Figure 4.5.

Similar to frequent pattern mining, correlated pattern mining is defined as finding all patterns $\pi \in \mathcal{L}$ whose correlation with $\mathcal{D}_c$ is greater than a threshold θ_c^{cor} , that is $\gamma(p_\pi^c, n_\pi^c) \geq \theta_c^{\text{cor}}$. According to our objectives for pattern mining, we check correlation of only frequent patterns for each class $c \in \mathcal{C}$. To decrease computational cost we can define a lower bound for support threshold θ_c^{supp} in the previous section with respect to correlation threshold θ_c^{cor} . The rest of this section is devoted to the calculations of this relation.

Lemma 4.1 (Lower and upper bounds of γ function).

The lower and upper bounds of $\gamma(p, n)$ is given by

$$\min_{p \geq p', n \leq n'} \gamma(p, n) = \max_{p \leq p', n \geq n'} \gamma(p, n) = \gamma(p', n'). \quad (4.34)$$

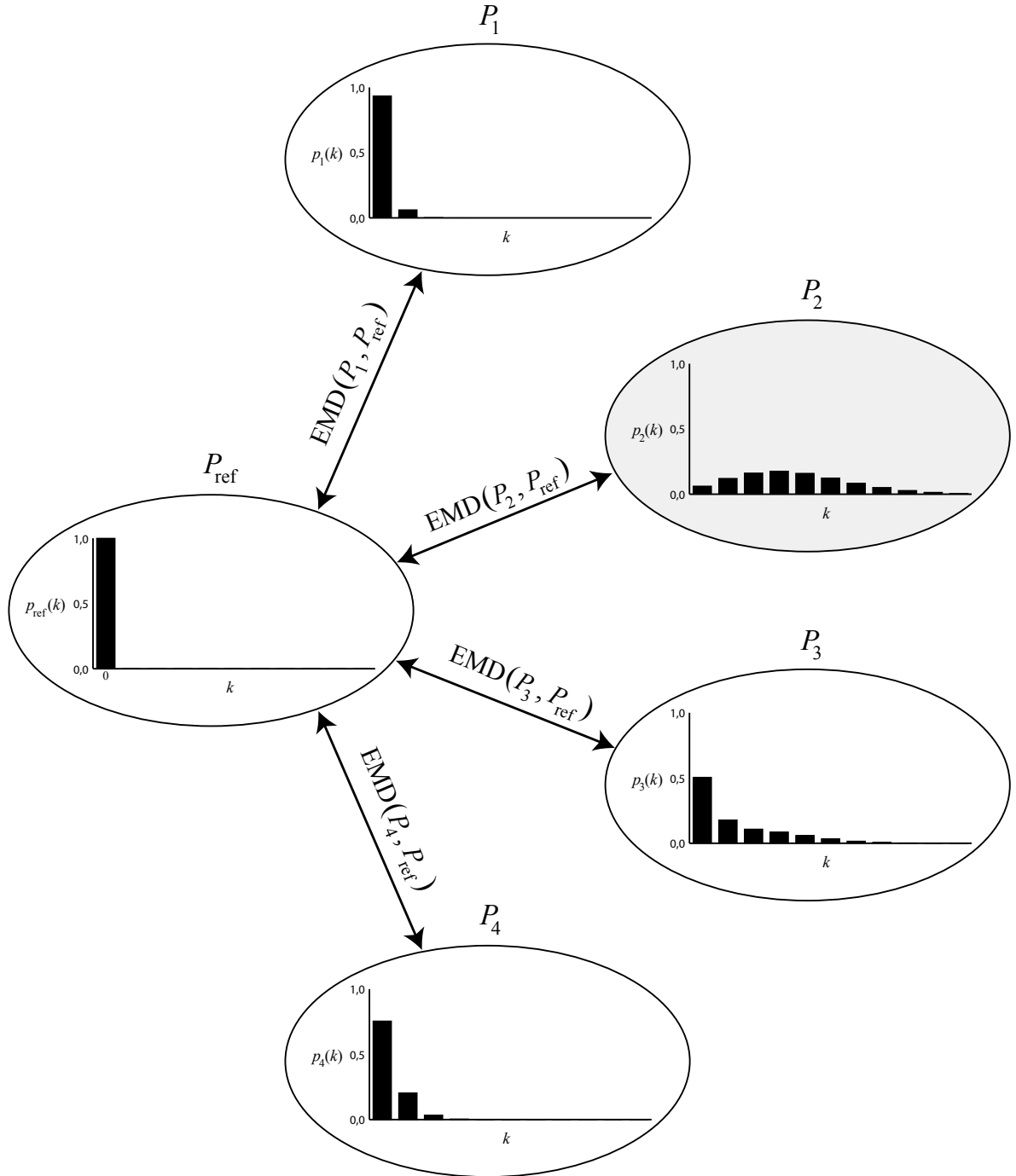
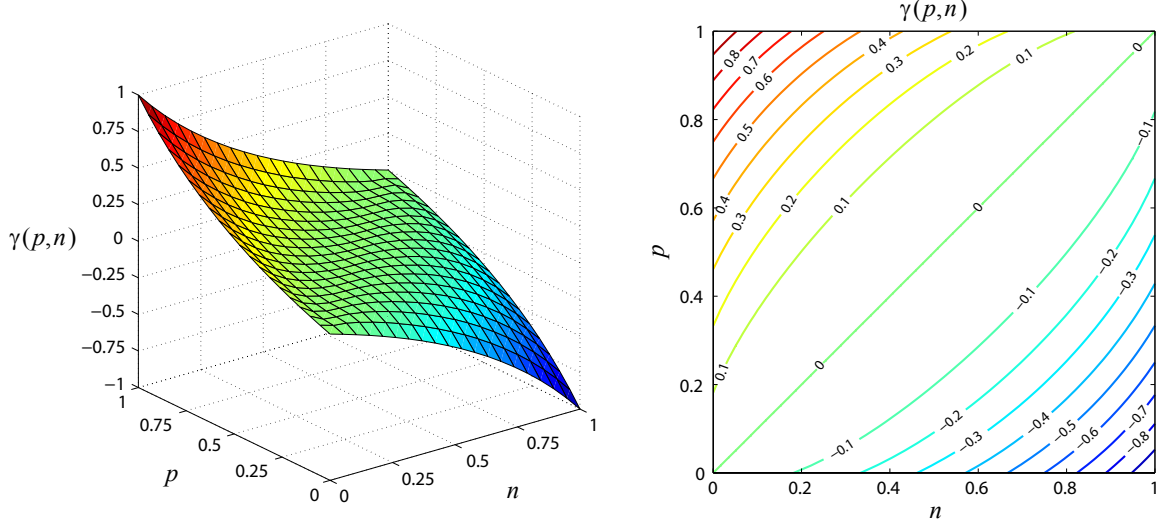


Figure 4.4: The procedure for positive and negative distance computation is illustrated for four classes. The interest class is the second one and the distances are computed as $p = \text{EMD}(P_2, P_{\text{ref}})$ and $n = \text{EMD}(P_3, P_{\text{ref}})$.

Figure 4.5: The correlation function $\gamma(p, n)$

Proof. Partial derivatives give the limits of $\gamma(p, n)$.

$$\begin{aligned} \frac{\partial \gamma(p, n)}{\partial p} &= \frac{2(p - 2pn + 3n - 2n^2)|p - n|}{(p + n)^2(2 - p - n)^2} \\ &= \frac{2(p(1 - n) + n(3 - p - 2n))|p - n|}{(p + n)^2(2 - p - n)^2}. \end{aligned} \quad (4.35)$$

Due to definition of $d(i, j)$, the pn space is defined for $0 \leq p, n \leq 1$. Hence, all terms in (4.35) are nonnegative. As a result, $\frac{\partial \gamma(p, n)}{\partial p} \geq 0$ and we can say

$$\gamma(p, n) \geq \gamma(p', n) \quad \text{for } 0 \leq p' \leq p \leq 1. \quad (4.36)$$

Similarly,

$$\begin{aligned} \frac{\partial \gamma(p, n)}{\partial n} &= -\frac{2(3p - 2p^2 - 2pn + n)|p - n|}{(p + n)^2(2 - p - n)^2} \\ &= -\frac{2(p(3 - 2p - n) + n(1 - p))|p - n|}{(p + n)^2(2 - p - n)^2}. \end{aligned} \quad (4.37)$$

All terms in (4.37) are nonnegative. Consequently, $\frac{\partial \gamma(p, n)}{\partial n} \leq 0$ and we can also say

$$\gamma(p, n) \geq \gamma(p, n') \quad \text{for } 0 \leq n \leq n' \leq 1. \quad (4.38)$$

Combining (4.36) and (4.38) gives (4.34). $\square$

Theorem 4.1 (Correlation–Positive support relation).

For a given pattern $\pi \in \mathcal{L}$, if $\gamma(p_\pi^c, n_\pi^c) \geq \theta_c^{\text{cor}}$ for some thresholds $\theta_c^{\text{cor}} \geq 0$ where p_π^c and n_π^c are computed from $\mathcal{D}_c$ and $\mathcal{D}_{c'}$ for all $c' \in \mathcal{C}$ such that $c' \neq c$, respectively, then

$$\text{supp}(\pi, \mathcal{D}_c) \geq -\frac{|\mathcal{D}_c|}{1 - e^{-\xi}} \ln \left(\frac{1 - \theta_c^{\text{cor}}}{1 + \theta_c^{\text{cor}}} \right). \quad (4.39)$$

Proof. According to Lemma 4.1, we can say

$$\gamma(p_\pi^c, 0) \geq \gamma(p_\pi^c, n_\pi^c) \geq \theta_c^{\text{cor}}. \quad (4.40)$$

Putting (4.33) into (4.40),

$$\frac{p_\pi^c}{2 - p_\pi^c} \geq \theta_c^{\text{cor}} \quad \Rightarrow \quad p_\pi^c \geq \frac{2\theta_c^{\text{cor}}}{1 + \theta_c^{\text{cor}}}. \quad (4.41)$$

Now, we compute p_π^c from (4.32). Let the parameters of Poisson mixture be $\Theta = (\alpha_1, \dots, \alpha_m, \lambda_1, \dots, \lambda_m)$ found by the EM algorithm for $p_c(k | \pi, \mathcal{D}_c)$ on observed data $\mathcal{K} = \{\text{match}(\pi, x_1), \dots, \text{match}(\pi, x_n)\}$ where $\mathcal{D}_c = \{x_1, \dots, x_n\}$. The positive distance is computed as

$$\begin{aligned} p_\pi^c &= \mathbb{E}[d'(K) | \pi, \mathcal{D}_c] = \sum_{i=0}^{\infty} p_c(i | \pi, \mathcal{D}_c) d'(i) \\ &= \sum_{i=0}^{\infty} \sum_{j=1}^m \alpha_j \text{Pois}(i | \lambda_j) d'(i) \\ &= \sum_{i=0}^{\infty} \sum_{j=1}^m \alpha_j \frac{\lambda_j^i e^{-\lambda_j}}{i!} (1 - e^{-\xi i}) \\ &= \sum_{j=1}^m \alpha_j e^{-\lambda_j} \sum_{i=0}^{\infty} \left(\frac{\lambda_j^i}{i!} - \frac{(\lambda_j e^{-\xi})^i}{i!} \right) \\ &= \sum_{j=1}^m \alpha_j e^{-\lambda_j} \left(e^{\lambda_j} - e^{(\lambda_j e^{-\xi})} \right) \\ &= \sum_{j=1}^m \left(\alpha_j - \alpha_j e^{-\lambda_j(1-e^{-\xi})} \right) = 1 - \sum_{j=1}^m \alpha_j e^{-\lambda_j d'(1)}. \end{aligned} \quad (4.42)$$

Introducing $\varepsilon = d'(1)$ results in $p_\pi^c = 1 - \sum_{j=1}^m \alpha_j e^{-\varepsilon \lambda_j}$. Combining (4.41) and (4.42) gives

$$1 - \sum_{j=1}^m \alpha_j e^{-\varepsilon \lambda_j} \geq \frac{2\theta_c^{\text{cor}}}{1 + \theta_c^{\text{cor}}} \Rightarrow \sum_{j=1}^m \alpha_j e^{-\varepsilon \lambda_j} \leq \frac{1 - \theta_c^{\text{cor}}}{1 + \theta_c^{\text{cor}}}. \quad (4.43)$$

Before continuing the proof, we need to define *convex function* and *Jensen's inequality*.

Definition 4.8 (Convex function).

Let C be a convex subset of $\mathbb{R}^n$. A function $f : C \rightarrow \mathbb{R}$ is called convex if

$$f(ax_1 + (1-a)x_2) \leq af(x_1) + (1-a)f(x_2) \quad \text{for } a \in [0, 1] \text{ and } x_1 \neq x_2. \quad (4.44)$$

See Figure 4.6 for a sample convex function.

Theorem 4.2 (Jensen's inequality).

If f is a convex function and X is a random variable, then

$$f(\mathbb{E}[X]) \leq \mathbb{E}[f(X)]. \quad (4.45)$$

In finite form, numbers $x_1, x_2, \dots, x_n$ are in domain of f , and positive weights a_i such that $\sum_{i=1}^n a_i = 1$, Jensen's inequality can be stated as

$$f\left(\sum_{i=1}^n a_i x_i\right) \leq \sum_{i=1}^n a_i f(x_i). \quad (4.46)$$

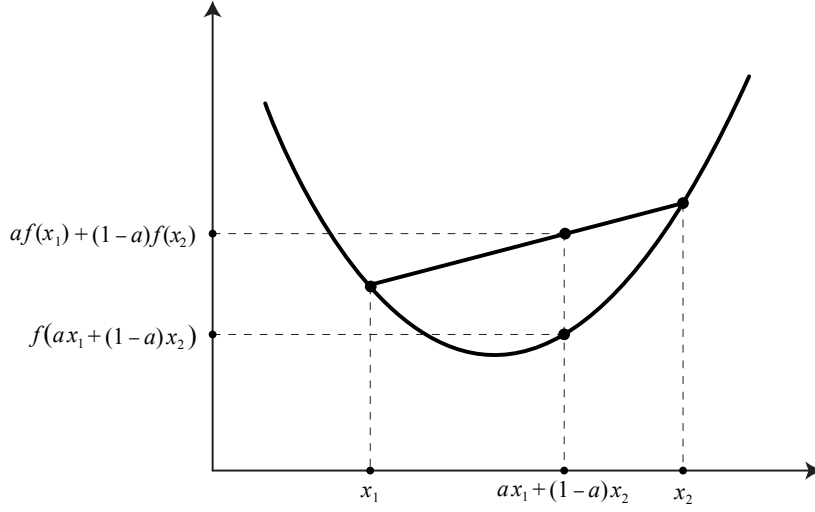
—

We define function $f : \mathbb{R} \rightarrow \mathbb{R}$ such as $f(x) = e^{-\varepsilon x}$. This function is strictly convex because $\frac{d^2 f(x)}{dx^2} = \varepsilon^2 e^{-\varepsilon x} > 0$. Now, we can extend (4.43) using (4.46) and f function as follows

$$\sum_{j=1}^m \alpha_j f(\lambda_j) \leq \frac{1 - \theta_c^{\text{cor}}}{1 + \theta_c^{\text{cor}}} \Rightarrow f\left(\sum_{j=1}^m \alpha_j \lambda_j\right) \leq \frac{1 - \theta_c^{\text{cor}}}{1 + \theta_c^{\text{cor}}}. \quad (4.47)$$

Substituting $f(x)$ with $e^{-\varepsilon x}$,

$$\Rightarrow \exp\left(-\varepsilon \sum_{j=1}^m \alpha_j \lambda_j\right) \leq \frac{1 - \theta_c^{\text{cor}}}{1 + \theta_c^{\text{cor}}} \Rightarrow -\varepsilon \sum_{j=1}^m \alpha_j \lambda_j \leq \ln\left(\frac{1 - \theta_c^{\text{cor}}}{1 + \theta_c^{\text{cor}}}\right). \quad (4.48)$$


 Figure 4.6: Plot of a convex function f

Using (4.19) and the definition of $\mathcal{K}$, we can say

$$\sum_{j=1}^m \alpha_j \lambda_j = \frac{1}{n} \sum_{i=1}^n k_i = \frac{1}{n} \sum_{i=1}^n \text{match}(\pi, x_i) = \frac{1}{n} \text{supp}(\pi, \mathcal{D}_c). \quad (4.49)$$

Finally, using (4.48) and (4.49), and substituting ε with $1 - e^{-\xi}$, and n with $|\mathcal{D}_c|$ makes

$$\text{supp}(\pi, \mathcal{D}_c) \geq -\frac{|\mathcal{D}_c|}{1 - e^{-\xi}} \ln \left(\frac{1 - \theta_c^{\text{cor}}}{1 + \theta_c^{\text{cor}}} \right). \quad (4.50)$$

This completes our proof. $\square$

The support threshold should be defined considering this relation. Likewise, the positive distance of a frequent pattern depends on the support threshold as follows.

Corollary 4.2 (Support threshold–positive distance relation).

Let $\pi \in \mathcal{L}$ be a frequent pattern whose support is greater than the threshold θ_c^{supp} for dataset $\mathcal{D}_c$, then

$$p_\pi^c \geq 1 - \exp \left(-\frac{\theta_c^{\text{supp}}}{|\mathcal{D}_c|} (1 - e^{-\xi}) \right). \quad (4.51)$$

Proof. Using same convex f function of the previous proof in (4.42), we get

$$\begin{aligned}
p_\pi^c &= 1 - \sum_{j=1}^m \alpha_j e^{-\lambda_j d'(1)} = 1 - \sum_{j=1}^m \alpha_j f(\lambda_j) \\
&\geq 1 - f\left(\sum_{j=1}^m \alpha_j \lambda_j\right) \\
&\geq 1 - \exp\left(-d'(1) \sum_{j=1}^m \alpha_j \lambda_j\right) \\
&\geq 1 - \exp\left(-\frac{d'(1)}{n} \text{supp}(\pi, \mathcal{D}_c)\right) \\
&\geq 1 - \exp\left(-\frac{(1 - e^{-\xi}) \theta_c^{\text{supp}}}{|\mathcal{D}_c|}\right).
\end{aligned} \tag{4.52}$$

□

4.4 Redundancy-Aware Top- k Patterns

After two steps of pattern mining, we have a set of frequent and class-correlated patterns. Usage of this set for classification suffers from curse of dimensionality because the set contains many redundant patterns and its size is not limited. Let the graph patterns in Figure 4.7 be frequent and correlated patterns. As seen in the figure, the only difference between patterns is an edge. The pattern in 4.7(b) is found everywhere the pattern in 4.7(a) is found. The number of occurrences of these two patterns is very close or same in every example of a dataset. As a result, the significance of these patterns together in a set is equal to significance of only one of them. In this section we seek a set of most significant k patterns, which has low redundancy. The method described in this section is suggested by Xin *et al.* in [45]. This study gives the definitions for *pattern significance* and *pattern redundancy* as a part of formal problem formulation.

Definition 4.9 (Pattern Significance, [45]).

Given pattern language $\mathcal{L}$, a significance measure is a function $S : \mathcal{L} \rightarrow \mathbb{R}$, $S(\pi)$ is the degree of interestingness (or usefulness) of pattern π .

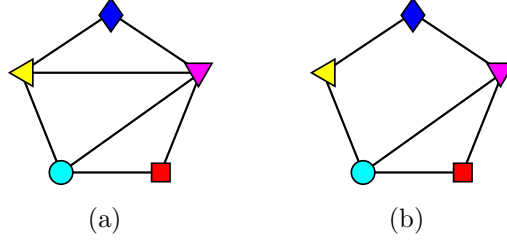


Figure 4.7: Two sample redundant graph patterns

Since we focus on classification problem in this study, we choose γ function as a measure of pattern significance, so $S(\pi) = \gamma(p_\pi^c, n_\pi^c)$. Xin *et al.* extend the pattern significance to *combined significance* and *relative significance*. Let the combined significance $S(\pi, \pi')$ denote the collective significance of two individual patterns π and π' . Before defining relative significance, we need to define pattern redundancy. Given significance measures, pattern redundancy is defined as follows.

Definition 4.10 (Pattern Redundancy, [45]).

Given the significance measure S , the redundancy R between two patterns π and π' is defined as $R(\pi, \pi') = S(\pi) + S(\pi') - S(\pi, \pi')$.

Subsequently, the relative significance of π given π' is $S(\pi | \pi') = S(\pi) - R(\pi, \pi')$. Assuming that combined significance is not less than the significance of any individual pattern and not greater than the sum of two individual significance; then the redundancy between patterns π and π' should satisfy

$$0 \leq R(\pi, \pi') \leq \min(S(\pi), S(\pi')). \quad (4.53)$$

According to [45], the ideal redundancy measure $R(\pi, \pi')$ is usually hard to obtain. Therefore, they suggest using distance between patterns in order to measure patterns redundancy.

Definition 4.11 (Pattern Distance, [45]).

A distance measure $D : \mathcal{L} \times \mathcal{L} \rightarrow [0, 1]$ is a mapping from two patterns $\pi, \pi' \in \mathcal{L}$ to a value in $[0, 1]$, where 0 means π and π' are completely relevant and 1 means π and π' are totally independent.

The pattern distance is used for approximating the pattern redundancy. The following equation which satisfies (4.53) is given in [45].

$$R(\pi, \pi') = (1 - D(\pi, \pi')) \times \min(S(\pi), S(\pi')). \quad (4.54)$$

The distance between two patterns depends on the pattern structure. It can be the edit distance for sequences or the graph edit distance [37] for graph patterns. Alternatively, in the distance function we use the number of occurrences of patterns in a dataset because it provides the correlation between two patterns in document level. We define the distance function using *cosine similarity* in the field of information retrieval as follows.

$$D(\pi, \pi') = 1 - \cos_{\mathcal{D}}(\pi, \pi') = 1 - \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|} \quad (4.55)$$

where $\mathbf{u} = \vec{f}_{\mathcal{D}}(\pi)$ and $\mathbf{v} = \vec{f}_{\mathcal{D}}(\pi')$. We use the whole dataset $\mathcal{D} = \bigcup_{i \in \mathcal{C}} \mathcal{D}_i$, in distance computation where $\vec{f}_{\mathcal{D}}$ is defined as

$$\vec{f}_{\mathcal{D}}(\pi) = (\text{match}(\pi, x_1), \dots, \text{match}(\pi, x_n))^{\top} \quad (4.56)$$

where $\mathcal{D} = \{x_1, \dots, x_n\}$. By the help of this distance function, we try to find a set of patterns which occur in not only some examples of the dataset $\mathcal{D}_c$ in which we are interested, but in all examples of $\mathcal{D}_c$ if possible.

Finally, the formulation is extended to a set of k patterns $\mathcal{P}^k \subset \mathcal{L}$. Let G be a significance measure for a set of patterns, and the redundancy measure for a set of patterns be L which is hard to obtain [45]. In general,

$$G_{gen}(\mathcal{P}^k) = \sum_{i=1}^k S(\pi_i) - L(\mathcal{P}^k). \quad (4.57)$$

The authors of [45] suggest two heuristic evaluation functions G_{as} (average significance) and G_{ms} (marginal significance). We prefer using marginal significance so we only explain this function. The computational model for the new concept is a graph called *redundancy graph*:

Definition 4.12 (Redundancy Graph, [45]).

Given a significance measure S and a redundancy measure R for individual patterns, a redundancy graph of a set of k patterns $\mathcal{P}^k$ is a weighted graph where

each pattern, π_i , corresponds to node i whose weight is pattern significance $S(\pi_i)$ and the weight on an edge (i, j) is the redundancy $R(\pi_i, \pi_j)$.

Marginal significance of a set of patterns is computed as

$$G_{ms}(\mathcal{P}^k) = \sum_{i=1}^k S(\pi_i) - w(MST(\mathcal{P}^k)) \quad (4.58)$$

where $w(MST(\mathcal{P}^k))$ denotes the sum of edge weights on the *maximum spanning tree* of the redundancy graph. Given a pattern language $\mathcal{L}$, the problem of maximal marginal significance (MMS) is to find a set of k patterns $\mathcal{P}^k$ such that $G_{ms}(\mathcal{P}^k)$ is maximized. Finally, the study [45] gives Algorithm 2 for the problem of MMS.

According to our definitions of pattern significance and distance, we obtain a redundancy-aware set of frequent top- k patterns in terms of correlation. We denote this set by $\mathcal{S}_c$ for class $c \in \mathcal{C}$ and we apply these mining steps for each class. Then, all selected patterns are collected in one set $\mathcal{S}$ which will be used for classification. This completes the process of finding a set of patterns satisfying our objectives given in the beginning of this chapter.

4.5 Summary of the Mining Algorithm

The procedure so far is aiming to find a set of patterns satisfying our objectives given at the beginning of this chapter. Each step can be considered as a pattern filter which is responsible for one criterion. To reduce the computational cost, the filters are connected in series, in other words the input patterns of a filter is the output of another. The order of filters are same with the section order. First, the patterns generated by the language $\mathcal{L}$ are tested for support with respect to the threshold θ_c^{supp} and the frequent ones are collected in the set $\mathcal{F}_c$. Next, the members of this set are tested for the correlation with respect to the threshold θ_c^{cor} and the set $\mathcal{R}_c$ contains patterns which are both frequent and correlated. The last filter seeks a small subset $\mathcal{P}_c$ which has significant and not redundant

Algorithm 2 Greedy Algorithm for MMS, [45]

Input: A set of n patterns, $\mathcal{L} = \{\pi_1, \dots, \pi_n\}$ Number of output patterns, k Significance measure, S Divergence measure, D **Output:** Top- k pattern set, $\mathcal{P}^k$

```

1:  $t \leftarrow 0$ ,  $T \leftarrow \max_{\pi \in \mathcal{L}} S(\pi)$ 
2:  $\text{selected}[i] \leftarrow \text{false}$  for  $i = 1, \dots, n$ 
3:  $\text{removed}[i] \leftarrow \text{false}$  for  $i = 1, \dots, n$ 
4: for  $i \leftarrow 1$  to  $k$  do
5:   if there is no pattern left then
6:      $T \leftarrow \frac{T+t}{2}$ 
7:     goto line 2
8:   end if
9:    $\pi_s \leftarrow$  the most significant pattern s.t.
        $\text{selected}[s] = \text{false}$  and  $\text{removed}[s] = \text{false}$ 
10:   $\text{selected}[s] \leftarrow \text{true}$ 
11:   $\text{removed}[s] \leftarrow \text{true}$ 
12:  for  $j \leftarrow 1$  to  $n$  do
13:    if  $\text{removed}[j] = \text{false}$  and  $\text{selected}[j] = \text{false}$  then
14:      if  $S(\pi_j \mid \pi_s) \leq \frac{T+t}{2}$  then
15:         $\text{removed}[j] \leftarrow \text{true}$ 
16:      end if
17:    end if
18:  end for
19: end for
20: if there are patterns left ( $\pi_i$  s.t.  $\text{removed}[i] = \text{false}$ ) then
21:    $t = \frac{t+T}{2}$ 
22:   goto line 2
23: end if
24: return selected  $k$ -patterns ( $\pi_i$  s.t.  $\text{selected}[i] = \text{true}$ )

```

patterns. These steps are repeated for each class $c \in \mathcal{C}$ and then the patterns satisfying these criteria for some classes are grouped into the final set $\mathcal{S}$ which is used for classification. Algorithm 3 shows the steps of the mining algorithm.

Algorithm 3 Pattern Mining Algorithm

Input: A pattern language, $\mathcal{L}$
 A dataset $\mathcal{D}$ labeled with classes $\mathcal{C}$
 Support thresholds, θ_c^{supp} for each $c \in \mathcal{C}$
 Correlation thresholds, θ_c^{cor} for each $c \in \mathcal{C}$
 Number of top patterns for each class, k
 An evaluation predicate, φ

Output: A pattern set, $\mathcal{S}$

```

1:  $\mathcal{S} \leftarrow \emptyset$ 
2: for each class  $c \in \mathcal{C}$  do
3:    $\mathcal{F}_c \leftarrow \emptyset$ 
4:   for each pattern  $\pi$  generated by  $\mathcal{L}$  do
5:     if  $\text{supp}(\pi, \mathcal{D}_c, \varphi) \geq \theta_c^{\text{supp}}$  then {supp defined in Definition 4.4}
6:        $\mathcal{F}_c \leftarrow \mathcal{F}_c \cup \{\pi\}$ 
7:     end if
8:   end for
9:    $\mathcal{R}_c \leftarrow \emptyset$ 
10:  for each pattern  $\pi \in \mathcal{F}_c$  do
11:    compute  $p_\pi^c$  and  $n_\pi^c$  on  $\mathcal{D}$  {defined in (4.32)}
12:    if  $\gamma(p_\pi^c, n_\pi^c) \geq \theta_c^{\text{cor}}$  then { $\gamma$  defined in (4.33)}
13:       $\mathcal{R}_c \leftarrow \mathcal{R}_c \cup \{\pi\}$ 
14:    end if
15:  end for
16:  for each pattern  $\pi \in \mathcal{R}_c$  do
17:    compute pattern significance as  $\mathbf{S}[\pi] = \gamma(p_\pi^c, n_\pi^c)$ 
18:    for each pattern  $\pi' \in \mathcal{R}_c$  do
19:      compute pattern distances  $\mathbf{D}[\pi, \pi'] = 1 - \cos_{\mathcal{D}}(\pi, \pi')$  {defined in (4.55)}
20:    end for
21:  end for
22:   $\mathcal{P}_c \leftarrow$  output of Greedy Algorithm for MMS with  $\mathcal{R}_c$ ,  $k$ ,  $\mathbf{S}$  and  $\mathbf{D}$ 
23:   $\mathcal{S} \leftarrow \mathcal{S} \cup \mathcal{P}_c$ 
24: end for
25: return  $\mathcal{S}$ 

```

We can analyze search regions of patterns in the pn space. Assume that the support threshold θ_c^{supp} is chosen greater than the lower bound given in Theorem 4.1. The members of $\mathcal{F}_c$ are found in the shaded area (union of dark and

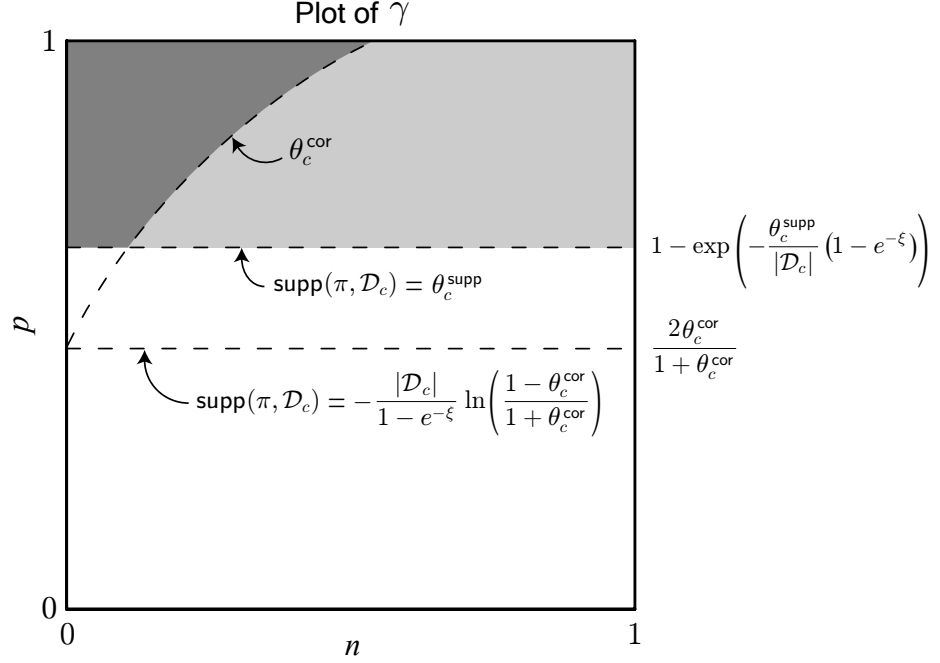


Figure 4.8: The pn space showing the search regions for the first two steps of the algorithm. The shaded area (union of dark and light gray) represent the domain region of $\mathcal{F}_c$ and dark gray area represents the domain region of $\mathcal{R}_c$.

light gray) in Figure 4.5 according to Corollary 4.2. After the correlation check, the region of $\mathcal{R}_c$ is reduced to the dark gray area in the figure. The final step does not reduce the search region in the pn space because set $\mathcal{P}_c$ may contain a pattern π on the correlation boundary $S(\pi) = \theta_c^{\text{cor}}$ because of the redundancy definition.

Someone may claim that changing the order of the first two steps causes the computational cost to reduce. However, it is not sensible most of the time because the support test has negligible computation cost compared to the correlation test which includes the execution of the EM algorithm. Furthermore, a structural language which has a generalization relation among its patterns enables us to define an *anti-monotonic matching function*. Such a function provides the pruning ability in generating frequent patterns of the language using the *pattern growth approach*. For example, many frequent graph mining algorithms generate subgraphs from the dataset. They start generating subgraphs from one node subgraphs and continue generating new subgraphs by adding nodes and edges to

the previous ones. In generating process, the subgraphs whose support in the dataset is less than the threshold are eliminated and only the frequent ones are preserved. The algorithm stops when no new subgraph can be generated from the frequent subgraphs.

To recall the theory of ϕ in Section 4.1, the mining algorithm also tries to find the theory of a selection predicate. Now, we can define the selection predicate ϕ to find the theory with respect to the language $\mathcal{L}$ and the dataset $\mathcal{D}$ as follows.

$$\phi(\pi, \mathcal{D}) = \begin{cases} true & \text{if } \exists c \in \mathcal{C} : \text{supp}(\pi, \mathcal{D}_c) \geq \theta_c^{\text{supp}} \wedge \gamma(p_\pi^c, n_\pi^c) \geq \theta_c^{\text{cor}} \wedge \pi \in \mathcal{P}_c \\ false & \text{otherwise.} \end{cases} \quad (4.59)$$

4.6 Graph Patterns

In this section, we narrow down the theoretical concepts in the previous sections to graph patterns. All computations in Sections 4.2 to 4.4 are based on definitions in Section 4.1 and the abstraction of these definitions for all pattern types is based on the evaluation predicate φ . We will define alternative φ predicates for graph patterns. Therefore, we start to use the following graph terminology:

pattern $\pi \longrightarrow$ subgraph g

example $x \longrightarrow$ image (input) graph G

dataset $\mathcal{D} \longrightarrow$ graph set $\mathcal{G}$

Accordingly, a mapping h of pattern π into example x , given in the matching function definition is called *subgraph isomorphism* in the graph terminology. The definition is given by Fiedler and Borgelt as follows.

Definition 4.13 (Subgraph isomorphism, [18]).

Let $g = (V_g, E_g, \ell_g)$ and $G = (V_G, E_G, \ell_G)$ be two labeled graphs. A subgraph isomorphism of g to G is an injective function $h : V_g \rightarrow V_G$ satisfying $\forall v \in V_g : \ell_g(v) = \ell_G(h(v))$ and $\forall (u, v) \in E_g : (h(u), h(v)) \in E_G \wedge$

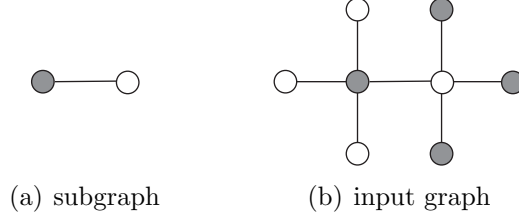


Figure 4.9: An example for overlapping embeddings

$$\ell_g((u, v)) = \ell_G((f(u), f(v))).$$

Every subgraph isomorphism of subgraph g to image graph G defines an *embedding* of subgraph g . Different embeddings of g may refer to same nodes in G as in Figure 4.9. They are called *overlapping subgraph isomorphism*.

Definition 4.14 (Overlapping subgraph isomorphism, [18]).

Let $g = (V_g, E_g, \ell_g)$ and $G = (V_G, E_G, \ell_G)$ be two labeled graphs, h_1 and h_2 two subgraph isomorphisms of g to G , and $V_i = \{v \in V_G \mid \exists u \in V_g : v = h_i(u)\}$, $E_i = \{e \in E_G \mid \exists (u, v) \in E_g : e = (h_i(u), h_i(v))\}$ for $i = 1$ or 2 . Two subgraph isomorphisms h_1 and h_2 are called overlapping iff $V_1 \cap V_2 \neq \emptyset$ and written $h_1 \bowtie h_2$. Also, h_1 and h_2 are called equivalent, written $h_1 \circ h_2$, iff $V_1 = V_2$ and $E_1 = E_2$. Finally, h_1 and h_2 are called identical, written $h_1 \equiv h_2$, iff $\forall v \in V_g : h_1(v) = h_2(v)$.

Two identical subgraph isomorphisms are treated as only one embedding because they refer to the same nodes. However, two equivalent subgraph isomorphisms may not be identical if the subgraph contains the same label for more than one node; for example, subgraph A-B-A and input graph A-B-A-C. We define the set of all embeddings which are not identical as below:

$$\mathcal{H}(g, G) = \{h \mid h(g) \subseteq G \wedge \forall h' \in \mathcal{H}(g, G) : h \neq h' \rightarrow \neg(h \equiv h')\}. \quad (4.60)$$

The simplest evaluation predicate is defined by

$$\varphi_{all}(h, g, G) = \begin{cases} true & \text{if } h \in \mathcal{H}(g, G) \\ false & \text{otherwise.} \end{cases} \quad (4.61)$$

Using this definition for the subgraph g in Figure 4.9(a) and the input graph G in Figure 4.9(b), the result of match function is $\text{match}(g, G, \varphi_{all}) = 7$. Let G' be an image graph containing 7 instances of g independently. The matching function using the predicate φ_{all} produces the same result for both cases. However, a desirable matching function should produce a greater value for the input graph G' . This example shows the importance of handling overlapping embeddings. Fortunately, more sophisticated methods have been proposed in the graph mining literature, that we will describe later.

Most of the graph mining methods use binary matching function which only checks whether input graph contains an instance of given subgraph, or not. This function can be obtained by an evaluation predicate, φ_{bin} , which returns true if just one single embedding exists. However, we do not prefer binary matching function because the image graphs we used in the experiments may contain a structure belonging to another class, i.e. a single house in an instance of forest image.

A method called the *maximum independent set (MIS) support* is proposed by Kuramochi and Karypis for handling overlapping embeddings [25]. They introduce the *overlap graph* for computing this support measure where each embedding corresponds to a node and an edge is inserted between two nodes if they are overlapping. Every embedding of the subgraph in Figure 4.9(a) is shown in Figure 4.10(a) and the corresponding overlap graph can be seen in Figure 4.10(b). The maximum independent set is found by removing minimum number of nodes from this graph, that makes the remaining nodes independent (unconnected) and the MIS support in a single graph is computed as the number of remaining nodes. The maximum independent set of the overlap graph in Figure 4.10(b) has two nodes: one node from the set $\{1, 2, 3\}$ and one node from the set $\{5, 6, 7\}$. We formally define this support measure as follows.

Definition 4.15 (Maximum independent set of embeddings).

Given a subgraph g , an input graph G , and an independence predicate ω which takes a set of embeddings as an argument and returns true if its members are not

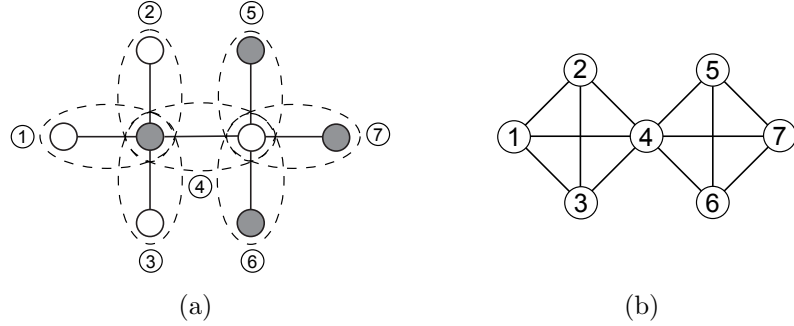


Figure 4.10: In (a) The embeddings of the subgraph in Figure 4.9(a); in (b) the corresponding overlap graph.

overlapping. The maximum independent set (MIS) of embeddings is defined as

$$H_{\text{MIS}}(g, G) = \arg \max_{\substack{H \subset \mathcal{H}(g, G), \\ w(H) \text{ is true}}} |H| \quad (4.62)$$

The evaluation predicate of the MIS support is given by

$$\varphi_{\text{MIS}}(h, g, G) = \begin{cases} \text{true} & \text{if } h \in H_{\text{MIS}}(g, G) \\ \text{false} & \text{otherwise.} \end{cases} \quad (4.63)$$

According to this predicate, the result of the matching function for Figure 4.9 is only 2, which is more realistic. The evaluation predicate which we use for graph mining is an extension of φ_{MIS} . It also evaluates the subgraph $g = (V_g, E_g, \ell_g)$ in terms of graph size with the predefined size limits: minimum number of nodes $v_{\min}$ and maximum number of nodes $v_{\max}$. We define the new evaluation predicate as

$$\varphi(h, g, G) = \begin{cases} \text{true} & \text{if } v_{\max} \geq |V_g| \geq v_{\min} \wedge \varphi_{\text{MIS}}(h, g, G) \text{ is true} \\ \text{false} & \text{otherwise.} \end{cases} \quad (4.64)$$

This concludes our discussion of graph mining. Algorithm 3 summarizes the graph mining procedure with the evaluation predicate φ . Sample subgraphs mined by the algorithm using the predicate φ , and the embeddings of these subgraphs in the image graphs are given in Figure 4.11 for three different classes. The effects of parameters, thresholds and the computational complexity will be discussed in Chapter 6.

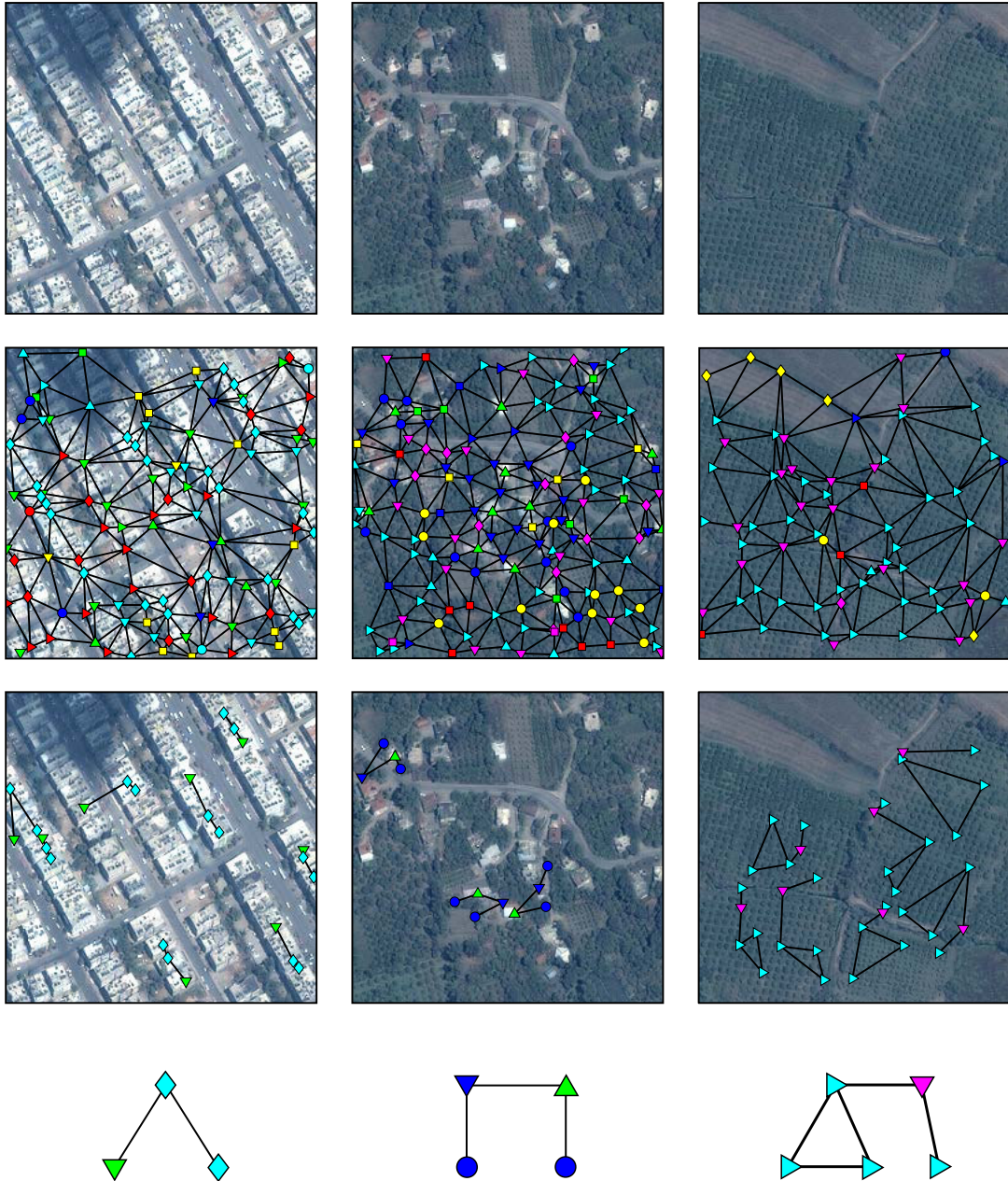


Figure 4.11: Images from top to down are original images from three different classes, image graphs for 36 labels, embeddings of sample subgraphs found by the mining algorithm and the sample subgraphs where the color and shape of a node represents its label.

Chapter 5

Scene Classification

All models are wrong, but some are useful.

GEORGE E. P. BOX

After finding the set of subgraphs $\mathcal{S}$, we represent every image graph as a subgraph histogram vector. These histogram vectors are used for model learning for each class. Finally, the support vector machine (SVM) using these models decides the best model for an unseen image. Another learning model which we use is Latent Dirichlet Allocation (LDA). Given a subgraph histogram of an image, LDA provides further representation of the image based on the theme distribution in the image. This representation enables classification of images according to their theme distributions and helps us to identify heterogeneous image content. In the following sections we describe the image representation, classification of images using SVM and theme discovery in the images using LDA.

5.1 Subgraph Histogram Representation

The subgraph histogram provides a powerful representation that is not as complex as full graph models, and reduces the complexity of graph similarity computation. The histogram is constructed using the support of each subgraph in the set $\mathcal{S}$

selected by graph mining. Each image graph G in the graph set $\mathcal{G}$ is transformed into a histogram feature vector

$$\mathbf{x} = (x_1, \dots, x_n)^\top \quad (5.1)$$

where $x_i = \text{match}(g_i, G, \varphi)$ and $g_i \in \mathcal{S}$ for $i = 1, \dots, n$. By this way, images can be classified in this feature space using statistical pattern recognition techniques.

5.2 Support Vector Machines

Support vector machines (SVM) are popular data classification technique. In this section we give a brief description of the SVM and discuss only the parameters; more details can be found in [13]. Given a training dataset with class labels for each instance, SVM maps training vectors $\mathbf{x}_i$ to a higher dimensional space. The goal is to find a linear separating hyperplane with the maximal margin in this higher dimensional space. We use a multi-class support vector machine with a radial basis function kernel (RBF) for image classification. The kernel function is

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2} \quad \text{for } \gamma > 0. \quad (5.2)$$

The multi-class SVM is a combination of one-against-one class SVMs where the output class is the one with the maximum number of votes [13].

We have only two SVM parameters to decide: The error parameter C and the kernel parameter γ . A *grid search* on parameters C and γ is recommended by libSVM [13] using cross-validation. Various values for the pair (C, γ) are tried and the pair with the best cross-validation accuracy on training data is selected. The values tried are selected from exponentially growing sequences to identify good parameters. Additionally, we normalize each feature to the range $[0,1]$ before training the SVM.

5.3 Latent Dirichlet Allocation

The dataset used in the experiments consists of homogeneous tiles cut from the Antalya image shown in Figure 1.2 where the tile size is 250×250 pixels. We have approximately 950 tiles excluding the sea tiles when we cut the Antalya image; however, the dataset contains only 585 tiles. Almost 40 percent of land tiles are not selected to the dataset due to heterogeneous content (or unclassified content). Further classification of these tiles enables partitioning of the whole satellite image into semantic class regions. For this purpose, we employ a generative probabilistic model for theme discovery in these images.

Latent Dirichlet allocation (LDA) introduced by Blei *et al.* in is a generative probabilistic model for collections of discrete data such as text corpora [7]. The basic idea of LDA is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words. LDA, which is originally developed from text modeling, is easily adapted to our graph data by making analogies between document–image graph, word–subgraph, corpus–image graph set and topic–theme. Likewise, LDA has been adapted to computer vision by drawing an analogy between words and image patches in [28]. LDA is defined using the following terms:

- A subgraph is the basic unit of an image graph, defined to be an item from a graph set indexed by $\{1, \dots, T\}$. The v th subgraph is represented by a T -vector g such that $g^v = 1$ and $g^u = 0$ for $u \neq v$.
- An image graph is a set of N subgraphs indexed by $G = \{g_1, \dots, g_N\}$, where g_i is the i th subgraph in the set.
- An image graph set is a collection of M image graphs denoted by $\mathcal{G} = \{G_1, \dots, G_M\}$.

LDA assumes the following generative process for each image graph G in an image graph set $\mathcal{G}$:

1. For each image graph, choose $\delta \sim \text{Dir}(\eta)$ where δ is the parameter of a multinomial distribution for choosing the themes and η is the K -dimensional Dirichlet parameter. Total number of themes is assumed known and fixed as K .
2. For each of the N subgraphs g_i in the image graph
 - (a) Choose a theme $z_i \sim \text{Multinomial}(\delta)$ where z_i is a K -dimensional unit vector. $z_i^k = 1$ indicates that the k th theme is selected.
 - (b) Choose a subgraph g_i from $p(g_i | z_i, \beta)$, a multinomial probability conditioned on the theme z_i and β is a $K \times T$ matrix where $\beta_{ij} = p(g^j = 1 | z^i = 1)$, is a fixed quantity for the graph set.

A K -dimensional Dirichlet random variable δ satisfies $\delta_i \geq 0$ and $\sum_{i=1}^K \delta_i = 1$. It has the following probability density:

$$p(\delta | \eta) = \frac{\Gamma(\sum_{i=1}^K \eta_i)}{\prod_{i=1}^K \Gamma(\eta_i)} \delta_1^{\eta_1-1} \dots \delta_K^{\eta_K-1} \quad (5.3)$$

where the parameter η is a K -vector with components $\eta_i > 0$, and $\Gamma(\cdot)$ is the Gamma function.

Given the parameters η and β , the joint distribution of a theme mixture δ , a set of N themes $Z = \{z_1, \dots, z_N\}$, and an image graph G having N subgraphs $\{g_1, \dots, g_N\}$ is given by

$$p(\delta, Z, G | \eta, \beta) = p(\delta | \eta) \prod_{n=1}^N p(z_n | \delta) p(g_n | z_n, \beta) \quad (5.4)$$

where $p(z_n | \delta)$ is simply δ_i for the unique i such that $z_n^i = 1$.

The LDA model is represented as a probabilistic graphical model in Figure 5.1. The parameters η and β are the dataset-level parameters. They are assumed to be sampled once for generating a graph set. The variable δ is graph-level variable which is sampled once per image graph. Finally, the variables z and g are subgraph-level parameters, sample once for each subgraph in an image graph.

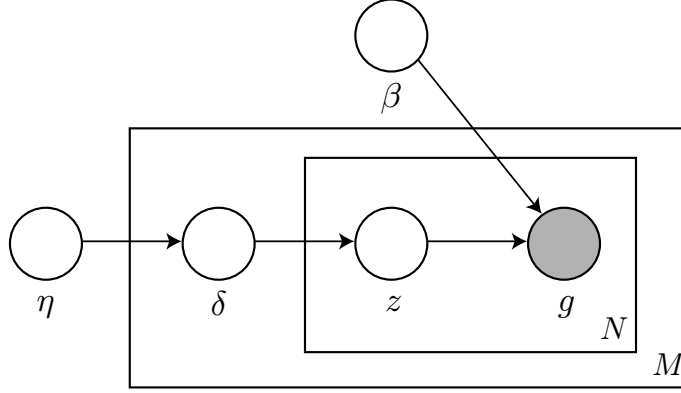


Figure 5.1: Graphical model representation of LDA. The boxes are *plates* representing replicates. The outer plate represents image graphs, while the inner plate represents the repeated choice of themes and subgraphs within an image graph [7].

In the context of text modeling, the topic probabilities provide an explicit representation of a document [7]. Equivalently, we use the theme probabilities to represent an image graph. To do so, we need to compute the posterior distribution of the hidden variables given an image graph:

$$p(\delta, Z | G, \eta, \beta) = \frac{p(\delta, Z, G | \eta, \beta)}{p(G | \eta, \beta)}. \quad (5.5)$$

Integrating (5.4) over δ and summing over z , we obtain the marginal distribution of an image graph in terms of the model parameters:

$$p(G | \eta, \beta) = \frac{\Gamma(\sum_{i=1}^K \eta_i)}{\prod_{i=1}^K \Gamma(\eta_i)} \int \left(\prod_{i=1}^K \delta_i^{\eta_i - 1} \right) \left(\prod_{n=1}^N \sum_{i=1}^K \prod_{j=1}^T (\delta_i \beta_{ij})^{g_n^j} \right) d\delta. \quad (5.6)$$

Unfortunately, this distribution is intractable due to the coupling between δ and β in the summation over latent themes [7]. However, a wide range of approximate inference algorithms can be considered, including Laplace approximation, variational approximation and MCMC method [7]. The solution proposed by Blei *et al.* is approximating the distribution $p(\delta, Z | G, \eta, \beta)$ by a simplified graphical model with free variational parameters as

$$q(\delta, Z | \psi, \phi) = q(\delta | \psi) \prod_{n=1}^N q(z_n | \phi_n) \quad (5.7)$$

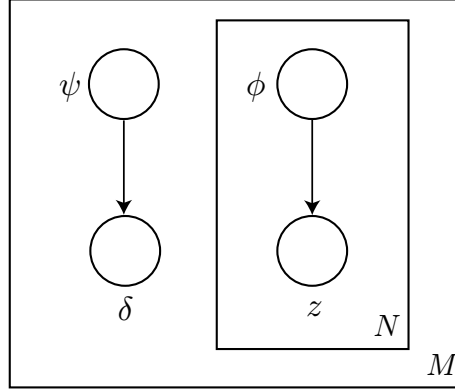


Figure 5.2: Graphical model representation of the variational distribution used to approximate the posterior in LDA [7].

where the Dirichlet parameter ψ and the multinomial parameters $(\phi_1, \dots, \phi_N)$ are the free variational parameters. The next step is to find the values of the variational parameters ψ and ϕ , which satisfy the best approximation as

$$(\psi^*, \phi^*) = \arg \min_{(\psi, \phi)} KL(q(\delta, Z | \psi, \phi) \parallel p(\delta, Z | G, \eta, \beta)). \quad (5.8)$$

Thus, the optimal values of the variational parameters are found by minimizing the Kullback-Leibler (KL) divergence between the variational distribution and the true posterior $p(\delta, Z | G, \eta, \beta)$. The values are computed using the EM algorithm; however, we do not include these steps to this thesis. The details of the computation can be found in [7]. The computations show that the optimal parameters (ψ^*, ϕ^*) are functions of the given graph G such that $(\psi^*(G), \phi^*(G))$. Therefore, the theme probabilities of an unseen image graph can be computed in the same manner. In addition, the values of the model parameters η and β are estimated from the variational parameter values of all image graphs in the graph set, iteratively. The graphical model of this variational distribution can be seen in Figure 5.2.

Given a set of image graphs, we can estimate the model parameters η and β using variational inference. The graph-level variable δ can be used for graph representation. It is a K -dimensional vector where the i th component equals to the probability $p(z_i | \delta, G)$ for the given graph G . The variational method finds the optimal Dirichlet parameter $\psi^*(G)$ of the distribution that generates the variable

δ for the given graph G . Therefore, normalizing the K -dimensional Dirichlet parameter gives the expected value of the theme distribution which can be used for graph representation.

After training the LDA model on homogeneous tiles for a defined number of themes, we compute the theme distribution vectors of all tiles in the whole satellite image using variational inference. Finally, the theme probabilities of tiles are used for partitioning the whole image into semantic regions.

Chapter 6

Experimental Results

Where is the ‘any’ key?

HOMER SIMPSON – IN RESPONSE TO
THE MESSAGE “PRESS ANY KEY”

In this chapter, we present the results of the experiments conducted for the proposed method in comparison to the bag-of-words model. The dataset used in the experiments is previously described in Section 1.3. In the following sections we first describe the experimental setup and then demonstrate the experimental results.

6.1 Experimental Setup

We explained the proposed method for image classification in the previous three chapters. The following section gives the values for the parameters of the method and the external softwares used as a part of experiments. We conduct experiments also for the bag-of-words model in the same dataset. For the bag-of-words model we extract the histogram vector of the node labels computed from individual MSERs for each image and then the same classification procedure given in Chapter 5 is applied.

6.1.1 Graph Construction Parameters

The transformation process has three parameter sets: The MSER parameter sets Ω_{low} , Ω_{high} and the parameter set for the number of labels Υ . We determine the MSER parameters experimentally as $\Omega_{\text{high}} = (10, 60, 5000, 0.4, 1)$ and $\Omega_{\text{low}} = (5, 35, 1000, 4, 1)$. Also, the ellipse expanding parameters in (3.6) and (3.7) are experimentally set to $r_1^2 = 5$ and $r_2^2 = 20$. The MSER algorithm is applied to the whole remote sensing image with these parameters and the features are extracted from the image. Then, the node labels are determined by the k -means algorithm with $\Upsilon = (k_{\text{sd}}, k_{\text{sb}}, k_{\text{ud}}, k_{\text{ub}})$. We set k_{sd} and k_{sb} to equal values; and similarly k_{ud} and k_{ub} . The number of highly stable MSERs is less than the number of less stable ones. Therefore, we choose the number of labels for less stable MSERs as twice the number of labels for highly stable ones. Hence, we use the total number of labels denoted by N_ℓ instead of the set Υ . For example, we say $N_\ell = 36$ for $\Upsilon = (6, 6, 12, 12)$. The parameter value of N_ℓ is selected from the set $\{18, 26, 36, 54, 71\}$ in the experiments. Next, the graph transformed from the whole image are cut into tiles and every homogeneous tile is labeled by one of the eight semantic classes. Finally, these tiles are divided into two sets which have (almost) same number of tiles for each class. One of them which has 295 images is used for model training and the other set containing 290 images is used for testing. The external software used in this section is VLFeat [43] for not only the MSER algorithm but also the k -means++ algorithm.

6.1.2 Graph Mining Parameters

The mining algorithm is based on the evaluation predicate. We use the predicate φ in (4.64) for graph mining. The graph mining algorithm has three steps. The parameter of the first step is the support threshold θ_c^{supp} for each class $c \in \mathcal{C}$. A global threshold for all classes is not reasonable because the size of dataset $\mathcal{D}_c$ varies from one class to another. A threshold for $\text{freq}(\pi, \mathcal{D})$ may be applied; however, the density of MSERs also differs between classes. We choose three different threshold values for each class in order to simulate the effects of high,

medium and low thresholds which allow small, medium and large number of subgraphs to pass to the next filter, respectively. The desired size of $\mathcal{F}_c$ for high, medium and low thresholds are determined as 200, 500 and 800, respectively. We denote this number by N_θ . To choose the support thresholds accordingly, we rank all subgraphs in order of support and select the support value of the N_θ th subgraph as the threshold θ_c^{supp} for each class. The same procedure is applied for the correlation threshold θ_c^{cor} . Note that the correlation thresholds are chosen from the ranking of *all subgraphs*, not only the frequent ones. In the experiments, we use the same parameter N_θ for the selection of both thresholds in a single run of the algorithm. The last significant parameter of the algorithm is the number N_s of subgraphs selected for the set $\mathcal{S}_c$ for each class. We have eight classes, so each image is represented by a $8 \times N_s$ -dimensional subgraph histogram vector. The domain of the parameter N_s is $\{1, 2, \dots, 20\}$ in the experiments.

Addition to the parameters N_ℓ , N_θ and N_s , there are some less important parameters: Subgraph size limits $v_{\min}$ and $v_{\max}$ in (4.64), the number of components in the Poisson mixtures m in (4.11), and the regularization constant ξ in (4.31). We mine subgraphs which have at least two nodes $v_{\min} = 2$, and not more than five nodes $v_{\max} = 5$. The upper limit is used for reducing the computational complexity of frequent subgraph mining and the lower limit is set for differentiating subgraphs from the visual words of individual nodes (MSERs). The number of components in the Poisson mixtures is set to $m = 5$ for all subgraphs and the parameter ξ is chosen differently for each class as $\xi_c = \ln(2) / \text{freq}(\pi, \mathcal{D}_c)$ because the distance which corresponds to the average match in the dataset becomes equal to 0.5 by

$$d'(\text{freq}(\pi, \mathcal{D}_c)) = 1 - e^{-\xi_c \text{freq}(\pi, \mathcal{D}_c)} = 0.5. \quad (6.1)$$

The external software of this section is the Molecular Substructure Miner (MoSS) implemented by Borgelt [8] for mining frequent subgraphs.

6.1.3 Classifier Parameters

As mentioned in the previous chapter, there are two parameters of the SVM classifiers: C and γ . We set these parameters using grid search on 5-fold cross-validation with several parameter pairs from exponentially growing sequences. The domains are $C \in \{2^{-3}, 2^{-2}, \dots, 2^5\}$ and $\gamma \in \{2^{-5}, 2^{-4}, \dots, 2^2\}$. The parameter pair with the highest classification accuracy is selected as SVM parameters. If more than one pair has the same classification accuracy, the pair with the maximum-margin is selected. On the other hand, LDA has only one parameter: The number of themes K . The LDA model is learned from the training graph set and this model is used for inference of variational parameters for all tiles in the Antalya image.

6.2 Classification Results

The experiments are performed on an Ikonos image of Antalya, Turkey given in Figure 1.2. The tiles (250×250 pixel size) cut from this image are used in the classification experiments for eight semantic classes:

- (a) dense residential areas with large buildings,
- (b) dense residential areas with small buildings,
- (c) dense residential areas with trees,
- (d) sparse residential areas,
- (e) greenhouses,
- (f) orchards,
- (g) forests, and
- (h) fields.

Table 6.1: The number of images in the training and testing datasets for each class. Class names are in the text.

class	#training	#testing	total
(a)	40	39	79
(b)	35	35	70
(c)	18	17	35
(d)	24	24	48
(e)	17	17	34
(f)	82	81	163
(g)	38	37	75
(h)	41	40	81
total	295	290	585

The number of images in the training and testing datasets for each class are given in Table 6.1. The experiments are repeated on different parameter combinations in order to demonstrate the effects of parameters on classification performance.

The distribution of some MSER clusters over the whole image is given in Figures 6.1 and 6.2. The instances of the MSER clusters in these images are concentrated over a unique compound structure type. This shows the success of feature extraction step. In other words, the features extracted from the MSERs and their surroundings are so adequate to capture the local image content that we can distinguish the scene types by monitoring the locations in which they are detected. This situation is the main reason for the high performance of the bag-of-words model.

The classification accuracy of the graph mining algorithm for all parameter combinations in the experiments is shown in Table 6.2. The accuracy is computed for the test dataset, as the ratio of correctly classified images to the total number images. Note that the number of images that belong to one class varies from 17 (greenhouses) to 81 (orchards) in the test dataset of 290 images of 8 classes. The best classification accuracy is 92.069 percent, achieved by the parameter set $(N_\ell, N_\theta, N_s) = (36, 200, 9)$.

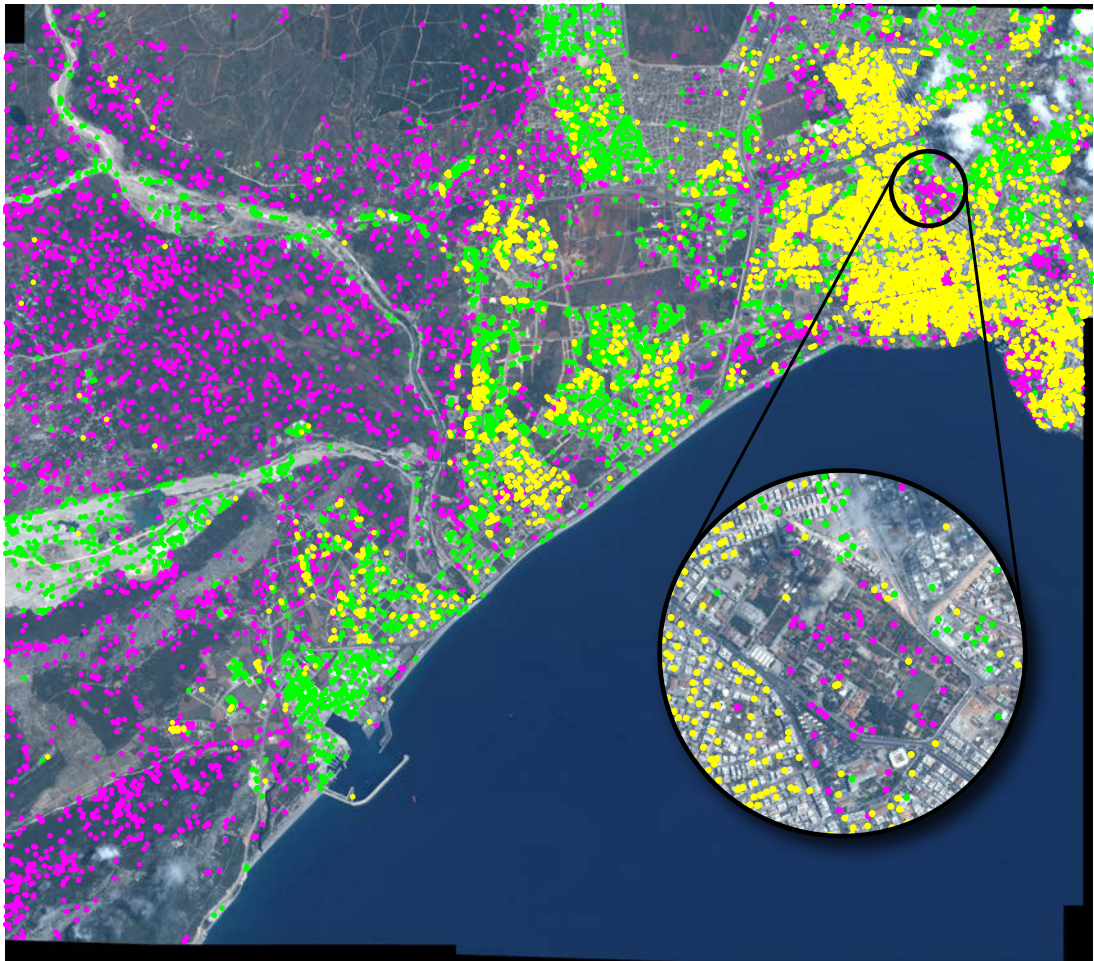


Figure 6.1: Three clusters of stable dark MSERs are drawn with different colors at ellipse centers for $N_\ell = 36$. Yellow, green and magenta points are concentrated on dense residential areas with large buildings, dense residential areas with small buildings and orchards, respectively.

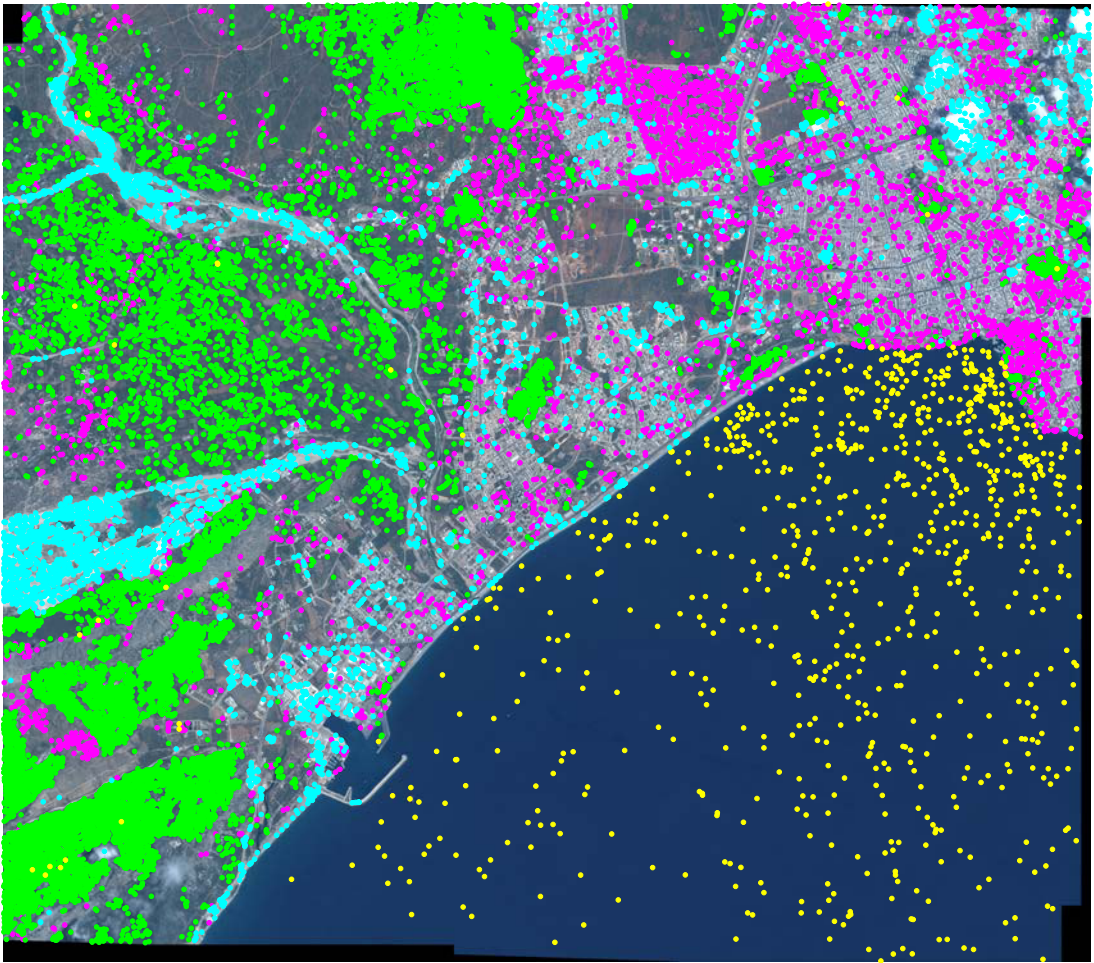


Figure 6.2: Four clusters of different type MSERs are drawn with different colors at ellipse centers for $N_\ell = 36$. Yellow, green, cyan and magenta points are concentrated on sea, forests, stream bed/clouds and dense residential areas with trees, respectively.

Table 6.2: The classification accuracy of the graph mining algorithm, in percentage (%), for all parameter sets in the experiments.

				Number of Subgraphs per class (N_s)																			
				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Number of Labels (N_ℓ)	18	N_θ	200	83	81	81	87	86	88	86	87	87	87	86	86	85	79	83	82	81	76	81	78
			500	73	82	85	83	82	82	88	86	88	87	88	87	85	82	82	79	78	76	75	74
			800	68	77	82	84	85	86	86	87	88	89	88	87	86	86	84	84	82	79	75	76
	26 ^a	N_θ	200	79	79	82	87	88	90	91	89	87	86	85	89	88	89	88	88	88	86	86	84
			500	74	72	84	83	84	85	86	81	86	88	88	87	83	86	84	83	84	84	83	81
			800	71	77	81	83	85	86	88	87	89	87	87	85	88	84	82	83	76	77	75	75
	36	N_θ	200	80	78	82	87	89	90	91	91	92	90	90	90	88	91	88	91	91	89	85	84
			500	80	77	87	85	87	90	90	89	88	88	89	88	89	88	87	87	73	71	73	69
			800	79	81	87	86	86	86	86	88	88	88	87	84	88	79	76	85	76	83	71	69
	54	N_θ	200	73	77	80	83	88	89	88	86	89	88	86	86	86	87	88	85	78	80	68	68
			500	74	79	81	82	82	84	84	88	87	88	88	84	81	72	82	73	76	69	67	66
			800	72	77	79	83	85	85	85	85	84	81	80	83	79	78	72	68	74	67	66	67
	72	N_θ	200	79	80	79	83	86	83	86	82	84	87	84	81	80	77	81	73	77	76	64	71
			500	79	78	80	80	81	81	80	82	84	81	81	80	81	80	76	72	68	67	65	62
			800	79	79	81	84	82	81	80	81	76	71	71	77	74	71	70	68	72	64	61	60

^aCorresponding cluster parameters is $\Upsilon = (4, 4, 9, 9)$

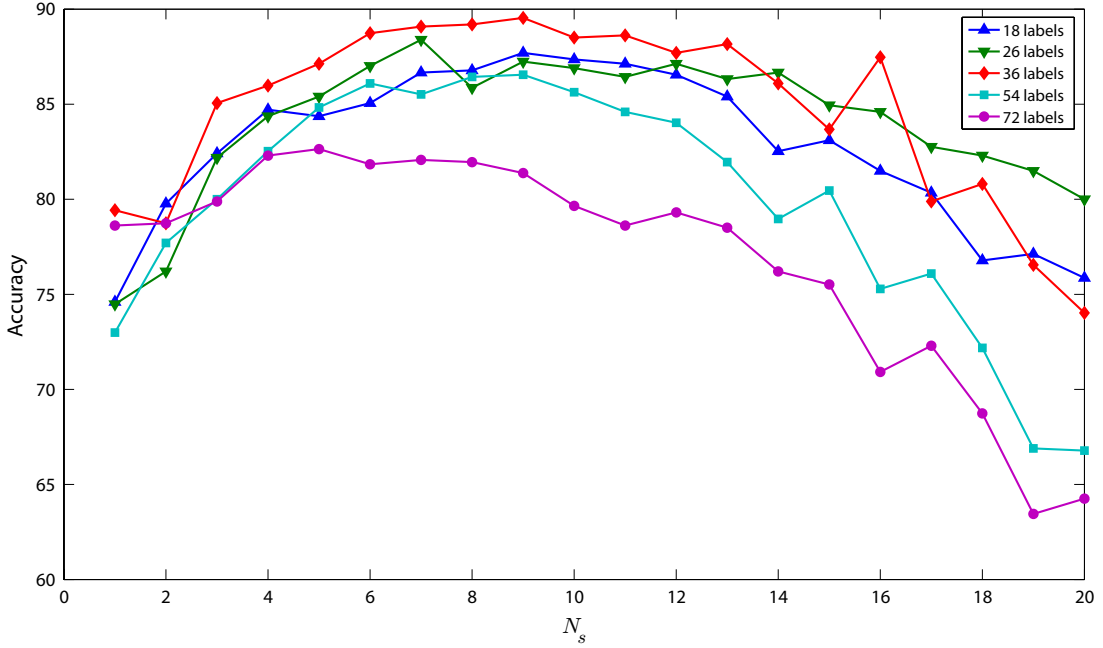


Figure 6.3: Plot of classification accuracy of the graph mining algorithm for five different number of labels over the number of subgraphs per class. The lines are drawn by averaging the accuracy values for the parameters $N_\theta \in \{200, 500, 800\}$.

Figure 6.3 demonstrates the effects of N_ℓ and N_s . Given these parameters, the line is drawn by averaging the classification accuracy over the set $N_\theta \in \{200, 500, 800\}$. If we start from $N_s = 1$ and increase the number of subgraphs selected for the set $\mathcal{S}$, i.e. the dimensionality of the feature space, the classification accuracy also increases until about $N_s = 8$ or 10 for all values of N_ℓ . However, after this point the classification accuracy decreases dramatically as expected because of the curse of dimensionality. A similar behavior is observed for the parameter N_ℓ . The best accuracy results are obtained when the number of labels is 36. High number of node labels such as 72 causes a significant decrease in subgraph frequency which leads to a fall in the accuracy. On the other hand, small number of node labels like 18 increases the subgraph frequency but reduces the correlation between subgraphs and classes. As a result, the confusion caused by less correlated subgraphs results in low classification accuracy. To sum up, every number of node labels entails trade-offs between frequency and correlation. Finding the optimal number of node labels in terms of classification accuracy requires experiments on the dataset with a range of N_ℓ values.

Besides classification performance, the number of node labels plays an important role in computational complexity. The crucial part of the graph mining algorithm with respect to the computational complexity is the frequent subgraph mining. The computational cost depends heavily on the number of overlapping embeddings because the MIS support solves an optimization problem for every overlapping embeddings. Small number of labels increases the number of overlapping embeddings and some classes, especially natural lands such as forest, fields and orchards tend to produce overlapping embeddings more than man-made structures like residential areas.

Figure 6.4 demonstrates the effects of N_θ and N_s . Given these parameters, the line is drawn by averaging the classification accuracy over the set $N_\ell \in \{18, 26, 36, 54, 72\}$. The parameter N_s , namely the number of subgraphs selected per class represents similar property as in Figure 6.3. On the other hand, a decrease in the thresholds θ_c^{supp} and θ_c^{cor} (inversely proportional to N_θ) causes the classification accuracy to decrease. The analysis of this case is harder than the previous. One possible explanation is that the set $\mathcal{S}$ contains strongly correlated but less frequent subgraphs for low thresholds. The last part of the mining algorithm uses the correlation function as a significance measure. Therefore, the selected subgraphs may be still highly correlated despite the low correlation threshold but they may be less frequent. For example, assume a pattern π which is found once or twice in almost all examples of class c but not found in other examples. The pattern π is considered to be a correlated pattern but it is not frequent and may not be found in test examples. In summary, the last part of the mining algorithm compensates a lower correlation threshold by selecting the most correlated and not redundant pattern. Although the correlation measure and pattern support are somewhat related (see Theorem 4.1), a lower threshold for the subgraph support may not be fully recovered by the other parts. Consequently, the selection of support threshold should be done carefully, that allows a sufficient number of subgraphs to satisfy the frequency criterion considering the redundancy between patterns.

The classification performance of the bag-of-words model in comparison to the graph mining algorithm can be seen in Table 6.3. As seen in the table, the

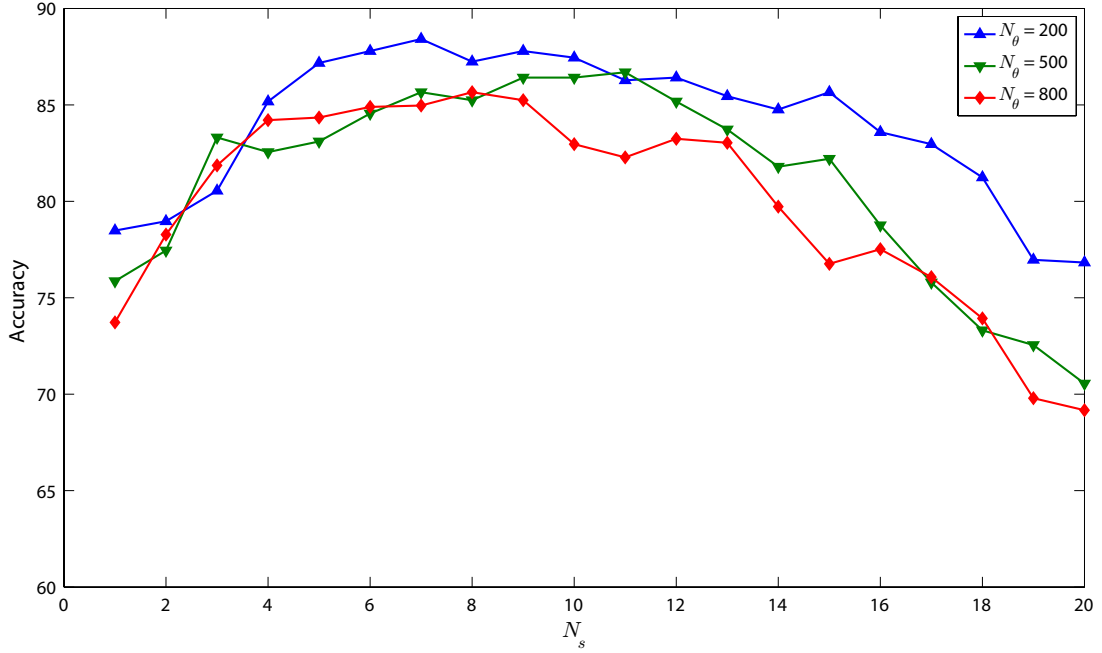


Figure 6.4: Plot of classification accuracy of the graph mining algorithm for three different N_θ values over the number of subgraphs per class. The lines are drawn by averaging the accuracy values for the parameters $N_\ell \in \{18, 26, 36, 54, 72\}$.

frequency of visual words in the bag-of-words model is less affected from the number of node labels because the graph mining method seek structural elements in the images. On the hand, bag-of-words model fails to classify images for small number of node labels. The graph mining is less affected from the confusion using the spatial relationships between local image patches. The best classification accuracies of both methods are almost same. However, further improvement cannot be obtained because of the intrinsic properties of the dataset. The small tile size due to the heterogeneous content of the satellite image limits the frequency of the subgraphs and the method suffers from the low subgraph support. Using bigger tiles might improve the performance of the graph mining method.

The confusion matrices of the graph mining algorithm and the bag-of-words models for the best parameter sets are presented in Figures 6.5 and 6.6, respectively. The most confused class in both matrices is greenhouses. It is about another problem of the dataset. Greenhouses are naturally located near sparse residential areas (villages) and orchards. Therefore, the images of greenhouses

Table 6.3: Classification accuracy of the bag-of-word model and the mining algorithm, in percentage terms, for different number of words/labels.

#Labels (N_ℓ)	BoW Accuracy	Max. Mining Accuracy
18	79.310	88.966 ($N_s = 10, N_\theta = 500$)
26	92.414	91.034 ($N_s = 7, N_\theta = 200$)
36	91.035	92.069 ($N_s = 9, N_\theta = 200$)
54	91.035	88.621 ($N_s = 9, N_\theta = 200$)
72	91.724	86.897 ($N_s = 10, N_\theta = 200$)

in the dataset are not completely homogeneous and contain structures belonging to other classes as seen in Figure 6.7. Our graph mining algorithm essentially handles such problems by mining class correlated subgraphs. It seeks a set of subgraphs which are commonly found among the examples of the class and some images having other structures do not constitute an important problem for the class. However, greenhouses are located sparsely in the Antalya image and almost all greenhouse images contain structures of orchards or sparse residential areas. As a result, the mining algorithm treats those structures as if they are correlated with the greenhouses class. This explains the reason for the relatively lower performance on the greenhouses and puts the obstacles in the way of improving the general performance of the mining algorithm. Sample images for each class are shown in Figure 6.7. They are grouped according to their classification result by the graph mining algorithm.

Finally, we apply the SVM model learned from the training set to every tile in the Antalya image. The classification result is drawn in Figure 6.8. Similarly, we discover themes in the Antalya image using the LDA model whose parameters are estimated from the training dataset for 12 themes. The set of subgraphs used in the LDA model is the set $\mathcal{S}$ which is found by the graph mining algorithm. The LDA model gives insight into distributions of themes over the whole image. Hence, heterogeneous tiles contents are analyzed in a better manner. However, presenting all topics distribution in this study for the Antalya image is a difficult task. Therefore, we combine this distributions into one image in Figure 6.9 where each color represents a region where the corresponding theme dominates other

dense-large	0.95	0.05	0.0	0.0	0.0	0.0	0.0	0.0
dense-small	0.03	0.97	0.0	0.0	0.0	0.0	0.0	0.0
dense-trees	0.0	0.06	0.88	0.0	0.0	0.0	0.0	0.06
sparse	0.0	0.0	0.0	0.88	0.08	0.0	0.0	0.04
greenhouses	0.0	0.0	0.0	0.0	0.71	0.18	0.0	0.12
orchards	0.0	0.0	0.0	0.0	0.0	0.95	0.02	0.02
forests	0.0	0.0	0.0	0.3	0.0	0.03	0.95	0.0
fields	0.0	0.0	0.0	0.0	0.0	0.10	0.0	0.90
	dense-large	dense-small	dense-trees	sparse	greenhouses	orchards	forests	fields

Figure 6.5: The confusion matrix of the graph mining algorithm using the parameters $N_\ell = 36$, $N_\theta = 200$ and $N_s = 9$. Class names are given in short: *sparse* and *dense* are used for sparse and dense residential areas, respectively. Also, *large* and *small* mean large and small buildings, respectively.

dense-large	0.90	0.10	0.0	0.0	0.0	0.0	0.0	0.0
dense-small	0.0	1.00	0.0	0.0	0.0	0.0	0.0	0.0
dense-trees	0.0	0.0	0.94	0.06	0.0	0.0	0.0	0.0
sparse	0.0	0.0	0.04	0.96	0.0	0.0	0.0	0.0
greenhouses	0.0	0.0	0.0	0.12	0.76	0.06	0.0	0.06
orchards	0.0	0.0	0.04	0.0	0.02	0.91	0.02	0.0
forests	0.0	0.0	0.03	0.0	0.0	0.0	0.97	0.0
fields	0.0	0.0	0.0	0.0	0.03	0.08	0.0	0.90
	dense-large	dense-small	dense-trees	sparse	greenhouses	orchards	forests	fields

Figure 6.6: The confusion matrix of the bag-of-words model for 26 labels. Class names are given in short: *sparse* and *dense* are used for sparse and dense residential areas, respectively. Also, *large* and *small* mean large and small buildings, respectively.

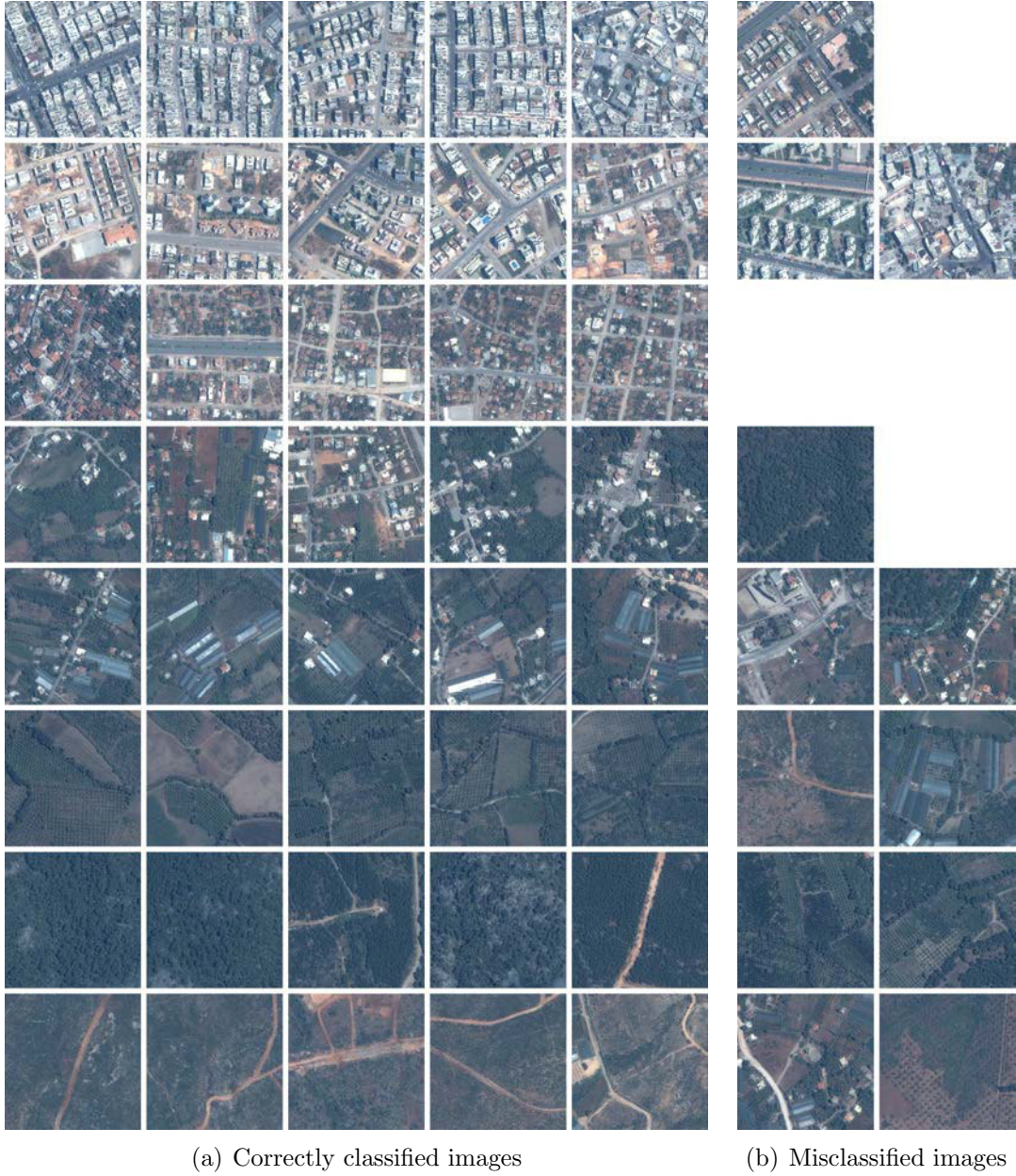


Figure 6.7: Sample images from the dataset. The images at the left are correctly classified by the graph mining algorithm while the images at right-hand side are misclassified using the parameters $N_\ell = 36$, $N_\theta = 200$ and $N_s = 9$. The image classes from top to down are in the order: dense residential areas with large buildings, dense residential areas with small buildings, dense residential areas with trees, sparse residential areas, greenhouses, orchards, forests and fields.

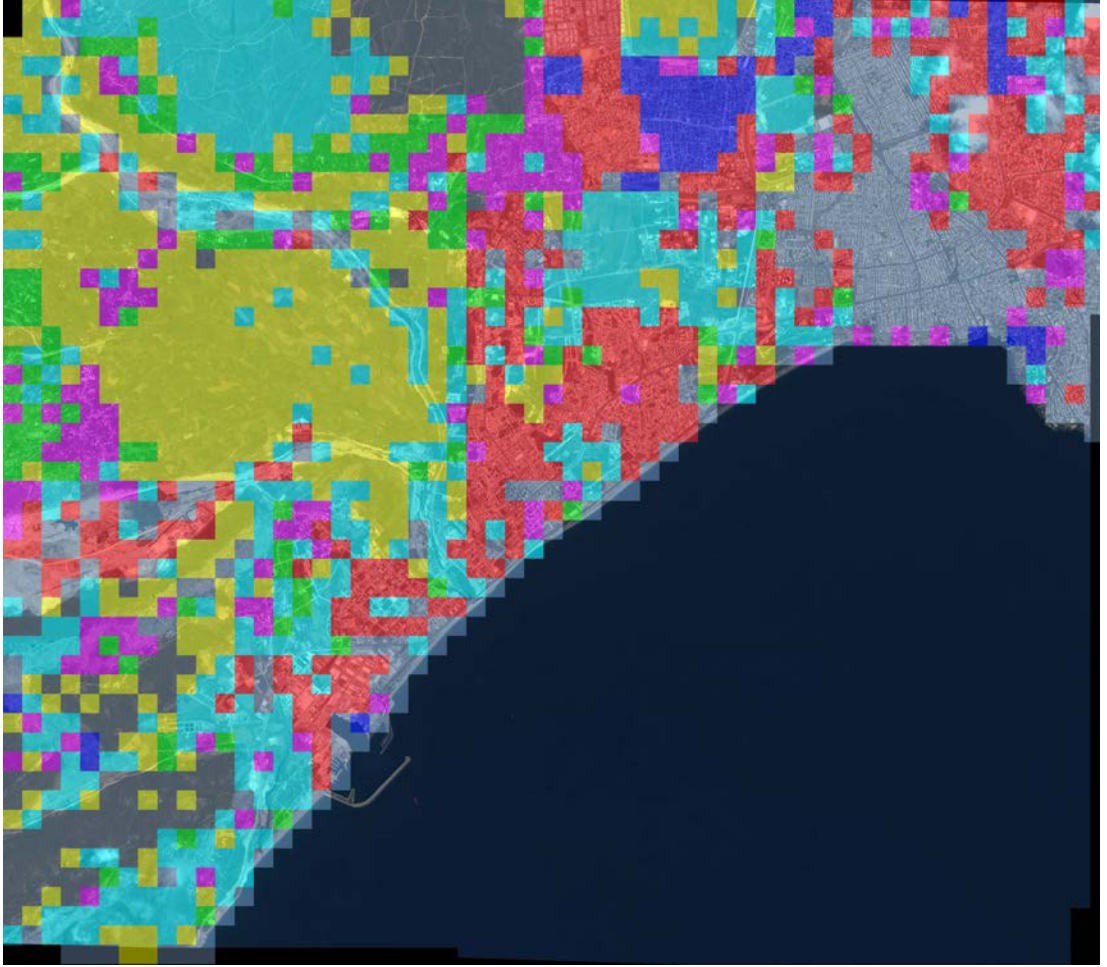


Figure 6.8: The classification of all tiles except sea using the SVM learned from the training set for the parameters $N_\ell = 36$, $N_\theta = 200$ and $N_s = 9$. Each color represents a unique class.

themes. The probability distributions of the most dominating 6 themes found by the LDA model trained for 16 themes are drawn in Figure 6.10. The further extensions of the LDA model remain as the future work for this study such as theme localization by analyzing the locations where the correlated subgraphs are found and testing with the model with other subgraph sets.

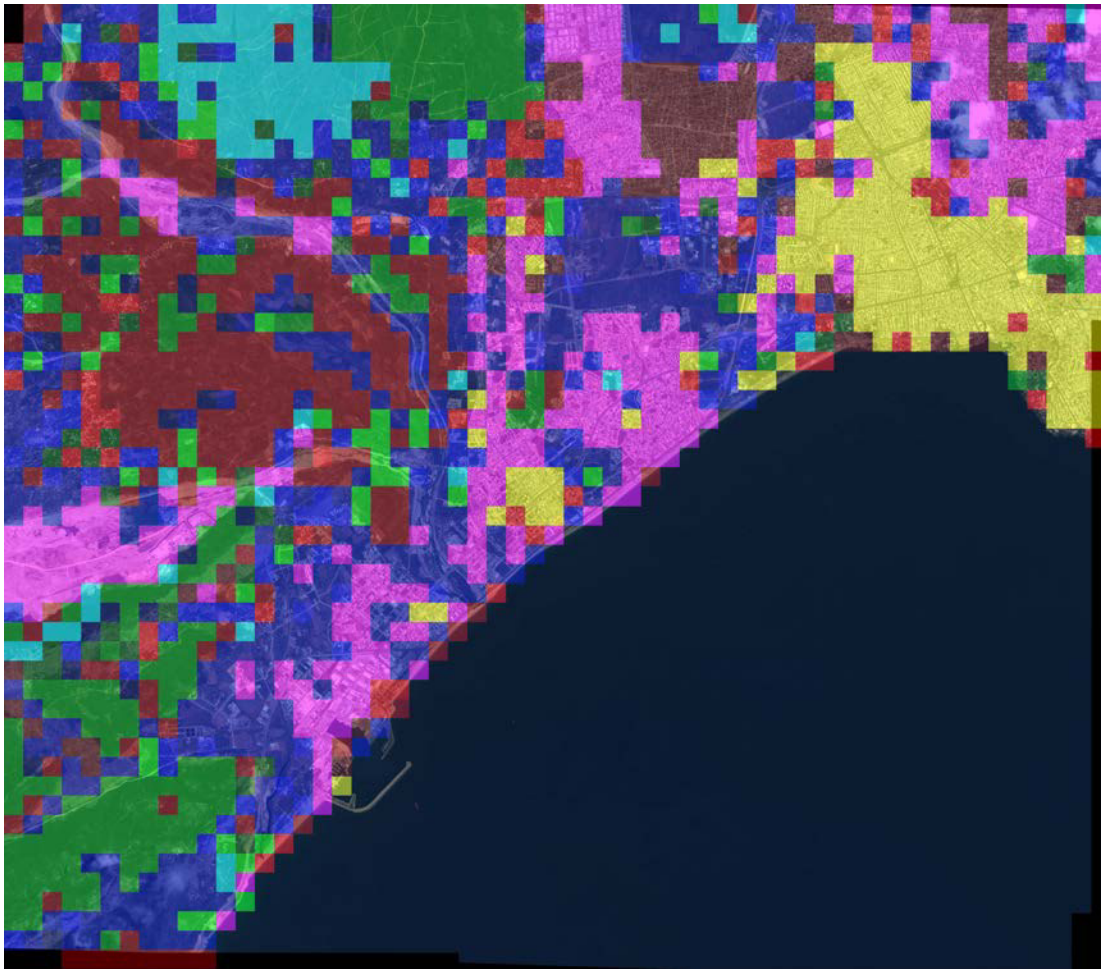


Figure 6.9: Every tile is labeled by a unique color which indicates the corresponding theme that dominates the other themes in that tile. The theme distributions are inferred from the LDA model for 12 themes. The subgraph set is the one mined in the previous experiments for the best parameters.

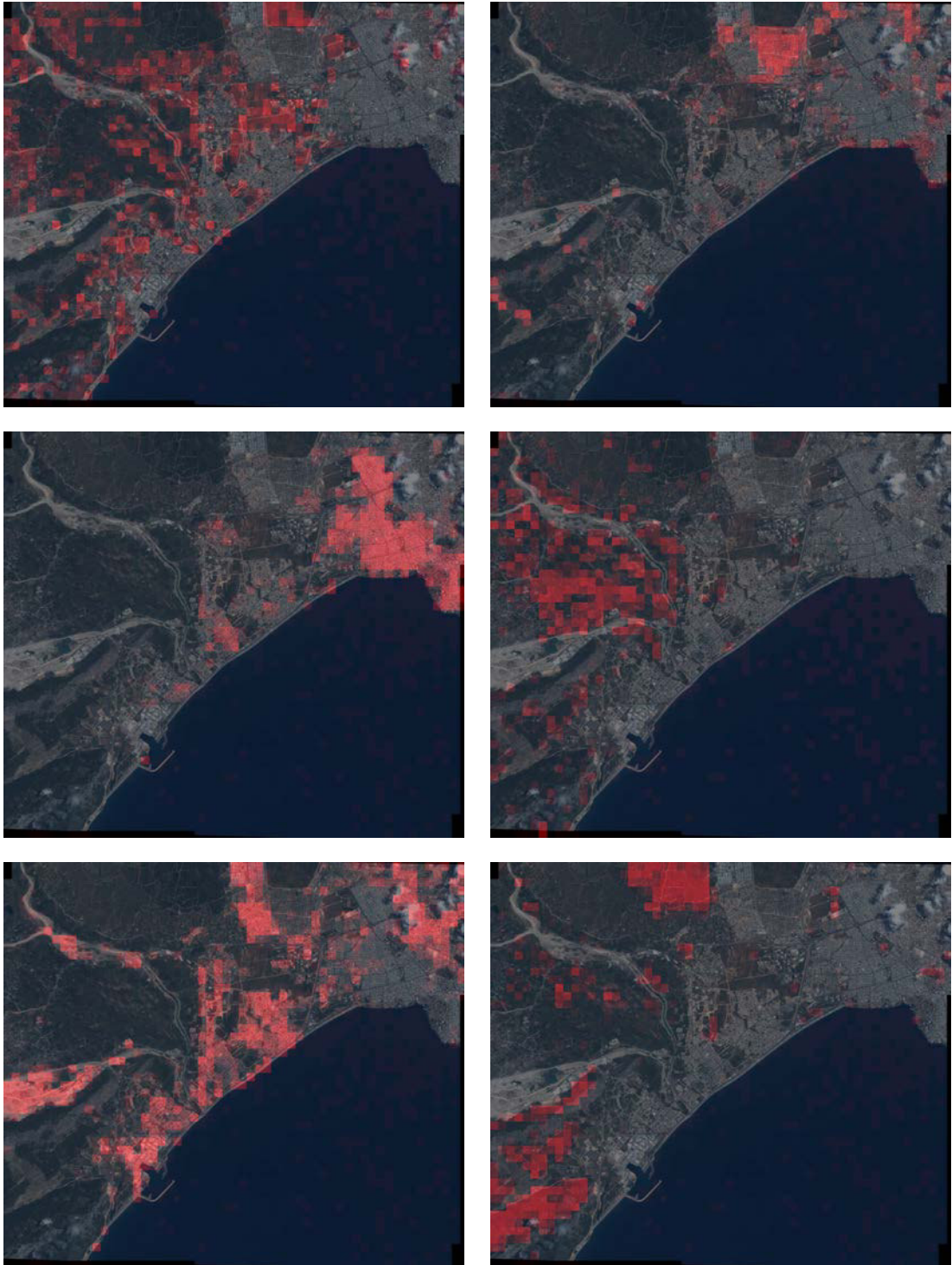


Figure 6.10: The most dominating 6 themes are shown, found by the LDA model trained for 16 themes. The intensity of red color represents the probability of the theme in an individual tile.

Chapter 7

Conclusions and Future Work

*It would seem that perfection is attained
not when there is nothing more to add,
but when there is nothing left to take away.*

“WIND, SAND AND STARS” – ANTOINE DE SAINT-EXUPÉRY

7.1 Conclusions

We emphasized the importance of high-level understanding of the image content through compound structures and we discussed the challenges of detecting compound structures. Accordingly, we described a new image content representation using the histogram of a subgraph set for classifying complex scenes such as dense and sparse urban areas. As the first step of this method, we transformed images to graphs where the nodes store local image content and the edges encode spatial information. We proposed a graph construction method where the patches, encoded by the graph nodes, are detected using maximally extremal stable regions and discriminative information about these regions are obtained by extracting features from these regions in relation to their surroundings. The features were quantized to form a codebook of local information that determined the node labels and the neighborhood relation between local patches were found from the

Voronoi tessellations of the patches.

In the second step of the method, we selected the subgraphs which are used in the histogram representation of images where the subgraphs encapsulate the local patches and their spatial arrangements within a specific structure. We described a graph mining algorithm to find the set of frequent and discriminative subgraphs which also has low redundancy. The algorithm first mines the frequent subgraphs in the image graph set. Then, it selects the discriminative subgraphs among the frequent ones with respect to the correlation between the subgraphs and the classes. We introduced a novel algorithm for extracting class-correlated patterns using the probabilistic model of subgraph frequency in an individual graph. Finally, the redundancy between the subgraphs in the set was resolved by choosing the most significant subgraphs considering the distances between them.

The third step of the method is the model learning from the vector space representations of the images. A multi-class support vector machine was employed for classifying the images. Furthermore, the latent Dirichlet allocation model was used for further classification of images. The LDA model provides the theme distribution representation of images computed from the subgraph histograms. Themes provide a better understanding of the images having heterogeneous content.

In experimental work, we evaluated the performance of the graph mining algorithm in image classification compared to the bag-of-words model. The dataset consists of tiles cut from an Ikonos image of Antalya image and each tile is labeled by one of the eight high-level semantic scene types. The classification accuracy of the graph mining algorithm shows the effectiveness of the proposed method in classification of complex scene types. We observed and discussed the effects of the parameters on the classification accuracy. We concluded that the graph mining algorithm is capable of discriminating images of different scene types successfully. Furthermore, the LDA model manages to discover interesting themes in the whole satellite image.

7.2 Future Work

The proposed image representation can be easily adapted to other application areas of computer vision. To illustrate, the application of subgraph histogram representation can be extended to image retrieval by defining a distance measure between subgraph histograms or borrowing a distance definition from the information retrieval literature. Given an input image, the most similar images in the dataset are the closest ones according to the distance function.

The LDA model offers new applications of the proposed image representation. Given a given graph set and a subgraph set, the LDA model finds the subgraph-theme probabilities β . A subgraph that is highly correlated with a theme can be used for localizing the theme distribution in a heterogeneous content. This enables high-level partitioning of heterogeneous images. Another application of the LDA model is unsupervised partitioning of the whole satellite image using subgraph histograms in case the labeled data are not available or the scene types are unknown. In such cases, the subgraph set contains all subgraphs generated by the graph language for a fixed-size. Given the number of themes, the LDA tries to discover themes from the whole image using the occurrence numbers of all subgraphs in tiles.

Finally, the set of subgraphs selected by the graph mining algorithm from an Ikonos image of Antalya can be used for classification of images cut from another satellite image which is retrieved from another satellite in a different spatial-resolution, by the help of a mapping function. The function maps the features extracted from the MSERs of the second image to the node labels determined for the first image. The parameters of the graph construction method should be adjusted for the spatial resolution of the second image, then the mapping function determines, after normalization, the closest cluster center of the first image to the features of an MSER in the second image.

Bibliography

- [1] S. Aksoy. Modeling of remote sensing image content using attributed relational graphs. In D. Yeung, J. T. Kwok, A. Fred, F. Roli, and D. Ridder, editors, *Structural, Syntactic, and Statistical Pattern Recognition*, volume 4109, chapter 52, pages 475–483. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.
- [2] S. Aksoy. *Signal and Image Processing for Remote Sensing*, chapter Spatial techniques for image classification, pages 491–513. Taylor & Francis, 2006.
- [3] D. Arthur and S. Vassilvitskii. k-means++: the advantages of careful seeding. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 1027–1035, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics.
- [4] E. Barbu, P. Hroux, S. Adam, and E. Trupin. Clustering document images using graph summaries. In P. Perner and A. Imiya, editors, *Machine Learning and Data Mining in Pattern Recognition*, volume 3587 of *Lecture Notes in Computer Science*, pages 194–202. Springer, 2005.
- [5] C. Berge. *Hypergraphs*. North-Holland Mathematical Library, 1989.
- [6] J. Bilmes. A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Technical report, International Computer Science Institute, 1998.
- [7] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

- [8] C. Borgelt. MoSS: Molecular substructure miner. <http://www.borgelt.net/moss.html>, 2009.
- [9] J. Bourgain. On lipschitz embedding of finite metric spaces in hilbert space. *Israel Journal of Mathematics*, 52(1):46–52, March 1985.
- [10] B. Bringmann. *Mining Patterns in Structed Data*. PhD thesis, Katholieke Universiteit Leuven, Celestijnenlaan 200 A – B-3001 Leuven, Belgium, 2009.
- [11] H. Bunke, P. Dickinson, M. Kraetzl, M. Neuhaus, and M. Stettler. *Applied Pattern Recognition*, chapter Matching of Hypergraphs – Algorithms, Applications, and Experiments, pages 131–154. Springer Berlin / Heidelberg, 2008.
- [12] H. Bunke and K. Riesen. Graph classification based on dissimilarity space embedding. In *Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, pages 996–1007, Berlin, Heidelberg, 2008. Springer-Verlag.
- [13] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [14] H. Cheng, X. Yan, J. Han, and C. wei Hsu. Discriminative frequent pattern analysis for effective classification. In *IEEE International Conference on Data Engineering*, pages 716–725, 2007.
- [15] K. W. Church and W. A. Gale. Poisson mixtures. *Natural Language Engineering*, 1:163–190, 1995.
- [16] M. Deshpande, M. Kuramochi, N. Wale, and G. Karypis. Frequent substructure-based approaches for classifying chemical compounds. *IEEE Transactions on Knowledge and Data Engineering*, 17(8):1036 – 1050, aug. 2005.
- [17] E. Dogrusoz and S. Aksoy. Modeling urban structures using graph-based spatial patterns. In *IEEE International Geoscience and Remote Sensing Symposium*, pages 4826 –4829, 23-28 2007.

- [18] M. Fiedler and C. Borgelt. Subgraph support in a single large graph. In *IEEE International Conference on Data Mining Workshops*, pages 399–404, Washington, DC, USA, 2007. IEEE Computer Society.
- [19] R. Gaetano, G. Scarpa, and G. Poggi. Hierarchical texture-based segmentation of multiresolution remote-sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 47(7):2129–2141, july 2009.
- [20] X. Gao, B. Xiao, D. Tao, and X. Li. Image categorization: Graph edit distance+edge direction histogram. *Pattern Recognition*, 41(10):3179–3191, 2008.
- [21] Z. Harchaoui and F. Bach. Image classification with segmentation graph kernels. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.
- [22] G. Ivncsy, R. Ivncsy, and I. Vajk. Graph mining-based image indexing. In *International Symposium of Hungarian Researchers on Computational Intelligence*, pages 313–323, Budapest, Hungary, Nov 2004.
- [23] H. Jiang and C. W. Ngo. Image mining using inexact maximal common subgraph of multiple args. In *International Conference on Visual Information Systems*, 2003.
- [24] Y. G. Jiang, J. Yang, C. W. Ngo, and A. G. Hauptmann. Representations of keypoint-based semantic concept detection: A comprehensive study. *IEEE Transactions on Multimedia*, 12(1):42–53, November 2009.
- [25] M. Kuramochi and G. Karypis. Finding frequent patterns in a large sparse graph. *Data Mining and Knowledge Discovery*, 11(3):243–271, 2005.
- [26] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2169–2178, 2006.

- [27] E. Levina and P. Bickel. The earth mover's distance is the mallows distance: some insights from statistics. In *IEEE International Conference on Computer Vision*, volume 2, pages 251–256 vol.2, 2001.
- [28] F.-F. Li and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 524–531, Washington, DC, USA, 2005. IEEE Computer Society.
- [29] T. Li, T. Mei, I.-S. Kweon, and X.-S. Hua. Contextual bag-of-words for visual categorization. *IEEE Transactions on Circuits and Systems for Video Technology*, PP(99):1 –1, Jan 2010.
- [30] H. Mannila and H. Toivonen. Levelwise search and borders of theories in knowledge discovery. *Data Mining and Knowledge Discovery*, 1(3):241–258, 1997.
- [31] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *British Machine Vision Conference*, 2002.
- [32] R. Myers, R. Wison, and E. Hancock. Bayesian graph edit distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(6):628 –635, jun 2000.
- [33] S. Nijssen, T. Guns, and L. De Raedt. Correlated itemset mining in roc space: a constraint programming approach. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 647–656, New York, NY, USA, 2009. ACM.
- [34] S. Nijssen and J. N. Kok. Multi-class correlated pattern mining. In *Knowledge Discovery in Inductive Databases*, pages 165–187, 2005.
- [35] S. Nowozin, K. Tsuda, T. Uno, T. Kudo, and G. H. Bakir. Weighted substructure mining for image analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

- [36] E. Pekalska and R. P. W. Duin. *The Dissimilarity Representation for Pattern Recognition: Foundations And Applications (Machine Perception and Artificial Intelligence)*. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 2005.
- [37] K. Riesen and H. Bunke. Iam graph database repository for graph based pattern recognition and machine learning. In *Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, pages 287–297, Berlin, Heidelberg, 2008. Springer-Verlag.
- [38] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover’s distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, November 2000.
- [39] G. Scarpa, R. Gaetano, M. Haindl, and J. Zerubia. Hierarchical multiple markov chain model for unsupervised texture segmentation. *IEEE Transactions on Image Processing*, 18(8):1830 –1843, aug. 2009.
- [40] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering object categories in image collections. In *IEEE International Conference on Computer Vision*, 2005.
- [41] P. Soille. *Morphological Image Analysis: Principles and Applications*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2003.
- [42] M. Stasolla and P. Gamba. Spatial indexes for the extraction of formal and informal human settlements from high-resolution sar images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 1(2):98 –106, june 2008.
- [43] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. <http://www.vlfeat.org/>, 2008.
- [44] L. Vincent and E. Dougherty. *Digital Image Processing Methods*, chapter Morphological Segmentation for Textures and Particles, pages 43–102. CRC, 1994.

- [45] D. Xin, Cheng, Hong, Yan, Xifeng, and J. Han. Extracting redundancy-aware top-k patterns. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 444–453, New York, NY, USA, 2006. ACM.
- [46] J. Yuan, Y. Wu, and M. Yang. Discovery of collocation patterns: from visual words to visual phrases. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, Jun 2007.
- [47] D. Zamalieva. Unsupervised detection of compound strctures using image segmentation and graph-based texture analysis. Master’s thesis, Bilkent University, August 2009.

Distance Functions: Theory, Algorithms and Applications

Thesis submitted for the degree of
“Doctor of Philosophy”

By
Ofir Pele

Submitted to the Senate of the Hebrew University
July, 2011

This work was carried out under the supervision of:
Prof. Michael Werman

Acknowledgments

It is difficult to overstate my gratitude to my Ph.D. supervisor, Prof. Michael Werman with his encouragement, support, and his great efforts through the research and most importantly his passion for research. Mike helped to make this thesis feasible and was a friend through the process.

I also thank my many friends in the corridor and lab with whom it was always fun to talk and discuss various issues.

In addition, I would like to thank many persons at the School of Engineering and Computer Science, both academic and administrative staff, for creating a stimulating environment to learn and grow. Special thanks to Prof. Shmuel Peleg for his priceless advises.

Lastly, and most importantly, I wish to thank my family for their encouragement, support, and love. My late father, whose dream was to see his children study. My mother, who did everything she could to allow her children to realize their full potential. My brothers, Oved and Eli for their encouragement, great support, and love. My wife, Liat, and my children, Itamar and Maayan who are the source of my happiness. Liat, you are the most perfect life partner I can imagine. I dedicate this thesis to them.

Abstract

Distance functions are at the core of numerous computer vision and machine learning tasks. For example, a multiple view geometry application (such as homography estimation), often includes a descriptor matching step. First, small image patches are selected. Then, descriptors (vectors) are computed to describe these patches. To decide if two descriptors (from two different images) are from the same real-world location the descriptors are compared using a distance function. The choice of the distance function determines both the accuracy and the speed of the method. Distance functions are also used for image retrieval and determine both the percent of similar images returned and the time it takes to return them. In machine learning, k-nearest neighbor classification performance is effected by the chosen distance function.

This thesis introduces several new distance functions and investigates their properties. Special emphasis is on practical applicability with theoretical insights. The thesis presents efficient algorithms to compute several distances. The distances are designed to be robust to common image noise such as occlusion, geometrical transforms, light changes and non-rigid deformations. On the other hand, they are designed to be as distinctive as possible. Our proposed methods have been successfully used both by computer vision researchers and by researchers in other fields. The success of the methods in other fields is probably because the noise characteristics in those fields are similar to image noise characteristics.

The first and second parts of the thesis are devoted to the Earth Mover's Distance (EMD) [1]. The EMD is a generalization of the transportation distance (also known as Monge-Kantorovich, Mallows, 1st Wasserstein and match distance) for non-normalized histograms. We introduce a new generalization of the transportation distance for non-normalized histograms called, $\widehat{EMD}$. Unlike the classic EMD, $\widehat{EMD}$ is a metric even when the histograms are non-normalized. Additionally, in practice, $\widehat{EMD}$ is often better than EMD. We also show that $\widehat{EMD}$ is a generalization of the L_1 distance.

Histograms of oriented gradient descriptors are ubiquitous tools in numerous computer vision tasks. Unfortunately, the distances that are used to match such descriptors do not take into account the special structure of these histograms. One problem with these distances is that they only compare corresponding bins of histograms. Although there are cross-bin distances that take into account the amount of similarity between all bins, they are not adequate to the histograms of oriented gradients that are computed from images. Additionally, most cross-bin distances are too slow. We present a

linear time algorithm for the comparison of orientation histograms. This method outperforms state-of-the-art distances on a widely used, publicly available benchmark.

In the second part of the thesis we present a robust family of Earth Mover’s Distances which is applicable to any type of histograms (not just orientation histograms). We developed an algorithm that computes these robust EMDs by an order of magnitude faster than the original approach. The resulting method has excellent image retrieval performance.

In the third part of the thesis, we present a new histogram distance family, the Quadratic-Chi (QC). QC members are Quadratic-Form distances with a cross-bin χ^2 -like normalization. The cross-bin χ^2 -like normalization reduces the effect of large bins having undue influence. Normalization was shown to be helpful in many cases, where the χ^2 histogram distance outperformed the L_2 norm. However, χ^2 is sensitive to quantization effects, such as caused by light changes, shape deformations etc. The Quadratic-Form part of QC members takes care of cross-bin relationships (*e.g.* red and orange), alleviating the quantization problem. We present two new cross-bin histogram distance properties: *Similarity-Matrix-Quantization-Invariance* and *Sparseness-Invariance* and show that QC distances have these properties. We also show experimentally that the properties improve performance. QC distances computation time complexity is linear in the number of non-zero entries in the bin-similarity matrix and can easily be parallelized. We present results for image retrieval and shape classification. We show that the new QC members outperforms state-of-the-art distances for these tasks, and have a short running time.

Finally, in the fourth part of the thesis, we describe a novel method for non-Mahalanobis metric learning denoted *Interpolated-Discretized Metric Learning*. This part of the work was motivated by the success of our non-Mahalanobis distance, the QC. A QC distance has parameters that can be learned. However, a serious limitation is that it can only model χ^2 -like distances. In addition, it is applicable only to non-negative vectors. Finally, it is non-convex with respect to its parameters, so learning them is hard. To overcome these problems we suggest a new non-Mahalanobis distance family, the *Interpolated-Discretized* (ID) distance family, and show how to efficiently learn its parameters. An ID distance can approximate any *unknown* distance. Many metric learning methods can be used with our ID distances. We demonstrate one possible way of learning ID distances using the Large Margin Nearest Neighbor (LMNN) framework [2]. With ID distances, the LMNN optimization problem is a linear program. Finally, we show that our method often outperforms other state-of-the-art metric learning approaches.

Contents

1	Introduction	1
1.1	Distance Functions Overview	1
1.1.1	Bin-to-Bin Distance Functions	1
1.1.2	Metric Properties	3
1.1.3	Cross-Bin Distances	3
1.2	Novel Aspects	7
2	A Linear Time Histogram Metric for Improved SIFT Matching	9
3	Fast and Robust Earth Mover’s Distances	24
4	The Quadratic-Chi Histogram Distance Family	33
5	Interpolated-Discretized Metric Learning using Linear Programming	48
6	Conclusions	57

Chapter 1

Introduction

Distance functions are at the core of numerous computer vision and machine learning tasks. For example, a multiple view geometry application (such as homography estimation), often includes a descriptor matching step. First, small image patches are selected. Then, descriptors (vectors) are computed to describe these patches. To decide if two descriptors (from two different images) are from the same real-world location the descriptors are compared using a distance function. On one hand, the distance function should be robust to common image noise or transformation (such as occlusion and light changes) so that the distance between two patches from the same real-world location will be small. On the other hand, the distance function should be as distinctive as possible, so that two patches from different real-world locations will be well separated. Additionally, we need efficient algorithms for the computation of the distance function as usually the number of distance computations is very large.

This thesis introduces several new distance functions and investigates their properties. Special emphasis is on theory that is built upon real-world assumptions. This introduction gives an overview of distance functions and their properties. Then, we describe the novel aspects of the thesis.

1.1 Distance Functions Overview

1.1.1 Bin-to-Bin Distance Functions

Bin-to-Bin distance functions compare corresponding bins of a vector to its exact corresponding bin in the second vector. That is, let $P, Q \in \mathbb{R}^N$ be two vectors, P_i is only compared to Q_i .

The most widely used distance family is the Minkowski-Form, also known

as L_p norms:

$$L_p(P, Q) = \left(\sum_i |P_i - Q_i|^p \right)^{\frac{1}{p}} \quad (1.1)$$

For $p \geq 1$, L_p is a metric (see section 1.1.2). The most widely used distances from this family are the L_1 distance also known as the Manhattan distance and the L_2 distance also known as the Euclidean distance.

The Kullback–Leibler (KL) divergence is a distance function for probability distributions (although it is used also for non-negative vectors which are not normalized). The KL divergence is defined as:

$$\text{KL}(P, Q) = \sum_i P_i \log \frac{P_i}{Q_i} \quad (1.2)$$

The KL divergence measures the expected number of extra bits required to code samples from P when using a code based on Q . It is non-symmetric. A serious limitation of the distance is that when $P_i \neq 0$ and $Q_i = 0$, KL equals infinity.

To overcome KL problems, the Jensen-Shannon (JS) divergence was suggested by Lin [3]. It is defined as:

$$\begin{aligned} \text{JS}(P, Q) &= \text{KL}(P, M) + \text{KL}(Q, M) \\ M &= \frac{P + Q}{2} \end{aligned} \quad (1.3)$$

The square root of a Jensen-Shannon divergence is a metric [4, 5] and it can be embedded isometrically as a subspace of a real Hilbert space [6]. Using the Taylor extension we get that JS is equal to [7]:

$$\text{JS}(P, Q) = \sum_{n=1}^{\infty} \frac{1}{2n(2n-1)} \sum_i \frac{(P_i - Q_i)^{2n}}{(P_i + Q_i)^{2n-1}} \quad (1.4)$$

where $n = 1$ corresponds to Taylor degrees of one and two and $n = 2$ correspond to Taylor degrees of three and four, etc. The first term is known in the computer vision community as χ^2 distance:

$$\chi^2(P, Q) = \frac{1}{2} \sum_i \frac{(P_i - Q_i)^2}{(P_i + Q_i)} \quad (1.5)$$

The practical results of χ^2 and JS are almost identical [8–10]. χ^2 is more efficiently computed than the JS. The square root of χ^2 is a metric [7].

The χ^2 histogram distance also emerged independently from the χ^2 test-statistic [11] where it is used to test the fit between a distribution and observed frequencies. χ^2 distance has recently received considerable attention in the computer vision literature. It has been used, for example, in state-of-the-art: contour detection and segmentation algorithms [12], descriptors matching [13], texture and object categories classification [14–17], near duplicate image identification [18] and shape classification [19, 20].

1.1.2 Metric Properties

There are four conditions for a distance function, $\mathcal{D}$, to be a *metric*:

1. $\mathcal{D}(P, Q) \geq 0$ (*non-negativity*).
2. $\mathcal{D}(P, Q) = 0$ if and only if $P = Q$ (*identity of indiscernibles*).
3. $\mathcal{D}(P, Q) = \mathcal{D}(Q, P)$ (*symmetry*).
4. $\mathcal{D}(P, Q) \leq \mathcal{D}(P, K) + \mathcal{D}(K, Q)$ (*subadditivity*).

Non-negativity is implied by the others: $2\mathcal{D}(P, Q) = \mathcal{D}(P, Q) + \mathcal{D}(Q, P) \geq \mathcal{D}(P, P) = 0$.

Two useful generalizations of metrics are the *semi-metric*, which drops the triangle inequality property and *pseudo-metric* which weakens the second condition to:

2. $\mathcal{D}(P, Q) = 0$ if $P = Q$ (but possibly $\mathcal{D}(P, Q) = 0$ for some $P \neq Q$).

1.1.3 Cross-Bin Distances

Bin-to-Bin distance functions such as L_2 , L_1 and χ^2 compare only corresponding bin's of a vector to its exact corresponding bin in the second vector. The assumption when using these distances is that the histogram domains are aligned. However this assumption is violated in many cases due to quantization, shape deformation, light changes, etc. Bin-to-bin distances depend on the number of bins. If it is low, the distance is robust, but not discriminative, if it is high, the distance is discriminative, but not robust. Distances that take into account cross-bin relationships (cross-bin distances) can be both robust and discriminative.

The Quadratic-Form Distance

Let $A \in \mathbb{R}^{N \times N}$ the bin-similarity matrix. That is, a_{ij} encodes how much bin i is similar to bin j . The *Quadratic-Form* (QF) distance is defined as [21]:

$$\text{QF}^A(P, Q) = \sqrt{(P - Q)^T A (P - Q)} \quad (1.6)$$

When the bin-similarity matrix A is the inverse of the covariance matrix, the QF distance is called the Mahalanobis distance [22]. If the bin-similarity matrix is positive-semidefinite, the A matrix can be expressed as $A = LL^T$ for some real matrix L . Thus, the distance can be computed as the Euclidean norm between linearly transformed vectors:

$$\text{QF}^A(P, Q) = \|LP - LQ\|_2 \quad (1.7)$$

In this case the QF distance is a psuedo-metric (as it is possible that $\text{QF}^A(P, Q) = 0$ for some $P \neq Q$). If A is positive-definitive, the QF distance is a metric.

Recently there have been a lot of research effort on learning a positive-semidefinite matrix, A , from labeled examples or given relative distances constraints [2, 23–31]. Although much of the metric-learning work has focused on the QF distance¹, this measure has some inherent limitations. The most important one is that it is a function only of the difference between input vectors $\vec{x}^i - \vec{x}^j$. Thus, it will give the same distance whenever two vectors have the same difference vector. For example, a QF distance will not be able to differentiate between $(\vec{x}^i = 42, \vec{x}^j = 40)$ and $(\vec{x}^i = 2, \vec{x}^j = 0)$. To overcome this limitation several metric learning methods have been proposed [2, 27, 29, 32, 33]. Chopra et al. [32] proposed to learn a convolutional neural net as a non-linear transformation before applying the ℓ_2 norm. Babenko et al. [33] suggested a boosting framework for learning non-Mahalanobis metrics. These methods are non-convex and thus they might suffer from local minimas and training is sensitive to parameters. Kernel methods were also proposed in order to learn a QF distance over non-linear transformations of the data [2, 27, 29]. Computing such a distance between two vectors scales linear in the number of training examples, which makes it impractical for large datasets.

The Transportation and Earth Mover’s Distances

The second type of distance that takes into account cross-bin relationships is the transportation distance (also known as Monge-Kantorovich, Mallows,

¹The QF distance is called Mahalanobis distance in machine learning community even when A is not the inverse of the covariance matrix.

1st Wasserstein and match distance) which is defined between probability distributions:

$$\begin{aligned}
\text{transportation}^D(P, Q) &= \min_{\{F_{ij}\}} \sum_{i,j} F_{ij} D_{ij} \quad s.t \\
\sum_j F_{ij} &= P_i \\
\sum_i F_{ij} &= Q_j \\
\sum_{i,j} F_{ij} &= \sum_i P_i = \sum_j Q_j = 1 \\
F_{ij} &\geq 0
\end{aligned} \tag{1.8}$$

where $\{F_{ij}\}$ denotes the flows. Each F_{ij} represents the amount transported from the i th supply to the j th demand. We call D_{ij} the *ground distance* between bin i and bin j . When D_{ij} is a metric the transportation distance is also a metric.

The transportation problem was formalized in 1781 by Gaspard Monge [34]. Major advances were made in the field by Leonid Kantorovich [35]. Consequently, the problem is sometimes known as the Monge–Kantorovich transportation problem. Wasserstein proposed to use the definition of the transportation problem as a distance function [36]. Mallows also studied this distance [37].

Early work using the transportation distances for histogram comparison in the computer vision community can be found in [38–41]. Shen and Wong [38] proposed to unfold two integer histograms, sort them and then compute the L_1 distance between the unfolded histograms. To compute the modulo matching distance between cyclic histograms they proposed taking the minimum from all cyclic permutations. This distance is equivalent to transportation distance between two normalized histograms. Werman et al. [39] showed that this distance is equal to the L_1 distance between the cumulative histograms. They also proved that matching two cyclic histograms by examining only cyclic permutations is in effect optimal. Werman et al. [40] proposed an $O(M \log M)$ algorithm for finding a minimal matching between two sets of M points on a circle. Peleg et al. [41] suggested using the EMD for grayscale images and using linear programming to compute it.

Rubner et al. [1] suggested a new generalization of the transportation distance to non-normalized histograms, which he called the Earth Mover’s Distance (EMD):

$$\begin{aligned}
\text{EMD}^D(P, Q) &= \min_{\{F_{ij}\}} \frac{\sum_{i,j} F_{ij} D_{ij}}{\min(\sum_i P_i, \sum_j Q_j)} \quad s.t \\
\sum_j F_{ij} &\leq P_i \\
\sum_i F_{ij} &\leq Q_j \\
\sum_{i,j} F_{ij} &= \min(\sum_i P_i, \sum_j Q_j) \\
F_{ij} &\geq 0
\end{aligned} \tag{1.9}$$

The Earth Mover's Distance is the least amount of work needed in order to transform the smaller histogram into *part* of the bigger histogram (best partial matching), normalized with minimum between the sums of the histograms. It is a metric only if the histograms are normalized (then it equals the transportation distance) and D_{ij} is a metric. Rubner et al. suggested using EMD for color and texture images. They computed the EMD using a specific linear programming algorithm - the transportation simplex. The algorithm worst case time complexity is exponential. Practical run time was shown to be super-cubic ($\Omega(N^3) \cap O(N^4)$). Interior-point algorithms or Orlin's algorithm [42] both have a time complexity of $O(N^3 \log N)$ and can also be used.

A major research effort has been done on computing EMD- L_1 , that is, EMD with L_1 as the ground distance. Ling and Okada [43] showed that if the points lie on a Manhattan network (*e.g.* an image), the number of variables in the LP problem can be reduced from $O(N^2)$ to $O(N)$. To execute the EMD- L_1 computation, they employed a tree-based algorithm, Tree-EMD. Tree-EMD exploits the fact that a basic feasible solution of the simplex algorithm-based solver forms a spanning tree when the EMD- L_1 is modeled as a network flow optimization problem. The worst case time complexity is exponential. Empirically, they showed that this algorithm has an average time complexity of $O(N^2)$. Gudmundsson et al. [44] also put forward this simplification of the LP problem. They suggested an $O(N \log^{d-1} N)$ algorithm that creates a Manhattan network for a set of N points in $\mathbb{R}^d$. The Manhattan network has $O(N \log^{d-1} N)$ vertices and edges. Thus, using Orlin's algorithm [42] the EMD- L_1 can be computed with a time complexity of $O(N^2 \log^{2d-1} N)$. Indyk and Thaper [45] proposed approximating EMD- L_1 by embedding it into the L_1 norm. Embedding time complexity is $O(Nd \log \Delta)$, where N is the feature set size, d is the feature space dimension and Δ is the diameter of the union of the two feature sets. Grauman and

Darrell [46] substituted L_1 with histogram intersection in order to approximate partial matching. Shirdhonkar and Jacobs [47] presented a linear-time algorithm for approximating EMD- L_1 for low dimensional histograms using the sum of absolute values of the weighted wavelet coefficients of the difference histogram. Khot and Naor [48] showed that any embedding of the EMD over the d -dimensional Hamming cube into L_1 must incur a distortion of $\Omega(d)$, thus losing practically all distance information. Andoni et al. [49] showed that for sets with cardinalities upper bounded by a parameter s , the distortion reduces to $O(\log s \log d)$. A practical reduction in accuracy due to the approximation was reported by [47, 50].

A major limitation of EMD- L_1 is its sensitivity to outliers. That is, L_1 , as a ground distance between bins, assign different outliers different distances. Thus, in real-world applications the accuracy of EMD- L_1 (and same for EMD with L_2 ground distance) is often very low (often even lower than simple L_1 or χ^2 distances) [10, 51–54].

1.2 Novel Aspects

The main novelties of this thesis and several successful applications are:

- I. We present $\widehat{EMD}$, a new generalization of the transportation distance for non-normalized histograms. Unlike the classic generalization (EMD), $\widehat{EMD}$ is a metric even when the histograms are non-normalized. In practice, $\widehat{EMD}$ is often better than EMD.
- II. We present a new linear time algorithm for the computation of $\widehat{EMD}$ or EMD between two normalized cyclic histograms (*e.g.* histograms of oriented gradients) with L_1 as the ground distance. However, we show that using this distance for image descriptors matching, the accuracy is low. We explain the theoretical reason which is sensitivity to outliers.
- III. We present a new linear time algorithm for the computation of a $\widehat{EMD}$ or EMD between two not necessarily normalized cyclic histograms with a robust ground distance. We show that this distance has excellent image descriptors matching performance on a widely used and publicly available benchmark. Gupta et al. [54] also reported excellent image descriptors matching results using our distance.
- IV. We present a new efficient algorithm for the computation of robust $\widehat{EMD}$ or EMD between general non-negative vectors. The algorithm is faster by an order of magnitude than the classic algorithm. We show

excellent results for image retrieval. This algorithm was successfully used by other researchers both from the computer vision community and from other communities. For example, Boltz et al. used it for superpixels matching. In a comparison study between various image retargeting algorithms, Rubinstein et al. [55] reported that our distance had the best correlation with human judgements of image perceptual similarity (between an image and its retargeted version). Macindoe and Richards [56] used our method for social graph comparisons.

- V. We present a new family of histogram distances, called *Quadratic-Chi* (QC). Members of this family are robust to quantization effects and have a χ^2 -like normalization. Excellent image retrieval and shape classification results are demonstrated. Xu et al. [57] reported state-of-the-art wood recognition results using our QC distances.
- VI. We present two new cross-bin distances properties *Similarity-Matrix-Quantization-Invariance* and *Sparseness-Invariance* and a proof that $\widehat{EMD}$ and QC distances have these properties. We show, experimentally, that these properties are practically important.
- VII. We present a new family of distances called *Interpolated-Discretized* (ID) distances. These distances can approximate any kind of bin-to-bin distance and that their parameters are easy to learn. We show that our method often outperforms state-of-the-art metric learning approaches.

I., II. and III. are presented in chapter 2, IV. is presented in chapter 3, V. and VI. are presented in chapter 4 and VII. is presented in chapter 5.

Chapter 2

A Linear Time Histogram Metric for Improved SIFT Matching

A Linear Time Histogram Metric for Improved SIFT Matching

Ofir Pele and Michael S. Werman

School of Computer Science and Engineering
The Hebrew University of Jerusalem
{ofirpele,werman}@cs.huji.ac.il

Abstract. We present a new metric between histograms such as SIFT descriptors and a linear time algorithm for its computation. It is common practice to use the L_2 metric for comparing SIFT descriptors. This practice assumes that SIFT bins are aligned, an assumption which is often not correct due to quantization, distortion, occlusion etc.

In this paper we present a new Earth Mover's Distance (EMD) variant. We show that it is a metric (unlike the original EMD [1] which is a metric only for normalized histograms). Moreover, it is a natural extension of the L_1 metric. Second, we propose a linear time algorithm for the computation of the EMD variant, with a robust ground distance for oriented gradients. Finally, extensive experimental results on the Mikolajczyk and Schmid dataset [2] show that our method outperforms state of the art distances.

1 Introduction

Histograms of oriented gradient descriptors [3 4 5 6] are ubiquitous tools in numerous computer vision tasks. One of the most successful is the scale invariant Feature Transform (SIFT) [3]. In a recent performance evaluation [2] the SIFT descriptor was shown to outperform other local descriptors. The SIFT descriptor has proven to be successful in applications such as object recognition [3 8 9] object class detection [10 11 12] image retrieval [13 14 15 16] robot localization [1 7] building panoramas [18] and image classification [19].

It is common practice to use the L_2 metric for comparing SIFT descriptors. This practice assumes that the histogram domains are aligned. However this assumption is violated through quantization shape deformation detector localization errors etc. Although the SIFT algorithm has steps that reduce the effect of quantization this is still a liability as can be seen by the fact that increasing the number of orientation bins negatively affects performance [3].

The Earth Mover's Distance (EMD) [1] is a cross bin distance that addresses this alignment problem. EMD is defined as the minimal cost that must be paid to transform one histogram into the other where there is a ground distance between the basic features that are aggregated into the histogram. There are two main problems with the EMD. First it is not a metric between non normalized histograms. Second for a general ground distance it has a high run time.

In this paper we present an earth Mover's distance (EMD) variant. We show that it is a metric if it is used with a metric ground distance. Second we present a linear time algorithm for the computation of the EMD variant with a robust ground distance for oriented gradients. Finally we present experimental results for FT matching on the Miola city and chmid dataset [2] showing that our method outperforms state of the art distances such as L_2 EMD L_1 [20] diffusion distance [21] and EMD_{MOD} [22–23].

This paper is organized as follows. Section 2 is an overview of previous work. Section 3 introduces the new EMD variant. Section 4 introduces the new FT metric and the linear time algorithm for its computation. Section 5 describes the experimental setup and Section 6 presents the results. Finally conclusions are drawn in Section 7.

2 Previous Work

Early work using cross bin distances for histogram comparison can be found in [22–24, 25, 26]. Ben and Long [24] proposed to unfold two integer histograms sort them and then compute the L_1 distance between the unfolded histograms. To compute the modulo matching distance between cyclic histograms they proposed taking the minimum from all cyclic permutations. This distance is equivalent to EMD between two normalized histograms. Werman et al. [25] showed that this distance is equal to the L_1 distance between the cumulative histograms. They also proved that matching two cyclic histograms by examining only cyclic permutations is in effect optimal. Ha and Rihari [2] rediscovered these algorithms and described a character writer identification application. Werman et al. [22] proposed an $O(M \log M)$ algorithm for finding a minimal matching between two sets of M points on a circle. The algorithm can be adapted to compute the EMD between two N bin normalized histograms with time complexity $O(N)$ (Appendix in [23]).

Leleg et al. [26] suggested using the EMD for grayscale images and using linear programming to compute it. Rubner et al. [1] suggested using EMD for color and texture images. They computed the EMD using a sequential linear programming algorithm the transportation simplex. The algorithm worst case time complexity is exponential. Practical run time was shown to be super cubic ($(N^3) \cap O(N^4)$). Interior point algorithms with time complexity $O(N^3 \log N)$ can also be used. All of these algorithms have high computational cost.

Indy and Thaler [28] proposed approximating the EMD by embedding it into an Euclidean space. Embedding time complexity is $O(Nd \log d)$ where N is the feature set size d is the feature space dimension and d is the diameter of the union of the two feature sets.

Recently Ling and Radford proposed general cross bin distances for histogram descriptors. The first is EMD L_1 [20] i.e. EMD with L_1 as the ground distance. To execute the EMD L_1 computation they propose a tree based algorithm Tree EMD. Tree EMD exploits the fact that a basic feasible solution of the simplex algorithm based solver forms a spanning tree when the EMD L_1 is modeled as a network flow optimization problem. The worst case time complexity is

exponential. empirically they show that this new algorithm has an average time complexity $O(N^2)$. Ling and Sedagadani also proposed the diffusion distance [21]. They defined the difference between two histograms to be a temperature field. The diffusion distance was derived as the sum of dissimilarities over scales. The algorithm run time is linear.

For a comprehensive review of EMD and its applications in computer vision we refer the reader to Ling and Sedagadani's survey [20].

3 The New EMD variant - $\widehat{EMD}$

This section introduces $\widehat{EMD}$ a new earth Mover's distance variant. We show that it is a metric (unlike the original EMD [1] which is a metric only for normalized histograms). Moreover it is a natural extension of the L_1 metric.

The earth Mover's distance (EMD) [1] is defined as the minimal cost that must be paid to transform one histogram into the other where there is a ground distance between the basic features that are aggregated into the histogram.

Given two histograms P, Q the EMD as defined by Rubner et al. [1] is

$$EMD(P, Q) = \min_{\{f_{ij}\}} \frac{\sum_{i,j} f_{ij} d_{ij}}{\sum_{i,j} f_{ij}} \quad s.t \quad (1)$$

$$\sum_j f_{ij} \leq P_i, \quad \sum_i f_{ij} \leq Q_j, \quad \sum_{i,j} f_{ij} = \min(\sum_i P_i, \sum_j Q_j), \quad f_{ij} \geq 0 \quad (2)$$

where $\{f_{ij}\}$ denotes the flows. Each f_{ij} represents the amount transported from the i th supply to the j th demand. We call d_{ij} the *ground distance* between bin i and bin j in the histograms.

We propose $\widehat{EMD}$

$$\widehat{EMD}_\alpha(P, Q) = (\min_{\{f_{ij}\}} \sum_{i,j} f_{ij} d_{ij}) + |\sum_i P_i - \sum_j Q_j| \times \alpha \max_{i,j} \{d_{ij}\} \quad s.t \quad \alpha \geq 0.2 \quad (3)$$

Note that for two probability histograms (*i.e.* total mass equal to one) EMD and $\widehat{EMD}$ are equivalent. However if the masses are not equal $\widehat{EMD}$ adds one supplier or demander such that the masses on both sides becomes equal. The ground distance between this supplier or demander to all other demanders or suppliers respectively is set to be α times the maximum ground distance. In addition the $\widehat{EMD}$ is not normalized by the total flow.

Note that $\widehat{EMD}$ with $\alpha = 0.5$ and with the rounded δ ground distance multiplied by two ($d_{ij} = 0$ if $i = j$ 2 otherwise) is equal to the L_1 metric.

If $\alpha \geq 0.5$ and the ground distance is a metric $\widehat{EMD}$ is a metric (unlike the original EMD [1] which is a metric only for normalized histograms). Proof is given in Appendix [23]. Being a metric can lead to more efficient data structures and search algorithms.

we now give two examples of when the usage of $\widehat{EMD}$ is more appropriate (both of which are the case for the ℓ_1 FT descriptors). The first is when the total mass of the histograms is important. For example let $P = (1, 0)$ $Q = (0, 1)$ $P' = (9, 0)$ $Q' = (0, 9)$. Using L_1 as a ground distance and $\alpha = 1$ $EMD(P, Q) = 1 = EMD(P', Q')$ while $\widehat{EMD}(P, Q) = 1 < 9 = \widehat{EMD}(P', Q')$. The second is when the difference in total mass between histograms is a distinctive cue. For example let $P = (1, 0)$ $Q = (1, 1)$. Using L_1 as a ground distance and $\alpha = 1$ $EMD(P, Q) = 0$ while $\widehat{EMD}(P, Q) = 1$.

4 The $SIFT_{DIST}$ Metric

This section introduces ℓ_1 FT_{DIST} a new metric between ℓ_1 FT descriptors. It is common practice to use the L_2 metric for comparing ℓ_1 FT descriptors. This practice assumes that the ℓ_1 FT histograms are aligned so that a bin in one histogram is only compared to the corresponding bin in the other histogram. This is often not the case due to quantization distortion occlusion etc. Our distance has three instead of two matching costs: zero cost for exact corresponding bins, one cost for neighboring bins and two cost for farther bins and for the extra mass. Thus the metric is robust to small errors and outliers.

The section first defines FT_{DIST} and then presents a linear time algorithm for its computation.

4.1 $SIFT_{DIST}$ Definition

This section first describes the ℓ_1 FT descriptor. Second it introduces Thresholded Modulo arth Mover's distance (M_{TMOD}) an M variant for oriented gradient histograms. Finally it defines the FT_{DIST} .

The ℓ_1 FT descriptor [3] is a $M \times M \times N$ histogram. Each of the $M \times M$ spatial cells contains an N bin histogram of oriented gradients. See Fig. 1 for a visualization of ℓ_1 FT descriptors.

Let $A = \{0, \dots, N-1\}$ be N points equally spaced on a circle. The modulo L_1 distance for two points $i, j \in A$ is

$$D_{MOD}(i, j) = \min(|i - j|, N - |i - j|) \quad (4)$$

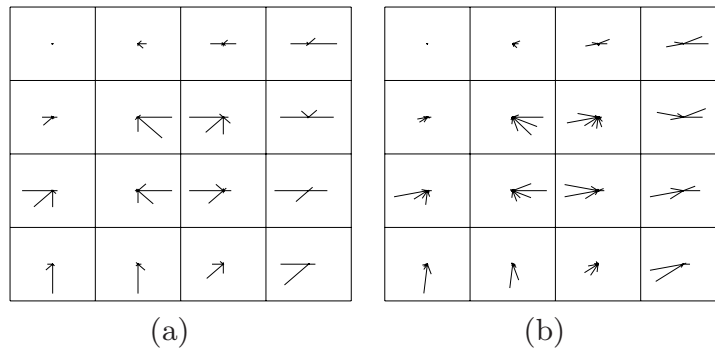


Fig. 1. (a) $4 \times 4 \times 8$ SIFT descriptor. (b) $4 \times 4 \times 16$ SIFT descriptor.

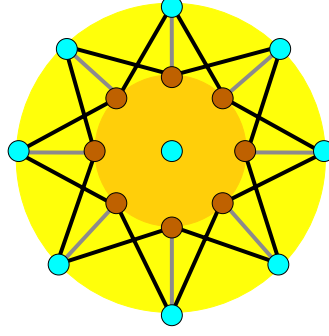


Fig. 2. The flow network of EMD_{TMODO} for $N = 8$. The brown vertexes on the inner circle are the bins of the “supply” histogram, P . The cyan vertexes on the outer circle are the bins of the “demand” histogram, Q . We assume without loss of generality that $\sum_i P_i \geq \sum_j Q_j$; thus we add one infinite sink in the middle. The short gray edges are *zero-cost* edges. The long black edges are *one-cost* edges. *Two-cost* edges that turn the graph into a full bi-partite graph between sinks and sources are not colored for visibility.

The thresholded modulo L_1 distance is defined as

$$D_{TDMO}(i, j) = \min(D_{MOD}(i, j), 2) \quad (5)$$

M_{TMODO} is defined as $\widehat{EMD}(\cdot, \cdot, 3)$ with ground distance $D_{TDMO}(i, j)$ and $\alpha = 1$. D_{TDMO} is a metric. It follows from Appendix [23] that M_{TMODO} is also a metric. Using M_{TMODO} the transportation cost to the two nearby bins is 1 while for farther bins and for the extra mass it is 2 (see Fig. 2). For example for $N = 16$ we assume that all differences larger than 22.5° are caused by outliers and should be assigned the same transportation cost.

The FT_{DIST} between two FT descriptors is defined as the sum over the M_{TMODO} between all the $M \times M$ oriented gradient histograms. FT_{DIST} is also an M where edges between spatial bins have an infinite cost.

$\widehat{EMD}(\cdot, \cdot, 3)$ has two advantages over Hubner’s $M(\cdot, \cdot, 1)$ for comparing FT descriptors. First the difference in total gradient magnitude between FT spatial cells is an important distinctive cue. Using Hubner’s definition this cue is ignored. Second $\widehat{EMD}$ is a metric even for non normalized histograms.

4.2 A Linear Time $SIFT_{DIST}$ Algorithm

Since FT_{DIST} is a sum of M_{TMODO} solutions we present a linear time algorithm for M_{TMODO} .

Like all other earth Mover’s instances M_{TMODO} can be solved by a maximum min cost algorithm. Each bin i in the first histogram is connected to bin i in the second histogram with a *zero-cost* edge to two nearby bins with *one-cost* edges and to all other bins with *two-cost* edges. See Fig. 2 for an illustration of this flow network.

The algorithm starts by saturating all *zero-cost* edges. Since the ground distance $(\cdot, \cdot, 5)$ obeys the triangle inequality this step does not change the minimum cost

solution [22]. Note that after the first step finishes from each of the supplier demanders pairs that were connected with a *zero-cost edge* either the supplier is empty or the demander is full.

To minimize the cost after the first step the algorithm needs to maximize the flow through the *one-cost edges* *i.e.* the problem becomes a max flow problem on a graph with N vertexes and at most N edges where the maximum path length is 1 as flow goes only from supply to demand (see Fig. 2). This step starts by checking whether all vertexes are degree 2 if yes we remove an arbitrary edge. Second we traverse the suppliers vertexes clockwise twice. For each degree one vertex we saturate its edge. If we did not remove an edge at the beginning of this step there will be no augmenting paths. If an edge was removed the algorithm flows through all augmenting paths. All these paths can be found by returning the edge and extending from it.

The algorithm finishes by flowing through all *two-cost edges* which is equivalent to multiplying the maximum of the total remaining supply and demand by two and adding it to the distance.

5 Experimental Setup

We evaluate FT_{DIST} using a test protocol similar to that of Miola et al. and Schmid [2]. The dataset was downloaded from [29]. The test data contain eight folders each with six images with different geometric and photometric transformations and for different scene types. Fig. 3 shows the first and the third image from each folder. Six image transformations are evaluated: viewpoint change, scale and rotation, image blur, light change and compression. The images are either of planar scenes or the camera position was fixed during acquisition. The images are therefore always related by a homography. The ground truth homographies are supplied with the dataset. For further details about the dataset we refer the reader to [2].

The evaluation criterion is based on the number of correct and false matches obtained for an image pair. The match definition depends on the matching strategy. The matching strategy we use is a symmetric version of Lowe's ratio matching [3]. Let $a \in A$ and $b \in B$ be two descriptors and $n(a, A)$ and $n(b, B)$ be their spatial neighbors that is all descriptors in A and B respectively such that the ratio of the intersection and union of their regions with a and b respectively is larger than 0.5. a and b are matched if all the following conditions hold: $a = \arg \min_{a' \in A} D(a', b)$, $a_2 = \arg \min_{a' \in A \setminus n(a, A)} D(a', b)$, $b = \arg \min_{b' \in B} D(a, b')$, $b_2 = \arg \min_{b' \in B \setminus n(b, B)} D(a, b')$ and $\min \left\{ \frac{D(a_2, b)}{D(a, b)}, \frac{D(a, b_2)}{D(a, b)} \right\} \geq \tau$. τ value is varied to obtain the curves. Note that for $\tau = 1$ the matching strategy is the symmetric nearest neighbor strategy. The technique of not using overly close regions as a second best match was used by Forssen and Lowe [30].

The match correctness is determined by the *overlap error* [2]. It measures how well regions A and B correspond under a known homography and is defined by the ratio of the intersection and union of the regions: $s = 1 - \frac{A \cap H^T B H}{A \cup H^T B H}$.

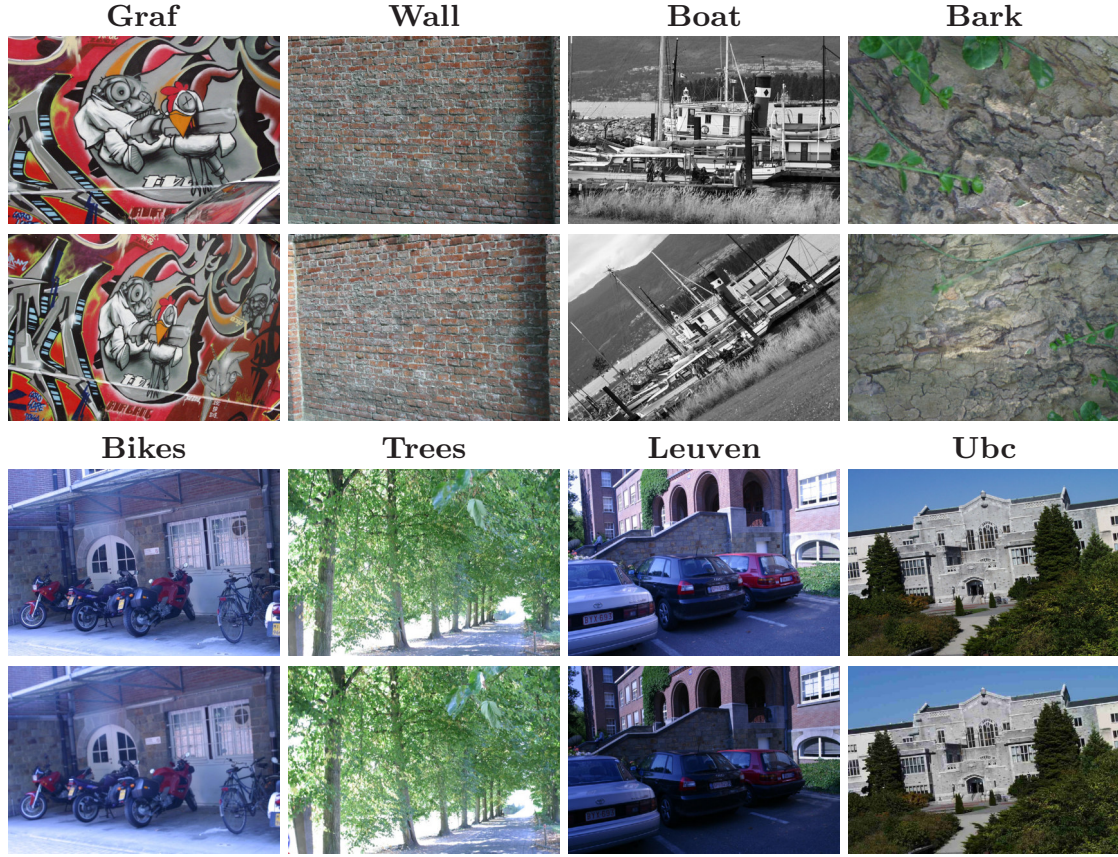


Fig. 3. Examples of test images (left): **Graf** (viewpoint change, structured scene), **Wall** (viewpoint change, textured scene), **Boat** (scale change + image rotation, structured scene), **Bark** (scale change + image rotation, textured scene), **Bikes** (image blur, structured scene), **Trees** (image blur, textured scene), **Leuven** (light change, structured scene), and **Ubc** (JPEG compression, structured scene)

s in [2] a match is assumed to be correct if $s < 0.5$. The correspondence number (possible correct matches) is the maximum matching size in the correct match bipartite graph. The results are presented with *recall* versus *precision* $recall = \frac{\#correct\ matches}{\#correspondences}$ $1 - precision = \frac{\#false\ matches}{\#all\ matches}$.

In all experiments we used eddies SIFT detector and descriptor implementation [31]. All parameters were set to the defaults except the oriented gradient bins number where we also tested our method with a SIFT descriptor having 16 oriented gradient bins (see Fig. 1 (b) in page 498).

6 Results

In this section we present and discuss the experimental results. The performance of FT_{DIST} is compared to that of L_2 [20] L_1 [20] division distance [21] and M_{MOD}^1 [22 23] for viewpoint change scale and rotation image blur light

¹ Note that as the fast algorithm for EMD_{MOD} assumes normalized histograms, we normalized each histogram for its computation.

change and compression. The matching was done between FT descriptors with eight and sixteen orientation bins (see Fig. 1 in page 498). Finally we present run time results.

Figs. 4 5 6 are precision vs. recall graphs. Due to space constraints we present graphs for the matching of the first to the third and fifth images from each folder in the Miola city and chmid dataset [29]. Results for the rest of the data are similar and are in [32].

In all of the experiments FT_{DIST} computed between FT descriptors with sixteen oriented gradient bins (FT_{16}) outperforms all other methods.

FT_{DIST} computed between the original FT descriptors with eight oriented gradient bins (FT_8) is usually the second. Also using FT_{DIST} consistently produces results with greater precision and recall for the symmetric nearest neighbor matching (the rightmost point of each curve).

Increasing the number of orientation bins from eight to sixteen decreases the performance of L_2 and increases the performance of FT_{DIST} . This can be explained by the FT_{DIST} robustness to quantization errors.

Fig. 4 shows matching results for viewpoint change on structured (**Graf**) and textured (**Wall**) scenes. FT_{DIST} has the highest ranking. SIFT performance is better on the textured scene. Performance decreases with viewpoint change (compare **Graf-3** to **Graf-5** and **Wall-3** to **Wall-5**) while the distance ranking remains the same. For large viewpoint change performance is poor (see **Graf-5**) and the resulting graphs are not smooth. This can be explained by the fact that the FT detector and descriptor are not affine invariant.

Fig. 5 shows matching results for similarity transformation on structured (**Boat**) and textured (**Bark**) scenes. FT_{DIST} outperforms all other distances. SIFT performance is better on the textured scene.

Fig. 6 shows matching results for image blur on structured (**Bikes**) and textured (**Trees**) scenes. FT_{DIST} has the highest ranking. The FT descriptor is affected by image blur. A similar observation was made by Miola city and chmid [2].

Fig. 7 shows matching results for light change. FT_{DIST} obtains the best matching score. The performance decreases with lack of light (compare **Leuven-3** to **Leuven-5**) although not drastically.

Fig. 8 shows matching results for compression. FT_{DIST} outperforms all other distances. Performance decreases with compression level (compare **Ubc-3** to **Ubc-5**).

Table 1 presents run time results. All runs were conducted on a dual core Meron 2.6 processor. The table contains the run time in seconds of

Table 1. Run time results in seconds of 10 distance computations

	SIFT _{DIST}	$(L_2)^2$	EMD _{MOD} [22,23]	Diffusion [21]	EMD- L_1 [20]
SIFT-8	1.5	0.35	18	28	192
SIFT-16	2.2	1.3	29	56	637

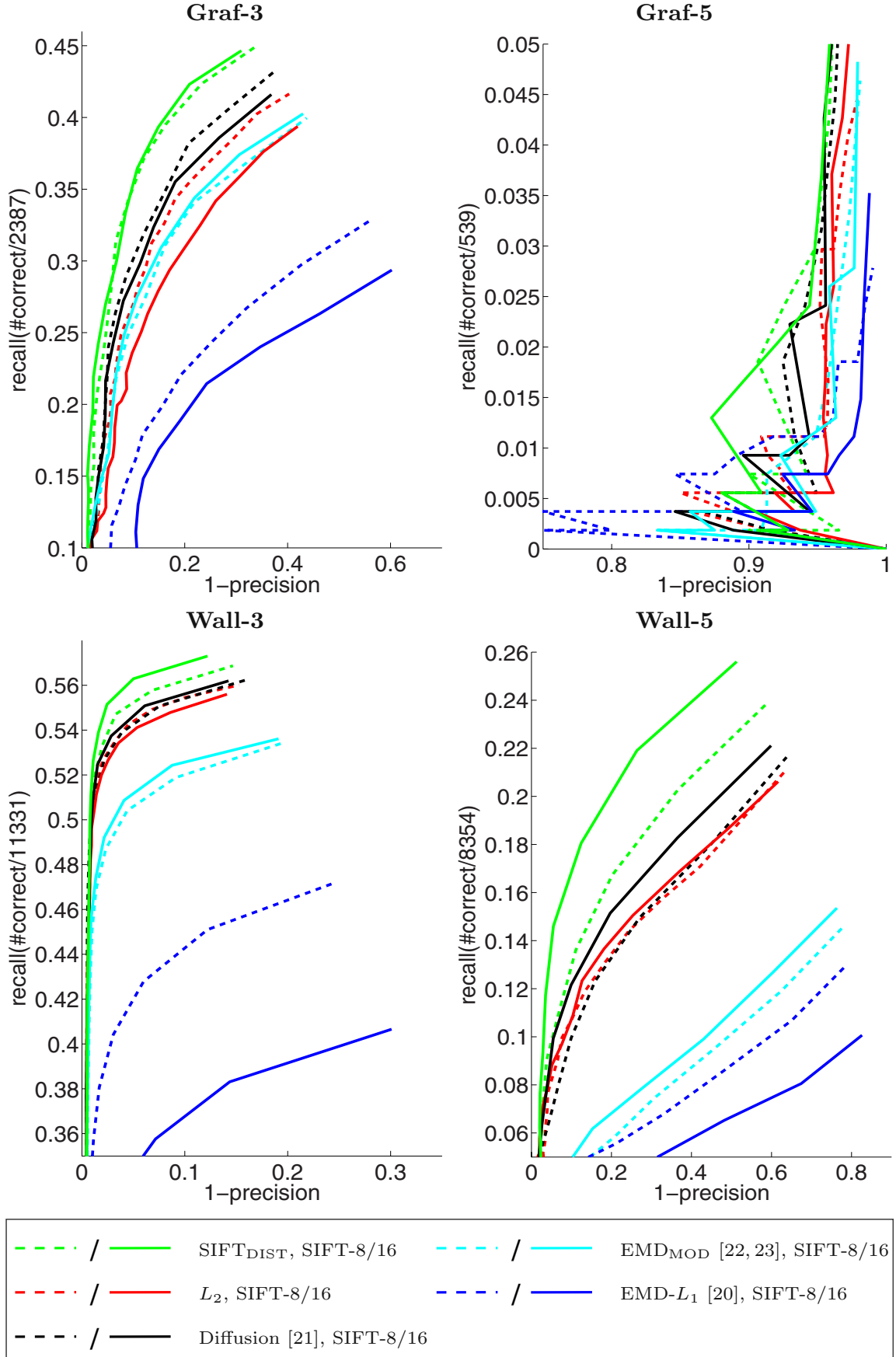


Fig. 4. Results on the Mikolajczyk and Schmid dataset [2]. Should be viewed in color.

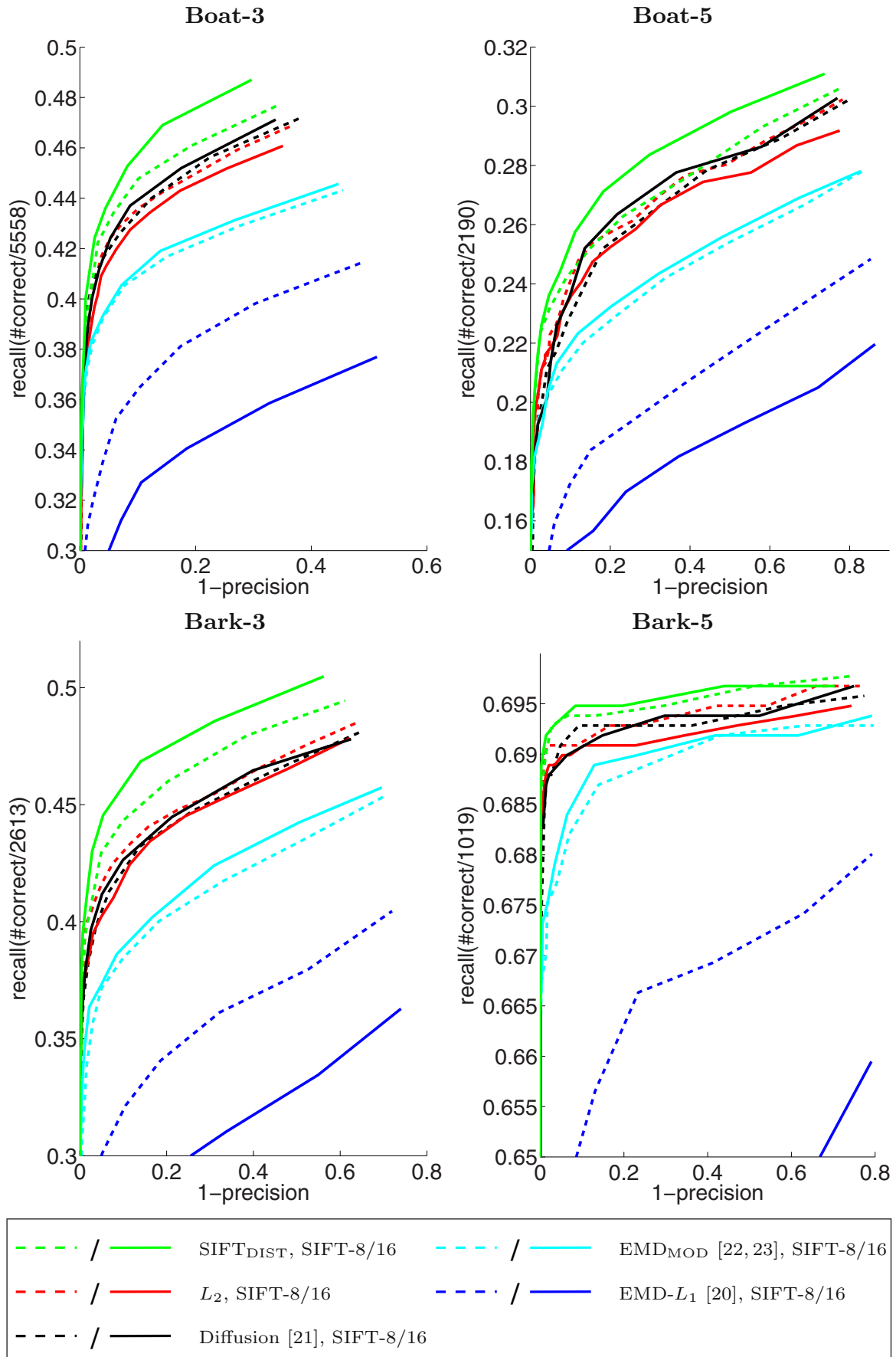


Fig. 5. Results on the Mikolajczyk and Schmid dataset [2]. Should be viewed in color.

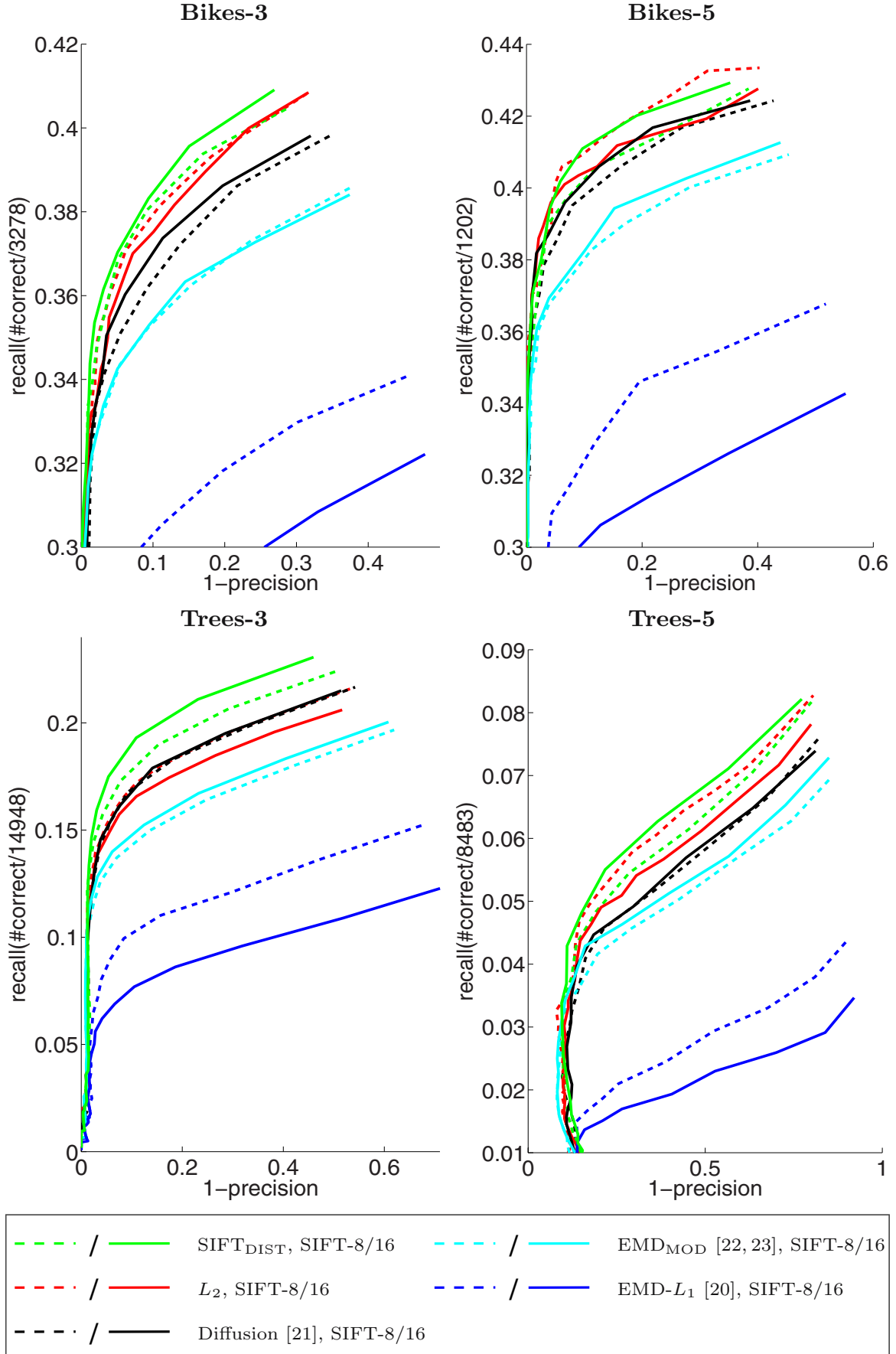


Fig. 6. Results on the Mikolajczyk and Schmid dataset [2]. Should be viewed in color.

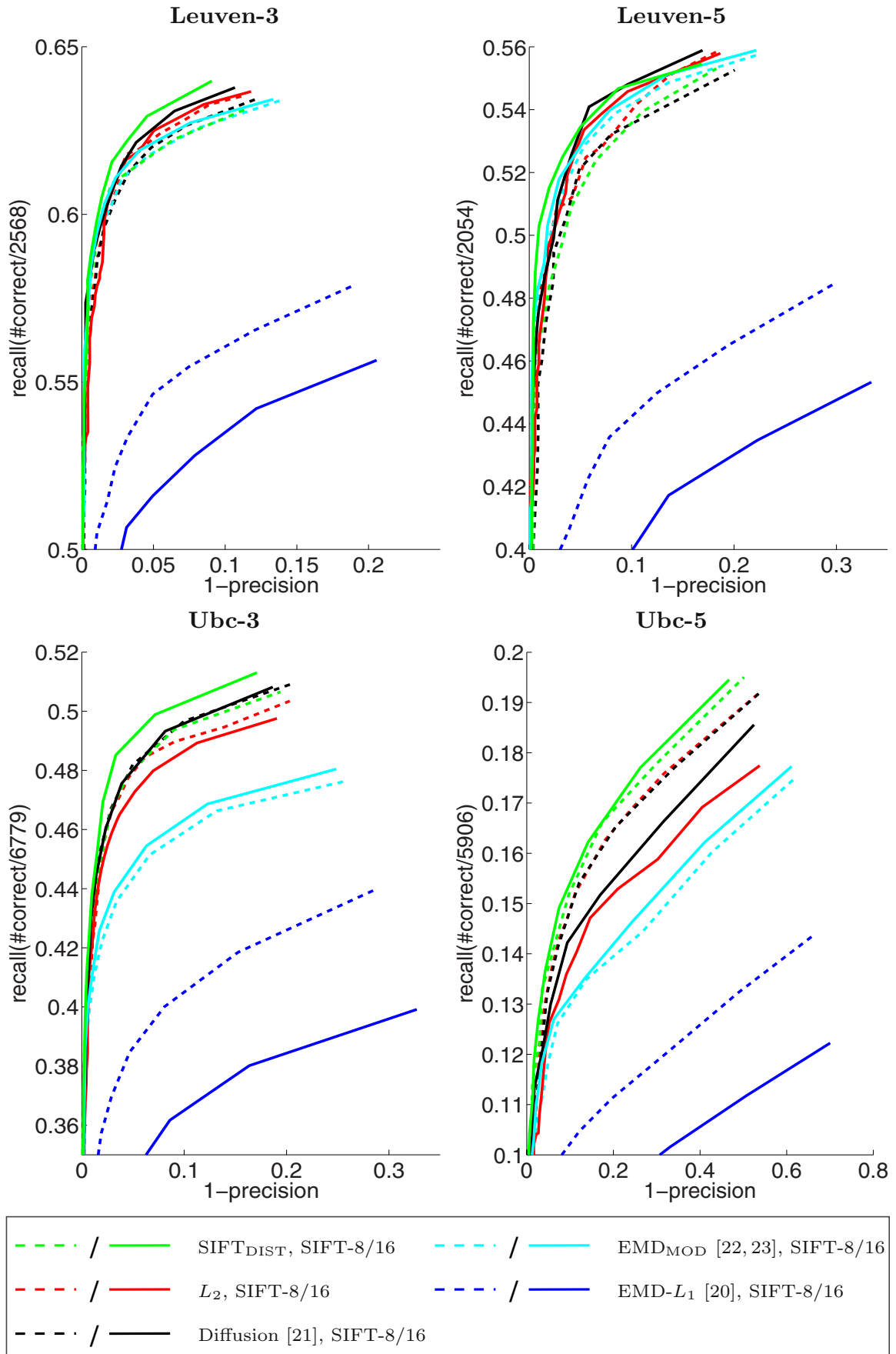


Fig. 7. Results on the Mikolajczyk and Schmid dataset [2]. Should be viewed in color.

each distance computation between two sets of 1000 FT descriptors with eight and sixteen oriented gradient bins. Note that we measured the run time of $(L_2)^2$ and not L_2 as computing the root does not change the order of elements and is time consuming. FT_{DIST} is the fastest cross bin distance.

7 Conclusions

We presented a new cross bin metric between histograms and a linear time algorithm for its computation. Extensive experimental results for FT matching on the Miola city and chmid dataset [2] showed that our method outperforms state of the art distances.

The speed can be further improved using techniques such as Bayesian sequential hypothesis testing [33] sub linear indexing [34] and a approximate nearest neighbor [35–36]. The new cross bin histogram metric may also be useful for other histograms either cyclic (*e.g.* hue in color images) or non cyclic (*e.g.* in intensity in grayscale images). The project home page including code (C++ and Matlab wrappers) is at <http://www.cs.huji.ac.il/~ofirpele/SiftDist>.

References

1. Rubner, Y., Tomasi, C., Guibas, L.J.: The earth mover’s distance as a metric for image retrieval. *International Journal of Computer Vision* 40(2), 99–121 (2000)
2. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. *IEEE Trans. Pattern Analysis and Machine Intelligence* 27(10), 1615–1630 (2005)
3. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision* 60(2), 91–110 (2004)
4. Bay, H., Tuytelaars, T., Gool, L.J.V.: Surf: Speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) *ECCV 2006*. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006)
5. Dalai, N., Triggs, B., Rhone-Alps, I., Montbonnot, F.: Histograms of oriented gradients for human detection. In: *CVPR*, vol. 1 (2005)
6. Heikkila, M., Pietikainen, M., Schmid, C.: Description of Interest Regions with Center-Symmetric Local Binary Patterns. In: *ICVGIP*, pp. 58–69 (2006)
7. Ferrari, V., Tuytelaars, T., Van Gool, L.: Simultaneous object recognition and segmentation by image exploration. In: Pajdla, T., Matas, J(G.) (eds.) *ECCV 2004*. LNCS, vol. 3021, pp. 40–54. Springer, Heidelberg (2004)
8. Sudderth, E., Torralba, A., Freeman, W., Willsky, A.: Learning hierarchical models of scenes, objects, and parts. In: *ICCV*, vol. 2, pp. 1331–1338 (2005)
9. Arth, C., Leistner, C., Bischof, H.: Robust Local Features and their Application in Self-Calibration and Object Recognition on Embedded Systems. In: *CVPR* (2007)
10. Mikolajczyk, K., Leibe, B., Schiele, B.: Multiple object class detection with a generative model. In: *CVPR* (2006)
11. Dorko, G., Schmid, C., Gravir-Cnrs, I., Montbonnot, F.: Selection of scale-invariant parts for object class recognition. In: *ICCV*, pp. 634–639 (2003)
12. Opelt, A., Fussenegger, M., Pinz, A., Auer, P.: Weak hypotheses and boosting for generic object detection and recognition. In: Pajdla, T., Matas, J(G.) (eds.) *ECCV 2004*. LNCS, vol. 3022. Springer, Heidelberg (2004)

13. Sivic, J., Zisserman, A.: Video Google: a text retrieval approach to object matching in videos. In: ICCV, pp. 1470–1477 (2003)
14. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: CVPR (2007)
15. Snavely, N., Seitz, S., Szeliski, R.: Photo tourism: exploring photo collections in 3D. *ACM Transactions on Graphics (TOG)* 25(3), 835–846 (2006)
16. Sivic, J., Everingham, M., Zisserman, A.: Person Spotting: Video Shot Retrieval for Face Sets. In: Leow, W.-K., Lew, M., Chua, T.-S., Ma, W.-Y., Chaisorn, L., Bakker, E.M. (eds.) CIVR 2005. LNCS, vol. 3568, pp. 226–236. Springer, Heidelberg (2005)
17. Se, S., Lowe, D., Little, J.: Local and global localization for mobile robots using visuallandmarks. In: IROS, vol. 1 (2001)
18. Brown, M., Lowe, D.: Recognising panoramas. In: ICCV, p. 3 (2003)
19. Nowak, E., Jurie, F., Triggs, B.: Sampling strategies for bag-of-features image classification. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 490–503. Springer, Heidelberg (2006)
20. Ling, H., Okada, K.: An Efficient Earth Mover’s Distance Algorithm for Robust Histogram Comparison. *IEEE Trans. Pattern Analysis and Machine Intelligence* 29(5), 840–853 (2007)
21. Ling, H., Okada, K.: Diffusion distance for histogram comparison. In: CVPR, vol. 1, pp. 246–253 (2006)
22. Werman, M., Peleg, S., Melter, R., Kong, T.: Bipartite graph matching for points on a line or a circle. *Journal of Algorithms* 7(2), 277–284 (1986)
23. <http://www.cs.huji.ac.il/~ofirpele/publications/ECCV2008.pdf>
24. Shen, H., Wong, A.: Generalized texture representation and metric. *Computer vision, graphics, and image processing* 23(2), 187–206 (1983)
25. Werman, M., Peleg, S., Rosenfeld, A.: A distance metric for multidimensional histograms. *Computer Vision, Graphics, and Image Processing* 32(3) (1985)
26. Peleg, S., Werman, M., Rom, H.: A unified approach to the change of resolution: Space and gray-level. *IEEE Trans. Pattern Analysis and Machine Intelligence* 11(7), 739–742 (1989)
27. Cha, S., Srihari, S.: On measuring the distance between histograms. *Pattern Recognition* 35(6), 1355–1370 (2002)
28. Indyk, P., Thaper, N.: Fast image retrieval via embeddings. In: 3rd International Workshop on Statistical and Computational Theories of Vision (October 2003)
29. <http://www.robots.ox.ac.uk/~vgg/research/affine/index.html>
30. Forssén, P., Lowe, D.: Shape Descriptors for Maximally Stable Extremal Regions. In: ICCV, pp. 1–8 (2007)
31. <http://vision.ucla.edu/~vedaldi/code/sift/sift.html>
32. <http://www.cs.huji.ac.il/~ofirpele/publications/ECCV2008addRes.pdf>
33. Pele, O., Werman, M.: Robust real time pattern matching using bayesian sequential hypothesis testing. *IEEE Trans. Pattern Analysis and Machine Intelligence* 30(8), 1427–1443 (2008)
34. Obdrzalek, S., Matas, J.: Sub-linear indexing for large scale object recognition. In: BMVC, vol. 1, pp. 1–10 (2005)
35. Arya, S., Mount, D., Netanyahu, N., Silverman, R., Wu, A.: An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM (JACM)* 45(6), 891–923 (1998)
36. Beis, J., Lowe, D.: Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In: CVPR, pp. 1000–1006 (1997)

Chapter 3

Fast and Robust Earth Mover's Distances

Fast and Robust Earth Mover's Distances

Ofir Pele

The Hebrew University of Jerusalem

ofirpele@cs.huji.ac.il

Michael Werman

The Hebrew University of Jerusalem

werman@cs.huji.ac.il

Abstract

We present a new algorithm for a robust family of Earth Mover's Distances - EMDs with thresholded ground distances. The algorithm transforms the flow-network of the EMD so that the number of edges is reduced by an order of magnitude. As a result, we compute the EMD by an order of magnitude faster than the original algorithm, which makes it possible to compute the EMD on large histograms and databases. In addition, we show that EMDs with thresholded ground distances have many desirable properties. First, they correspond to the way humans perceive distances. Second, they are robust to outlier noise and quantization effects. Third, they are metrics. Finally, experimental results on image retrieval show that thresholding the ground distance of the EMD improves both accuracy and speed.

1. Introduction

Histograms are ubiquitous tools in numerous computer vision tasks. It is common practice to use distances such as L_2 or χ^2 for comparing histograms. This practice assumes that the histogram domains are aligned. However this assumption is violated through quantization, shape deformation, light changes, etc.

The Earth Mover's Distance (EMD) [29] is a cross-bin distance that addresses this alignment problem. EMD is defined as the minimal cost that must be paid to transform one histogram¹ into the other, where there is a "ground distance" between the basic features that are aggregated into the histogram. The EMD as defined by Rubner is a metric only for normalized histograms. However, recently Pele and Werman [26] suggested $\widehat{EMD}$ and showed that it is a metric for all histograms.

A major issue that arises when using EMD is which ground distance to use for the basic features. This, of course, depends on the histograms, the task and practical

¹Rubner's noted that EMD can be used with sparse histograms which he coined *signatures*. Our algorithm is applicable to both histograms and signatures.

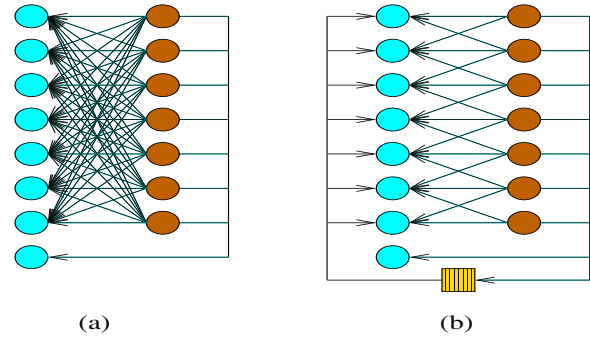


Figure 1. An example of the transformation on a flow network of an EMD or $\widehat{EMD}$ with a ground distance of: $d(a, b) = \min(2, |a - b|)$. (a) is the original flow network with $N^2 + N$ edges. Note that $N(N - 3)$ of these edges have cost 2. The bottom cyan vertex on the left is the sink that handles the difference between the total mass of the two histograms (ingoing edges cost is 0 for EMD and $\alpha \max_{ij} d_{ij}$ for $\widehat{EMD}$). (b) is the transformed flow network. The striped yellow square is the new transshipment vertex. Ingoing edge cost is the threshold (e.g. 2) and outgoing edge cost is 0.

considerations. In many cases we would like the distance to correspond to the way humans perceive distances (image retrieval, for example). In other cases we would like the distance to fit the distribution of the noise (keypoint matching, for example). Practical considerations include speed of computation and the metric property that enables fast algorithms for nearest neighbor searches [41, 7], fast clustering [10] and large margin classifiers [15, 36].

We propose using thresholded ground distances. *i.e.* distances that saturate to a constant value. These distances have many desirable properties. First, saturated distances correspond to the way humans perceive distances [34]. Second, many natural noise distributions have a heavy tail; *i.e.* outlier noise. Thresholded distances assign different outliers the same large distance. Finally, we present an algorithm that computes EMD with a thresholded ground distance faster by an order of magnitude than the original algorithm. The algorithm transforms the flow-network of the EMD so that the number of edges is reduced by an order of

magnitude (see Fig. 1).

The Earth Mover's Distance has been used successfully in many applications such as image retrieval [29, 23], edge and corner detection [30], keypoint matching [26, 8, 20], near duplicate image identification [40], classification of texture and object categories [42, 19], NMF [31] and contour matching [12]. Many of these works used saturated distances, usually the negative exponent function. The major contribution of this paper is a fast algorithm for the computation of the EMD with thresholded ground distance. We argue that thresholded distances have all the benefits of the negative exponent function that is typically used as a saturated distance; its big advantage is its much shorter computation time.

This paper is organized as follows. Section 2 is an overview of previous work. Section 3 describes the Earth Mover's Distance. Section 4 discusses thresholded distances and proves that they are metrics. Section 5 describes the fast algorithm. Section 6 presents the results. Finally, conclusions are drawn in Section 7.

2. Previous Work

This section first describes EMD algorithms. Second, it describes the use of saturated ground distances in the EMD framework.

2.1. EMD Algorithms

Early work using cross-bin distances for histogram comparison can be found in [33, 39, 38, 28]. Shen and Wong [33] suggested unfolding two integer histograms, sorting them and then computing the L_1 distance between the unfolded histograms. To compute the modulo matching distance between cyclic histograms they took the minimum from all cyclic permutations. This distance is equivalent to the EMD between two normalized histograms. Werman *et al.* [39] showed that this distance is equal to the L_1 distance between the cumulative histograms. They also proved that matching two cyclic histograms by only examining cyclic permutations is optimal. Werman *et al.* [38] proposed an $O(M \log M)$ algorithm for finding a minimal matching between two sets of M points on a circle. The algorithm was adapted by Pele and Werman [26] to compute the EMD between two N -bin, normalized histograms with time complexity $O(N)$. Peleg *et al.* [28] suggested using the EMD for grayscale images and using linear programming to compute it. Rubner *et al.* [29] suggested using the EMD for color and texture images and generalized the definition of the EMD to non-normalized histograms. They computed the EMD using a specific linear programming algorithm - the transportation simplex. The algorithm's worst case time complexity is exponential. Practical run time was shown to be super-cubic. Interior-point algorithms or Orlin's algo-

rithm [25] both have a time complexity of $O(N^3 \log N)$ and can also be used.

Ling and Okada proposed EMD- L_1 [20]; *i.e.* EMD with L_1 as the ground distance. They showed that if the points lie on a Manhattan network (*e.g.* an image), the number of variables in the LP problem can be reduced from $O(N^2)$ to $O(N)$. To execute the EMD- L_1 computation, they employed a tree-based algorithm, Tree-EMD. Tree-EMD exploits the fact that a basic feasible solution of the simplex algorithm-based solver forms a spanning tree when the EMD- L_1 is modeled as a network flow optimization problem. The worst case time complexity is exponential. Empirically, they showed that this algorithm has an average time complexity of $O(N^2)$. Gudmundsson *et al.* [14] also put forward this simplification of the LP problem. They suggested an $O(N \log^{d-1} N)$ algorithm that creates a Manhattan network for a set of N points in $\mathbb{R}^d$. The Manhattan network has $O(N \log^{d-1} N)$ vertices and edges. Thus, using Orlin's algorithm [25] the EMD- L_1 can be computed with a time complexity of $O(N^2 \log^{2d-1} N)$. Indyk and Thaper [17] proposed approximating EMD- L_1 by embedding it into the L_1 norm. Embedding time complexity is $O(Nd \log \Delta)$, where N is the feature set size, d is the feature space dimension and Δ is the diameter of the union of the two feature sets. Grauman and Darrell [13] substituted L_1 with histogram intersection in order to approximate partial matching. Shirdhonkar and Jacobs [35] presented a linear-time algorithm for approximating EMD- L_1 for low dimensional histograms using the sum of absolute values of the weighted wavelet coefficients of the difference histogram. Lv *et al.* [23] proposed embedding an EMD with thresholded ground distance into the L_1 norm. Khot and Naor [18] showed that any embedding of the EMD over the d -dimensional Hamming cube into L_1 must incur a distortion of $\Omega(d)$, thus losing practically all distance information. Andoni *et al.* [5] showed that for sets with cardinalities upper bounded by a parameter s , the distortion reduces to $O(\log s \log d)$. A practical reduction in accuracy due to the approximation was reported by [12, 23, 35]. In order to increase precision, Grauman and Darrell [12] and Lv *et al.* [23] used the approximation as a filter that returns a set of similar objects, and then used the exact EMD computation to rerank these objects. Khanh Do Ba *et al.* [6] presented optimal algorithms for estimating EMD- L_1 or EMD with a tree-metric as the ground distance.

Pele and Werman [26] proposed $\widehat{EMD}$ - a new definition of the EMD for non-normalized histograms. They showed that unlike Rubner's definition, the $\widehat{EMD}$ is also a metric for non-normalized histograms. In addition, they proposed a linear-time algorithm that computes the $\widehat{EMD}$ with a ground distance of 0 for corresponding bins, 1 for adjacent bins and 2 for farther bins and for the extra mass.

2.2. Saturated Ground Distances with the EMD

Rubner *et al.* [29] and Ruzon and Tomasi [30] used a negative exponent function to saturate their ground distance for the tasks of image retrieval and edge detection, respectively. The negative exponent function practically saturates large distances to a fixed threshold (see Fig. 2). The negative exponent function is used for saturating a metric, because it does not break the triangle inequality. We show that this is true as well for a thresholding function. Note that saturating with a negative exponent might have the drawback of changing the behavior of small distances (see Fig. 2).

Lv *et al.* [23] conducted image retrieval experiments from a database of 10000 images. They showed that thresholding the ground distance improves precision.

Pele and Werman [26] compared several distances for the task of SIFT matching. They proposed an $\widehat{EMD}$ variant. The ground distance of this $\widehat{EMD}$ is 0 for corresponding bins, 1 for adjacent bins and 2 for farther bins and for the extra mass; *i.e.* a thresholded distance. They showed that this distance improves SIFT matching, while $\widehat{EMD}$ with non-thresholded distances negatively affects performance.

3. The Earth Mover's Distance

The Earth Mover's Distance (EMD) [29] is defined as the minimal cost that must be paid to transform one histogram into the other, where there is a "ground distance" between the basic features that are aggregated into the histogram.

Given two histograms P, Q the EMD as defined by Rubner *et al.* [29] is:

$$EMD(P, Q) = (\min_{\{f_{ij}\}} \sum_{i,j} f_{ij} d_{ij}) / (\sum_{i,j} f_{ij}) \quad s.t. \quad f_{ij} \geq 0$$

$$\sum_j f_{ij} \leq P_i \quad \sum_i f_{ij} \leq Q_j \quad \sum_{i,j} f_{ij} = \min(\sum_i P_i, \sum_j Q_j)$$

where $\{f_{ij}\}$ denotes the flows. Each f_{ij} represents the amount transported from the i th supply to the j th demand. We call d_{ij} the *ground distance* between bin i and bin j in the histograms. Pele and Werman [26] suggested $\widehat{EMD}$:

$$\widehat{EMD}_\alpha(P, Q) = (\min_{\{f_{ij}\}} \sum_{i,j} f_{ij} d_{ij}) + |\sum_i P_i - \sum_j Q_j| \alpha \max_{i,j} d_{ij}$$

s.t. EMD constraints

Pele and Werman proved that $\widehat{EMD}$ is a metric for any two histograms if the ground distance is a metric and $\alpha \geq 0.5$ [26]. The metric property enables fast algorithms for nearest neighbor searches [41, 7]), fast clustering [10] and large margin classifiers [15, 36]. Because of these advantages we will use the Pele and Werman definition in the remainder of this paper (with $\alpha = 1$).

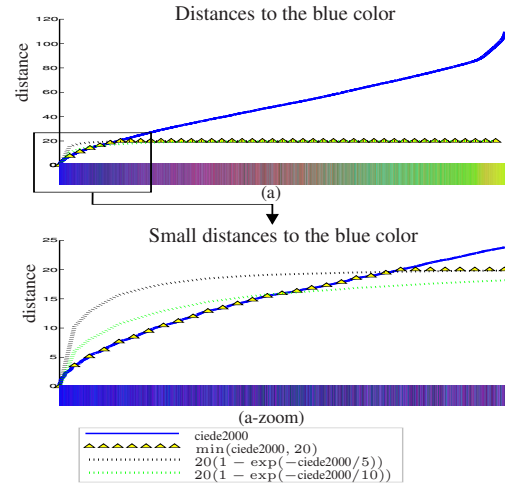


Figure 2. This figure should be viewed in color, preferably on a computer screen. The x-axes are colors, sorted by their distance to the blue color. The distances are the ciede2000 distance [22] and three monotonic saturating transformations applied to it: two negative exponent functions and a thresholding function; *i.e.* a minimum function. There are several observations we can derive from these graphs. First, although ciede2000 is considered as state of the art, it is still far from perfect, especially in the medium to large distance range. Second, color distances should be saturated. For example, although red and yellow are both simply different from blue, the ciede2000 distance between blue and red is 56, while the ciede2000 distance between blue and yellow is 102. This was already noted by Rubner *et al.* [29] and Ruzon and Tomasi [30] who suggested using a negative exponent function because it is a metric, if the distance that is raised to the power is a metric. In this paper we show that the thresholding function is also a metric if the thresholded distance is a metric. Finally, for most of the range, the negative exponent and thresholded functions are very similar. They mostly differ for small distances, where the negative exponent changes the original distance, while the thresholding function does not. Since the ciede2000 was designed and perceptually tested on this range [22], changing the distances on this range can negatively affect performance as was noted by Rubner *et al.* for the Euclidean distance on $L^*a^*b^*$ space [29].

4. Thresholded Distances

Thresholded distances are distances that saturate to a threshold; *i.e.* let $d(a, b)$ be a distance measure between two features - a, b . The thresholded distance with a threshold of $t > 0$ is defined as: $d_t(a, b) = \min(d(a, b), t)$.

We now prove that if d is a metric then d_t is also a metric. Non-negativity and symmetry hold trivially, so we only need to prove that the triangle inequality holds.

$$d_t(a, b) + d_t(b, c) \geq d_t(a, c) \text{ if } d \text{ is a metric.}$$

We consider three cases:

1. $(d_t(a, b) < t) \wedge (d_t(b, c) < t) \wedge (d_t(a, c) < t) \Rightarrow$
 $(d_t(a, b) = d(a, b)) \wedge (d_t(b, c) = d(b, c)) \wedge$
 $(d_t(a, c) = d(a, c)) \Rightarrow d_t(a, b) + d_t(b, c) \geq d_t(a, c)$
2. $(d_t(a, b) = t) \vee (d_t(b, c) = t) \Rightarrow$
 $d_t(a, b) + d_t(b, c) \geq t \geq d_t(a, c)$
3. $d_t(a, c) = t \Rightarrow$

Assume for contradiction that:

$$\begin{aligned}
 & d_t(a, b) + d_t(b, c) < d_t(a, c) = t \Rightarrow \\
 & (d_t(a, b) < t) \wedge (d_t(b, c) < t) \Rightarrow \\
 & (d_t(a, b) = d(a, b)) \wedge (d_t(b, c) = d(b, c)) \Rightarrow \\
 & d(a, b) + d(b, c) < t = d_t(a, c) \leq d(a, c) \Rightarrow \\
 & d(a, b) + d(b, c) < d(a, c)
 \end{aligned}$$

The last statement contradicts the triangle inequality of the metric d . The union of cases 2 and 3 is complement of case 1. Thus, cases 1-3 constitute the entire event space. Therefore we proved that d_t is a metric if d is a metric. It is noteworthy that d_t can be a metric even if d is not a metric.

5. Fast Computation of the EMD with a Thresholded Ground Distance

This section describes an algorithm that computes $\widehat{EMD}$ or EMD with a thresholded ground distance an order of magnitude faster than the original algorithm².

$\widehat{EMD}$ can be solved by a min-cost-flow algorithm. Our algorithm makes a simple transformation of the flow network that reduces the number of edges. If N is the number of bins in the histogram, the flow network of $\widehat{EMD}$ has exactly $N^2 + N$ edges (see (a) in Fig. 1). N^2 edges connect all sources to all sinks. The extra N edges connect all sources to the sink that handles the difference between the total mass of the two histograms (we assume without loss of generality that the source histogram total mass is greater or equal to the sink histogram total mass).

The transformation (see Fig. 1) first removes all edges with cost t . Second, it adds a new transshipment vertex. Finally we connect all sources to this vertex with edges of cost t and connect the vertex to all sinks with edges of cost 0.

Let K be the average number of edges going out of each bin that have a cost different than the threshold t . The new flow network has $NK + N$ edges from the original network, N edges connecting all sources to the transshipment vertex and N edges connecting the transshipment vertex to all sinks. Thus the total number of edges is $N(K + 3)$. If

²Note that the optimization problems in EMD and $\widehat{EMD}$ are exactly the same. Thus, any algorithm that computes EMD can compute $\widehat{EMD}$ and vice versa with the same time complexity.

K is a constant the number of edges is $O(N)$ as opposed to the original $\Theta(N^2)$. Note that the new flow network is no longer a transportation problem, but a transshipment problem [2]. However, both are special cases of the min-cost-flow problem. Thus any algorithm that solves min-cost-flow can be used for both problems.

Let $K = O(1)$; the min-cost-flow optimization problem can be solved with a worst case time complexity of:

$$\begin{aligned}
 & O(\min((N^2 \log \log U \log(NC)), (N^2 \log U \sqrt{\log C})), \\
 & (N^2 \log U \log N), (N^2 \log^2 N)))
 \end{aligned}$$

The algorithms are taken from: Ahuja *et al.* [1], Edmonds and Karp [9], used with Ahuja *et al.*'s shortest path algorithm [3], Edmonds and Karp [9], used with Fredman and Tarjan's shortest path algorithm [11] and Orlin [25]. Algorithms with a C term assume integral cost coefficients that are bounded by C . Algorithms with a U term assume integral supply and demands that are bounded by U .

We now prove that the original and the transformed flow networks have the same minimum-cost solution. Let $\mathcal{O}$ be the original flow network and let $\mathcal{T}$ be the transformed flow network. We first show how to create a feasible flow in $\mathcal{T}$, given a feasible flow in $\mathcal{O}$. Both flows will have the same cost. This will prove that the min-cost-flow solution for $\mathcal{T}$ is smaller or equal to the min-cost-flow solution for $\mathcal{O}$.

Given a feasible flow in $\mathcal{O}$, all flows on edges that were not removed are copied to the new flow for $\mathcal{T}$. For flows on edges with the cost of the threshold, we transfer the flow through the transshipment vertex. This gives us a feasible flow in $\mathcal{T}$ with the same cost.

We now show how to create a feasible flow in $\mathcal{O}$, given a feasible flow in $\mathcal{T}$. The flow in $\mathcal{O}$ will have a cost smaller or equal to the cost of the flow in $\mathcal{T}$. This will prove that the min-cost-flow solution for $\mathcal{T}$ is greater or equal to the min-cost-flow solution for $\mathcal{O}$. Together with the previous proof, this shows that the two flow networks have the same min-cost solution.

Given a feasible flow in $\mathcal{T}$, all flows on edges not connected to the transshipment vertex are copied to $\mathcal{O}$. Second, each unit of mass that flows from vertex i to the transshipment vertex and then from the transshipment vertex to vertex j is transferred directly from vertex i to vertex j . We note that this is possible since $\mathcal{O}$ is fully bi-partite. We also note that the cost of the new flow will be smaller or equal to the cost of the flow in $\mathcal{T}$ as all edges in $\mathcal{O}$ have a cost smaller or equal to the threshold. This completes the proof.

It is noteworthy that the algorithmic technique presented in this paper can be applied not only with thresholded ground distance, but in any case where a group of vertexes can be connected to another group with the same cost. For example, we can add a transshipment vertex for all vertexes with a specific color (*e.g.* blue) such that the cost of the transportation between them will be smaller than the cost of the transportation to other colors.

5.1. Implementation notes

Flow-network set-up time. For a fixed histogram configuration (e.g. SIFT) the flow-network can be pre-computed once. For sparse histograms (*signatures*), the flow-network set-up time complexity is $O(MN)$; where N is the number of non-zero bins and M is the average number of neighbors that need to be checked if the distance is lower than the threshold. M is at most N but it can be lower. For example, let t be the threshold. If we are comparing two images and the ground distance is a linear combination of the spatial distance and the color distance, then the distance computation to vertexes with L_1 distance bigger or equal to t can be skipped. That is, the set-up time in this case is $O(\min(t^2 N, N^2))$.

Pre-flowing Monge sequences. A Monge sequence contains edges in the flow-network that can be pre-flowed (in the order of the sequence) without changing the min-cost solution [24, 16]. For example, if the ground-distance is a metric, zero-cost edges are Monge sequence [38]. Alon *et al.* [4] introduced an efficient algorithm which determines the longest Monge sequence.

Pre-flowing to/from isolated nodes. If a source is connected only to the new transshipment vertex, we can pre-flow all its mass to the transshipment vertex and eliminate it. If a sink is connected only to the transshipment vertex, we can add its deficit to the transshipment vertex and eliminate it.

6. Results

In this section we present results for image retrieval. However, note that we do not claim that this is the optimal way to achieve image retrieval. Image retrieval is used here as an example of an application where the EMD has already been used, to show that thresholded distances yield good results. The major contribution is the faster algorithm. We show that by using our algorithm the running time decreases by an order of magnitude.

We use a database that contains 773 landscape images from the COREL database, that were also used in Wang *et al.* [37]. The dataset contains 10 classes³: People in Africa, Beaches, Outdoor Buildings, Buses, Dinosaurs, Elephants, Flowers, Horses, Mountains and Food. The number of images in each class ranges from 50 to 100.

From each class we selected 5 images as query images (numbers 1, 10, ..., 40). Then we searched for the 50 nearest neighbors for each query image. We computed the distance of each image to the query image and its reflection and took the minimum. We present results for three types of image representations: histograms of orientations, $L^*a^*b^*$

color space and finally linear combinations of the two. We conclude this section with running times.

6.1. SIFT

Our first image representation is orientation histograms. The first representation - SIFT is a $6 \times 8 \times 8$ SIFT descriptor [21] computed globally on the whole image. The second representation - CSIFT is a SIFT-like descriptor. This descriptor tackles two problems related to the SIFT descriptor for color image retrieval. First, it takes into account color edges by computing the SIFT descriptor on the compass edge image [30]. Note that on an edge image there should be no distinction between opposite directions (0 and 180 for example). Thus, opposite directions are considered equal. The second drawback of the SIFT descriptor for color image retrieval is its normalization. The normalization is problematic as we lose the distinctive cue of the amount of edge points in the image. In the CSIFT computation we skip the normalization step. The final CSIFT descriptor has $6 \times 8 \times 8$ bins. We used the following distances for these descriptors: L_1 , L_2 , χ^2 , EMD- L_1 [20], $\text{SIFT}_{\text{DIST}}$ [26] and $\widehat{\text{EMD}}$. Let M be the number of orientation bins. The ground distances between bins (x_i, y_i, o_i) and (x_j, y_j, o_j) we use are:

$$d_R = ||(x_i, y_i) - (x_j, y_j)||_2 + \min(|o_i - o_j|, M - |o_i - o_j|)$$

$$d_T = \min(d_R, T)$$

The results are given in Fig. 4(a). Due to lack of space we present for each distance measure, the descriptor with which it performed best. Results of all pairs of descriptors and distance measures can be found at:

www.cs.huji.ac.il/~7eofirpele/FastEMD/. The $\widehat{\text{EMD}}$ with a thresholded ground distance performs much better than $\widehat{\text{EMD}}$ with a non-thresholded ground distance. In fact, while $\widehat{\text{EMD}}$ with a non-thresholded ground distance negatively affects performance, $\widehat{\text{EMD}}$ with a thresholded ground distance improves performance. L_1 is equivalent to $\widehat{\text{EMD}}$ with the Kroncker δ ground distance [26]. $\text{SIFT}_{\text{DIST}}$ is the sum of $\widehat{\text{EMD}}$ over all the spatial cells (each spatial cell contains one orientation histogram). The ground distance for the orientation histograms is: $\min(|o_i - o_j|, M - |o_i - o_j|, 2)$. It was shown in [26] and here that this addition of a small invariance to the orientations shifts improves performance. Our distance also adds a small invariance to spatial shifts and thus improves performance even more. However, using a non-thresholded distance adds too much invariance at the expense of distinctiveness, thus reducing performance.

6.2. $L^*a^*b^*$ Color Space

The state of the art color distance is Δ_{00} - ciede2000 on $L^*a^*b^*$ color space [22, 32] (see also Fig. 2). Thus, our second type of image representation is simply a resized color image in the $L^*a^*b^*$ space. We resized each image to 32×48 and converted them to $L^*a^*b^*$ space. Let I_1, I_2 be

³The original database contains some visually ambiguous classes such as Africa that also contains images of beaches in Africa. We manually removed these ambiguous images.



Figure 3. Example of image retrieval using the best distance - $\widehat{EMD}$ with thresholded ground distances (top row) and the second best distance - L_1 like distance (bottom row). The nearest neighbor images are ordered from left to right by their distance from the query image. Note that by allowing small deformations in the $\widehat{EMD}$ we obtain results that are visually similar to the query image. Allowing larger deformations; *i.e.* using a non-thresholded distance negatively affects results.

the two L^*a*b^* images. We used the following distances:

$$\begin{aligned}
 L_1 \Delta_{00} &= \sum_{x,y} (\Delta_{00}(I_1(x,y), I_2(x,y))) \\
 L_1 \Delta_{00}^T &= \sum_{x,y} (\min(\Delta_{00}(I_1(x,y), I_2(x,y)), T)) \\
 L_2 \Delta_{00} &= \sum_{x,y} (\Delta_{00}(I_1(x,y), I_2(x,y)))^2 \\
 L_2 \Delta_{00}^T &= \sum_{x,y} (\min(\Delta_{00}(I_1(x,y), I_2(x,y)), T))^2
 \end{aligned}$$

We also used $\widehat{EMD}$, where the ground distance between two pixels $(x_i, y_i, L_i, a_i, b_i)$, $(x_j, y_j, L_j, a_j, b_j)$ is:

$$dc_T = \min(\|(x_i, y_i) - (x_j, y_j)\|_2 + \Delta_{00}((L_i, a_i, b_i), (L_j, a_j, b_j)), T)$$

The results for $\widehat{EMD}$ with a non-thresholded ground distance are not reported here since the experiments have not finished running (more than ten days). Results are presented in Fig. 4(b). As shown, $\widehat{EMD}$ with a thresholded ground distance outperforms all other distances.

6.3. Color and SIFT Combined

In the experiments described in this section, we used linear combinations of the orientation histograms and color. We combined the two best distances for each of the methods (note that each distance was normalized so that its average is 1):

$$dC = \alpha (\widehat{EMD} \ d_{T=2}, CSIFT) + (1 - \alpha) (\widehat{EMD} \ dc_{T=20})$$

We also used a combination of the two L_1 -like distances:

$$dC2 = \alpha (L_1, SIFT) + (1 - \alpha) (L_1 \Delta_{00}^{T=20})$$

We used three different α values: 0.1, 0.3, 0.5. The results appear in Fig. 4(c)-(f). The combination of two $\widehat{EMD}$ with thresholded distances performs best, especially for hard classes such as People in Africa and Food. The image results examples for one “Food” query image are given in Fig. 3. More image results can be found at:

www.cs.huji.ac.il/%7eofirpele/FastEMD/

6.4. Running Time Results

The algorithm we used for the computation of the $\widehat{EMD}$ is successive shortest path [2]. This algorithm has a worst time complexity of $O(N^2 U \log N)$. All runs were conducted on a Pentium 2.8GHz. A comparison of the practical running time of our algorithm and other methods are given in tables 1,2 and in Fig. 5. It is noteworthy that EMD- L_1 accuracy is much lower than our method and even lower than the simple L_1 norm (see Fig. 4(a)). Indyk and Thaper [17] and Shirdhonkar and Jacobs [35] approximate EMD- L_1 , so their accuracy is even lower. SIFT_{DIST} gives good accuracy (second best, see Fig. 4(a)) and is faster than our method. Our method has better accuracy. More importantly, SIFT_{DIST} is limited to a thresholded L_1 norm with a threshold of 2 for 1-dimensional histograms. Our method is much more general. For example, SIFT_{DIST} cannot be applied to colors. Lv *et al.* [23] report that their approximation runs 5 times faster than Rubner’s. Our method runs 75-700 times faster and returns the exact distance.

Our	Rubner’s [29]	EMD- L_1 [20]	SIFT _{DIST} [26]
0.04s	3s	0.04s	0.00007s

Table 1. 384-dimensional SIFT-like descriptors matching time

Our	Rubner’s [29]	EMD- L_1 [20]	SIFT _{DIST} [26]
6s	4400s	N.A.	N.A.

Table 2. L^*a*b^* images matching time. Note that EMD- L_1 can be applied only to regular grids and SIFT_{DIST} can be applied only to 1-dimensional histograms.

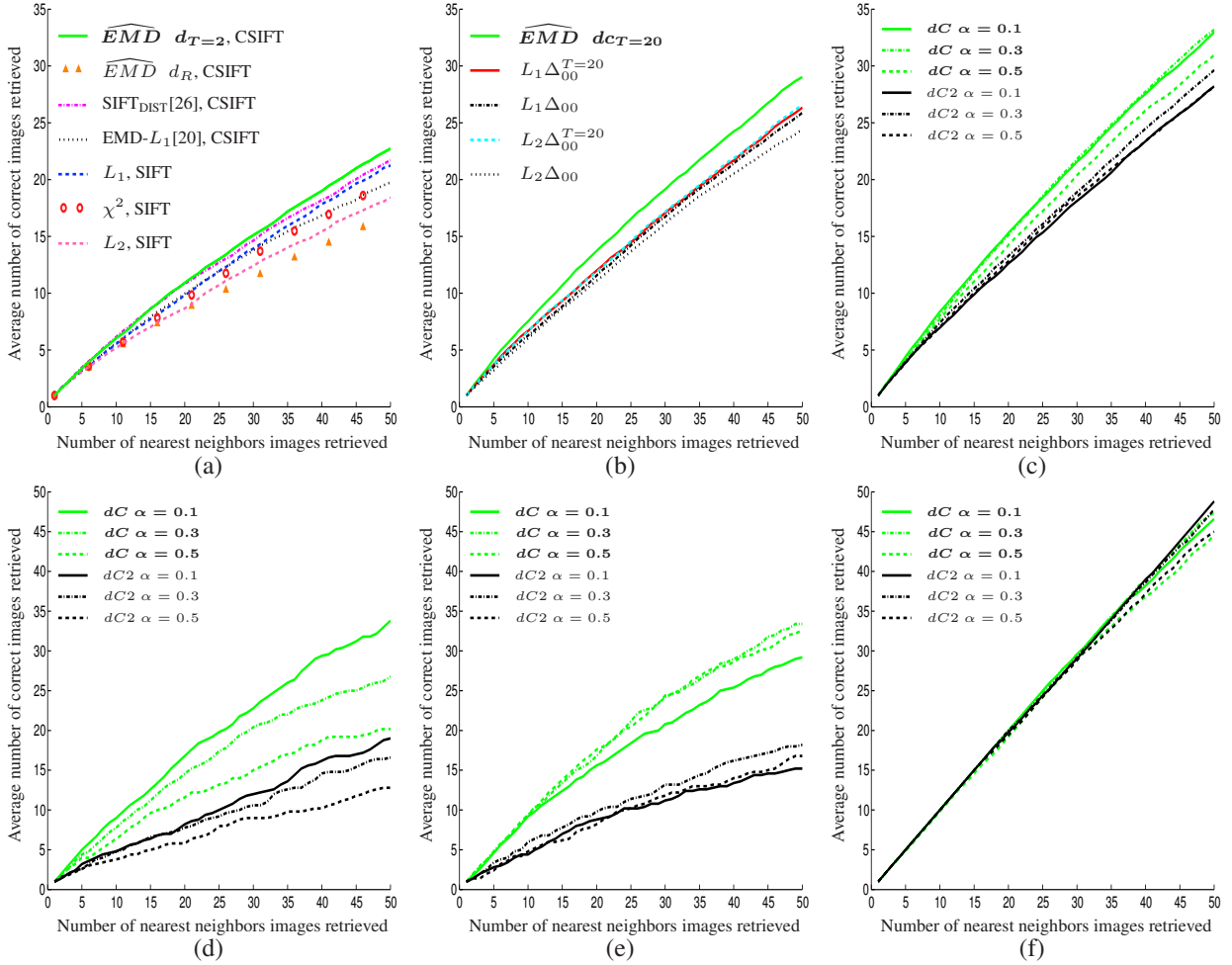


Figure 4. Results for image retrieval. Our method is in bold font. (a) Orientation histogram results. Due to lack of space we present for each distance measure, the descriptor with which it performed best. Results of all pairs of descriptors and distance measures can be found at:

www.cs.huji.ac.il/~7e0firpele/FastEMD/. (b) $L^*a^*b^*$ color space results. (c)-(f) Linear combinations of color and orientation results, where (c) is an average over all classes and (d)-(f) are for specific classes: (d) People in Africa (e) Food (f) Flowers. There are two key observations. First, $\widehat{EMD}$ with thresholded ground distances performs best. Second, some of the classes are easy, *e.g.* (f) Flowers. For these classes $\widehat{EMD}$ does not improve performance. However, for harder classes, such as (d) People in Africa and (e) Food, $\widehat{EMD}$ significantly improves results. Image result examples for one “Food” query image are given in Fig. 3.

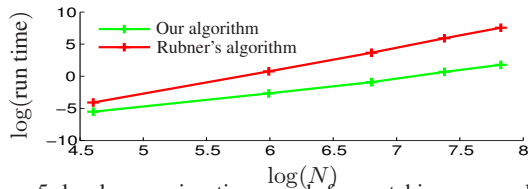


Figure 5. log-log running time graph for matching grayscale images with N pixels.. Linear fit for our algorithm: $2.3 \times \log(N) - 16$. Linear fit for Rubner’s algorithm: $3.6 \times \log(N) - 21$.

7. Conclusions

We presented a new family of Earth Mover’s Distances. Members of this family correspond to the way humans per-

ceive distances, and are robust to outlier noise and quantization effects. We proved that they are metrics. We also proposed a fast algorithm. The algorithm runs an order of magnitude faster than the original algorithm, which makes it possible to compute the EMD on large histograms and databases. Experimental results show that EMD has the best performance when it is used with thresholded distances. This has also been shown by Rubner *et al.* [29] and Ruzon and Tomasi [30] for saturated distances, which are essentially thresholded. This has also been demonstrated for thresholded distances by Lv *et al.* [23] and Pele and Werman [26]. Our results strengthen these findings. Most im-

portantly, our paper shows that using a thresholded distance not only improves accuracy, but reduces the run time, using our algorithm. The speed can be further improved using techniques such as Bayesian sequential hypothesis testing [27]. The project homepage, including code (C++ and Matlab wrappers) is at:

www.cs.huji.ac.il/~7e0firpele/FastEMD/.

References

- [1] R. Ahuja, A. Goldberg, J. Orlin, and R. Tarjan. Finding minimum-cost flows by double scaling. *Mathematical Programming*, 1992.
- [2] R. Ahuja, T. Magnanti, and J. Orlin. *Network flows: theory, algorithms, and applications*. 1993.
- [3] R. Ahuja, K. Mehlhorn, J. Orlin, and R. Tarjan. Faster algorithms for the shortest path problem. *JACM*, 1990.
- [4] N. Alon, S. Cosares, D. S. Hochbaum, and R. Shamir. An algorithm for the detection and construction of monge sequences. *LAA*, 1989.
- [5] A. Andoni, P. Indyk, and R. Krauthgamer. Earth mover distance over high-dimensional spaces. In *SODA*, 2008.
- [6] K. D. Ba, H. L. Nguyen, H. N. Nguyen, and R. Rubinfeld. Sublinear time algorithms for earth mover's distance. *CoRR*, 2009.
- [7] P. Ciaccia, M. Patella, and P. Zezula. M-tree: An Efficient Access Method for Similarity Search in Metric Spaces. In *ICVLDB*, 1997.
- [8] R. Collins and W. Ge. CSDD Features: Center-Surround Distribution Distance for Feature Extraction and Matching. In *ECCV*, 2008.
- [9] J. Edmonds and R. Karp. Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems. *JACM*, 1972.
- [10] C. Elkan. Using the Triangle Inequality to Accelerate k-Means. In *ICML*, 2003.
- [11] M. Fredman and R. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *JACM*, 1987.
- [12] K. Grauman and T. Darrell. Fast contour matching using approximate earth mover's distance. In *CVPR*, 2004.
- [13] K. Grauman and T. Darrell. The pyramid match kernel: Efficient learning with sets of features. *JMLR*, 2007.
- [14] J. Gudmundsson, O. Klein, C. Knauer, and M. Smid. Small Manhattan Networks and Algorithmic Applications for the Earth Movers Distance. In *EWCG*, 2007.
- [15] M. Hein, O. Bousquet, and B. Schölkopf. Maximal margin classification for metric spaces. *JCSS*, 2005.
- [16] A. Hoffman. On simple linear programming problems. In *SPM*, 1963.
- [17] P. Indyk and N. Thaper. Fast image retrieval via embeddings. In *IWSCVT*, 2003.
- [18] S. Khot and A. Naor. Nonembeddability theorems via Fourier analysis. *Mathematische Annalen*, 2006.
- [19] S. Lazebnik, C. Schmid, and J. Ponce. A sparse texture representation using local affine regions. *PAMI*, 2005.
- [20] H. Ling and K. Okada. An Efficient Earth Mover's Distance Algorithm for Robust Histogram Comparison. *PAMI*, 2007.
- [21] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.
- [22] M. Luo, G. Cui, and B. Rigg. The Development of the CIE 2000 Colour-Difference Formula: CIEDE2000. *CRA*, 2001.
- [23] Q. Lv, M. Charikar, and K. Li. Image similarity search with compact data structures. In *ICIKM*, 2004.
- [24] G. Monge. Déblai et remblai. *Mémoires de l'Académie des Sciences*, 1781.
- [25] J. Orlin. A faster strongly polynomial minimum cost flow algorithm. In *STOC*, 1988.
- [26] O. Pele and M. Werman. A linear time histogram metric for improved sift matching. In *ECCV*, 2008.
- [27] O. Pele and M. Werman. Robust real time pattern matching using bayesian sequential hypothesis testing. *PAMI*, 2008.
- [28] S. Peleg, M. Werman, and H. Rom. A unified approach to the change of resolution: Space and gray-level. *PAMI*, 1989.
- [29] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *IJCV*, 2000.
- [30] M. Ruzon and C. Tomasi. Edge, Junction, and Corner Detection Using Color Distributions. *PAMI*, 2001.
- [31] R. Sandler and M. Lindenbaum. Nonnegative Matrix Factorization with Earth Movers Distance Metric. In *CVPR*, 2009.
- [32] G. Sharma, W. Wu, and E. Dalal. The CIEDE2000 color-difference formula: implementation notes, supplementary test data, and mathematical observations. *CRA*, 2005.
- [33] H. Shen and A. Wong. Generalized texture representation and metric. *CVGIP*, 1983.
- [34] R. Shepard. Toward a universal law of generalization for psychological science. *Science*, 1987.
- [35] S. Shirdhonkar and D. Jacobs. Approximate earth movers distance in linear time. In *CVPR*, 2008.
- [36] U. von Luxburg and O. Bousquet. Distance-Based Classification with Lipschitz Functions. *JMLR*, 2004.
- [37] J. Wang, J. Li, and G. Wiederhold. SIMPLiCity: Semantics-Sensitive Integrated Matching for Picture Libraries. *PAMI*, 2001.
- [38] M. Werman, S. Peleg, R. Melter, and T. Kong. Bipartite graph matching for points on a line or a circle. *JoA*, 1986.
- [39] M. Werman, S. Peleg, and A. Rosenfeld. A distance metric for multidimensional histograms. *CVGIP*, 1985.
- [40] D. Xu, T. Cham, S. Yan, and S. Chang. Near Duplicate Image Identification with Spatially Aligned Pyramid Matching. In *CVPR*, 2008.
- [41] P. Yianilos. Data structures and algorithms for nearest neighbor search in general metric spaces. In *SODA*, 1993.
- [42] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *IJCV*, 2007.

Chapter 4

The Quadratic-Chi Histogram Distance Family

The Quadratic-Chi Histogram Distance Family

Ofir Pele and Michael Werman

School of Computer Science
The Hebrew University of Jerusalem
{ofirpele,werman}@cs.huji.ac.il

Abstract. We present a new histogram distance family, the Quadratic-Chi (QC). QC members are Quadratic-Form distances with a cross-bin χ^2 -like normalization. The cross-bin χ^2 -like normalization reduces the effect of large bins having undo influence. Normalization was shown to be helpful in many cases, where the χ^2 histogram distance outperformed the L_2 norm. However, χ^2 is sensitive to quantization effects, such as caused by light changes, shape deformations etc. The Quadratic-Form part of QC members takes care of cross-bin relationships (e.g. red and orange), alleviating the quantization problem. We present two new cross-bin histogram distance properties: *Similarity-Matrix-Quantization-Invariance* and *Sparseness-Invariance* and show that QC distances have these properties. We also show that experimentally they boost performance. QC distances computation time complexity is linear in the number of non-zero entries in the bin-similarity matrix and histograms and it can easily be parallelized. We present results for image retrieval using the Scale Invariant Feature Transform (SIFT) and color image descriptors. In addition, we present results for shape classification using Shape Context (SC) and Inner Distance Shape Context (IDSC). We show that the new QC members outperform state of the art distances for these tasks, while having a short running time. The experimental results show that both the cross-bin property and the normalization are important.

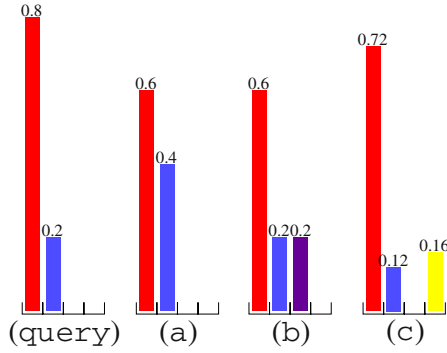
1 Introduction

It is common practice to use bin-to-bin distances such as the L_1 and L_2 norms for comparing histograms. This practice assumes that the histogram domains are aligned. However this assumption is violated in many cases due to quantization, shape deformation, light changes, etc. Bin-to-bin distances depend on the number of bins. If it is low, the distance is robust, but not discriminative, if it is high, the distance is discriminative, but not robust. Distances that take into account cross-bin relationships (cross-bin distances) can be both robust and discriminative.

There are two kinds of cross-bin distances. The first is the Quadratic-Form distance [1]. Let P and Q be two histograms and A the bin-similarity matrix. The Quadratic-Form distance is defined as:

$$\text{QF } (P, Q) = \sqrt{(P - Q)^T A (P - Q)} \quad (1)$$

When the bin-similarity matrix A is the inverse of the covariance matrix, the Quadratic-Form distance is called the Mahalanobis distance. If the bin-similarity matrix is positive-definitive, then the Quadratic-Form distance is a metric. In this case the



	(a)-(c) ordered by their distance (in small fonts) to the (query)					
distance						
QCN(our)	(a)	0.35	(b)	0.62	(c)	0.86
QCS(our)	(a)	0.31	(b)	0.41	(c)	0.43
QF	(c)	0.20	(a)	0.28	(b)	0.28
EMD	(c)	3.20	(a)	4.00	(b)	4.00
L_1	(c)	0.32	(a)	0.40	(b)	0.40
χ^2	(a)	0.05	(c)	0.09	(b)	0.11

Fig. 1. This figure should be viewed in color, preferably on a computer screen. A toy example showing the behavior of distances that reduce the effect of large bins and the behavior of distances that take cross-bin relationships into account. We show four color histograms, each histogram has four colors: red, blue, purple, and yellow. The Quadratic-Form (QF), the Earth Mover Distance (EMD) and the L_1 norm do not reduce the effect of large bins. Thus, they rank (query) to be more similar to (c) than to (a). χ^2 considers (a) to be more similar, but as it does not take cross-bin relationships into account it fails with (b). Our proposed members of the Quadratic-Chi histogram distance family, QCN and QCS consider (a) to be most similar, (b) the second and (c) the least similar as they take into account cross-bin relationships and reduce the effect of large bins, using an appropriate normalization.

Quadratic-Form distance is the L_2 norm between linear transformations of P and Q . If the bin-similarity matrix is positive-semidefinite, then the Quadratic-Form distance is a semi-metric.

The second type of distance that takes into account cross-bin relationships is the Earth Mover's Distance (EMD). EMD was defined by Rubner et al. [2] as the minimal cost that must be paid to transform one histogram (P) into the other (Q):

$$\text{EMD}(P, Q) = \left(\min_{\{f_{ij}\}} \sum_{i,j} f_{ij} D_{ij} \right) / \left(\sum_{i,j} f_{ij} \right) \quad s.t. \quad f_{ij} \geq 0$$

$$\sum_j f_{ij} \leq P_i \quad \sum_i f_{ij} \leq Q_j \quad \sum_{i,j} f_{ij} = \min\left(\sum_i P_i, \sum_j Q_j\right) \quad (2)$$

where $\{f_{ij}\}$ denotes the flows. Each f_{ij} represents the amount transported from the i th supply to the j th demand. We call D_{ij} the *ground distance* between bin i and bin j . If D_{ij} is a metric, the EMD as defined by Rubner is a metric only for normalized histograms. Recently Pele and Werman [3] suggested EMD:

$$\text{EMD}_\alpha(P, Q) = \left(\min_{\{f_{ij}\}} \sum_{i,j} f_{ij} D_{ij} \right) + \left| \sum_i P_i - \sum_j Q_j \right| \alpha \max_{i,j} D_{ij}$$

$$s.t. \quad \text{EMD constraints} \quad (3)$$

If D_{ij} is a metric and $\alpha \geq \frac{1}{2}$, EMD is a metric for all histograms [3]. For normalized histograms EMD and EMD are equal (e.g. Fig. 1).

In many natural histograms the difference between large bins is less important than the difference between small bins and should be reduced. See for example Fig. 1. The Chi-Squared (χ^2) is a histogram distance that takes this into account. It is defined as:

$$\chi^2(P, Q) = \frac{1}{2} \sum_i \frac{(P_i - Q_i)^2}{(P_i + Q_i)} \quad (4)$$

The χ^2 histogram distance comes from the χ^2 test-statistic [4] where it is used to test the fit between a distribution and observed frequencies. In this paper the histograms are not necessarily normalized, and thus not probabilities vectors. χ^2 was successfully used for texture and object categories classification [5,6,7], near duplicate image identification [8], local descriptors matching [9], shape classification [10,11] and boundary detection [12]. The χ^2 , like other bin-to-bin distances such as the L_1 and the L_2 norms, is sensitive to quantization effects.

2 Our Contribution

In this paper we present a new cross-bin histogram distance family: Quadratic-Chi (QC). Like the Quadratic-Form, its members take cross-bin relationships into account. Like the χ^2 , its members reduce the effect of differences caused by bins with large values. We discuss QC members' properties, including a formalization of a two new cross-bin histogram distance properties: *Similarity-Matrix-Quantization-Invariance* and *Sparseness-Invariance*. We show that all QC members and the EMD have these properties. We also show importance experimentally.

For full histograms QC distances computation time is linear in the number of non-zero entries in the bin-similarity matrix. In this case, QC distances can be implemented with 5 lines of Matlab code (see Algorithm 1). For two sparse histograms (for example bag-of-words histograms) with a total of n non-zeros entries and an average of K non-zeros entries in each row of the similarity matrix, a QC distance computation time complexity is $O(n/K)$. See code (C++ and Matlab wrappers) at:

<http://www.cs.huji.ac.il/~ofirpele/QC/>. Finally, QC distances' parallelization is trivial.

We present results for image retrieval on the Corel dataset using the SIFT descriptor [13] and small color images. We also present results for shape classification using

Algorithm 1. Quadratic-Chi Matlab Code for Full Histograms

```
function dist= QC(P,Q,A,m)

Z= (P+Q)*A;
% 1 can be any number as Z_i==0 iff D_i=0
Z(Z==0)= 1;
Z= Z.^m;
D= (P-Q)./Z;
% max is redundant if A is positive-semidefinite
dist= sqrt( max(D*A*D',0) );
```

Shape Context (SC) [10] and Inner Distance Shape Context (IDSC) [11]. QC members performance is excellent. They outperform state of the art distances including χ^2 , QF, L_1 , L_2 , EMD[14], SIFT_{DIST}[3], EMD- L_1 [15], Diffusion[16], Bhattacharyya [17], Kullback-Leibler[18] and Jensen-Shannon[19] while having a short running time. We have found that the normalization is very important. Surprisingly, excellent performance was achieved using a new bin-to-bin distance from the QC family, that has a large normalization factor. Its cross-bin version yielded an additional improvement, outperforming all other distances for SIFT, SC and IDSC.

3 The Quadratic-Chi Histogram Distance Family

3.1 The Quadratic-Chi Histogram Distance Definition

Let P and Q be two non-negative bounded histograms. That is, $P, Q \in [0, U]^N$. Let A be a non-negative symmetric bounded bin-similarity matrix such that each diagonal element is bigger or equal to every other element in its row (this demand is weaker than being a strongly dominant matrix). That is, $A \in [0, U]^N \times [0, U]^N$ and $\forall i, j \ A_{ii} \geq A_{ij}$. Let $0 \leq \alpha < 1$ be the normalization factor. A Quadratic-Chi (QC) histogram distance is defined as:

$$QC_m^A(P, Q) = \sum_{ij} \frac{P_i - Q_i}{\sum_c (P_c + Q_c) A_{ci}}^m \frac{P_j - Q_j}{\sum_c (P_c + Q_c) A_{cj}}^m A_{ij} \quad (5)$$

where we define $\frac{0}{0} = 0$. If A is positive-semidefinite, the argument inside the square root (the sum) is non-negative. If A is not positive-semidefinite we can get non-real (complex) distances. This is true also for the Quadratic-Form (Eq. 1). We prefer not to restrict ourselves to positive-semidefinite matrices. On the other hand, we don't want non-real distances. So, we define a complex distance as zero. In practice, this was never needed, even with non-positive-semidefinite matrices. This is due to the fact that the eigenvectors of the similarity matrices corresponding to negative eigenvalues were very far from smooth, while the difference vector for natural histograms P and Q is usually very smooth, see Fig. 2.

Each addend's denominator inside the square root is zero if and only if the addend's numerator is zero. A $QC_m^A(P, Q)$ distance is continuous. In particular, if the addend's denominator tends to zero, the whole addend tends to zero. Proofs are in [20].

The Quadratic-Chi distance family generalizes both the Quadratic-Form (QF) and a monotonic transformation of χ^2 . That is, $QC_0^A(P, Q) = QF^A(P, Q)$ and if I is the identity matrix, $QC_{0.5}^I(P, Q) = \sqrt{2\chi^2(P, Q)}$.

3.2 Metric Properties

There are three conditions for a distance function, $\mathcal{D}$, to be a semi-metric. The first is *non-negativity* (i.e. $\mathcal{D}(P, Q) \geq 0$), the second is *symmetry* (i.e. $\mathcal{D}(P, Q) = \mathcal{D}(Q, P)$) and the third is *subadditivity* (i.e. $\mathcal{D}(P, Q) \leq \mathcal{D}(P, K) + \mathcal{D}(K, Q)$). $\mathcal{D}$ is a metric if it is

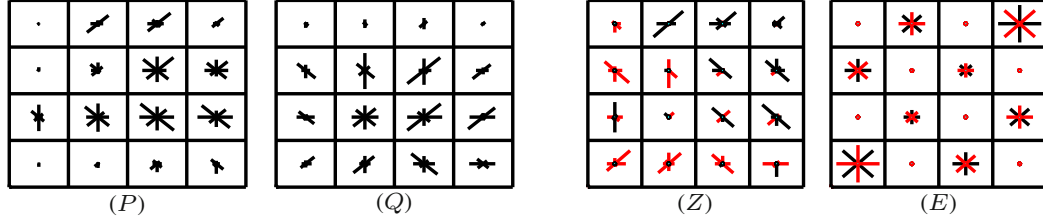


Fig. 2. This figure illustrates why it is not likely to get negative values in the square root argument of a QC distance for natural histograms and a typical similarity matrix. P and Q are two SIFT histograms. Z is the normalized difference vector. That is: $Z_i = \frac{i}{\sum_c c_{ci} m} - \frac{i}{\sum_c c_{ci} m}$. Negative values are represented with red, positive values are represented with black. E is one of the eigenvectors of the similarity matrix that we used in the experiments which correspond to a negative eigenvalue. Z is very smooth while E is very non-smooth. This is typical of eigenvectors with negative values with typical parameters.

a semi-metric and it also has the property of *identity of indiscernibles* (i.e. $\mathcal{D}(P, Q) = 0$ if and only if $P = Q$).

A QC_m^A distance without the square root, is non-negative if the bin-similarity matrix, A , is positive-semidefinite. If A is positive-definitive, then it also has the property of *identity of indiscernibles*. This follows directly from the fact that the argument inside the square root in a QC histogram distance is a quadratic-form between two vectors. A QC histogram distance is symmetric if the bin-similarity matrix, A , is symmetric.

We now discuss subadditivity (i.e. $\mathcal{D}(P, Q) \leq \mathcal{D}(P, K) + \mathcal{D}(K, Q)$) for several distances. The χ^2 histogram distance is not subadditive. For example let $i = 0, k = 1, j = 2$ we get $\chi^2(i, j) = 1 > \chi^2(i, k) + \chi^2(k, j) = \frac{2}{3}$. However, $\sqrt{\chi^2}$ is subadditive for one and two dimensional non-negative histograms (verified by analysis). Experimentally it appears that $\sqrt{\chi^2}$ is subadditive for an N -dimensional non-negative histograms. Experimentally, QC members with the identity matrix seems to be subadditive for non-negative histograms. However, QC members with some positive-definitive bin-similarity matrices are not subadditive. The question when the QC histogram distances are subadditive is currently unresolved. An additional discussion about triangle inequality can be found in Jacobs et al. [21].

4 Cross-Bin Histogram Distance Properties

4.1 The Similarity-Matrix-Quantization-Invariance Property

The *Similarity-Matrix-Quantization-Invariance* property ensures that if two bins in the histograms have been erroneously quantized, this will not affect the distance. Mathematically we define this as:

Definition 1. Let $\mathcal{D}$ be a cross-bin histogram distance between two histograms P and Q and let A be the bin-similarity/distance matrix. We assume P, Q and A are non-negative and that A is symmetric. Let $A_{k,:}$ be the k th row of A . Let $\mathbf{1} = [1, \dots, N]$

be a non-negative vector and $0 \leq \alpha \leq 1$. We define $\alpha^{k,b} = [\dots, \alpha_k, \dots, b + (1 - \alpha)_k, \dots]$. That is, $\alpha^{k,b}$ is a transformation of α where $(1 - \alpha)_k$ mass has moved from bin k to bin b . We define $\mathcal{D}$ to be *Similarity-Matrix-Quantization-Invariant* if:

$$A' = A \Rightarrow \forall 0 \leq \alpha \leq 1, 0 \leq \beta \leq 1 \quad \mathcal{D}(P, Q) = \mathcal{D}(P^{\alpha, \beta}, Q^{\alpha, \beta}) \quad (6)$$

We prove that EMD, EMD and all the Quadratic-Chi histogram distances are *Similarity-Matrix-Quantization-Invariant* in the appendix [20].

4.2 The Sparseness-Invariance Property

The *Sparseness-Invariance* property ensures that distances between sparse histograms will be equal to distances between full histograms. Mathematically we define this as:

Definition 2. Let $\mathcal{D}$ be a cross-bin histogram distance between two histograms $P \in \mathbb{R}^N$ and $Q \in \mathbb{R}^N$ and let A be the $N \times N$ bin similarity/distance matrix. Let A' be any $(N + 1) \times (N + 1)$ matrix whose upper-left sub-matrix equals A . We define $\mathcal{D}$ to be *Sparseness-Invariant* if:

$$\mathcal{D}([P_1, \dots, P], [Q_1, \dots, Q]) = \mathcal{D}'([P_1, \dots, P, 0], [Q_1, \dots, Q, 0]) \quad (7)$$

QC members, EMD and the EMD are *Sparseness-Invariant* directly from their definitions. A stronger property called *Extension-Invariance* was proposed by D'Agostino and Dardanoni for bin-to-bin distances [22]. This property requires that, if both histograms are extended by concatenating each of them with the same vector (not necessarily zeros), the distance is left unaltered. Cross-bin distances assumes dependence between histogram bins, thus this requirement is too strong for them.

4.3 Cross-Bin Histogram Distance Properties Discussion

A *Sparseness-Invariant* cross-bin histogram distance does not depend on the specific representation of the histograms (full or sparse). A *Similarity-Matrix-Quantization-Invariant* cross-bin histogram distance encompass its cross-bin relationships only in the bin-similarity matrix. Intuitively such properties are desirable. In the appendix [20], we compare experimentally distances which resembles QC distances, but are either not *Similarity-Matrix-Quantization-Invariant* or not *Sparseness-Invariant*. The comparison shows that these properties considerably boost performance (especially for sparse color histograms).

Rubner et al. [2,23] claim that one of the key advantages of the Earth Mover's Distance is that each compared object may be represented by an individual (possibly with a different number of bins) binning that is adapted to its specific distribution. The Quadratic-Form is regarded as not having this property (see for example, Table 1 in [23]). Since all the Quadratic-Chi histogram distances (including the Quadratic-Form) are both *Similarity-Matrix-Quantization-Invariant* and *Sparseness-Invariant* there is no obstacle to using them with individual binning; *i.e.* to use them to compare histograms that were adapted to each object individually.

Similarity-Matrix-Quantization-Invariant and *Sparseness-Invariant* can contradict. For example, any distance applied to the transformed vectors $P'_i = \sum_c (P_c) A_{ci}$ and $Q'_i = \sum_c (Q_c) A_{ci}$ is *Similarity-Matrix-Quantization-Invariant*. However the χ^2 distance between P' and Q' is not *Sparseness-Invariant* (with respect to P and Q).

5 Implementation Notes

5.1 The Similarity Matrix and the Normalization Factor

It is desirable to have a transformation from a distance matrix into a similarity matrix, as many spaces are equipped with a useful distance (*e.g.* color space [24]). Hafner et al. [1] proposed this transformation:

$$A_{ij} = 1 - \frac{D_{ij}}{\max_{ij}(D_{ij})} \quad (8)$$

Another possibility for choosing a similarity matrix is by using cross validation. However, we think that like for the Quadratic-Form, learning the similarity matrix (and for QC also the normalization factor) will be the best way to adjust them. This is left for future work. Currently we suggest to use thresholded ground distances as was used in [2,25,3,14] and choosing the normalization factor by cross validation.

5.2 Efficient Online Bin-Similarity Matrix Computation

For a fixed histogram configuration (*e.g.* SIFT, SC and IDSC) the bin-similarity matrix can be pre-computed once. Then, each distance computation is linear in the number of non-zero entries in the bin-similarity matrix.

There are cases where the bin-similarity matrix can not be pre-computed. For example, in our color experiments (Section 6.1), we used $N \times M$ color images as sparse histograms. That is, the query histogram was: $[1, \dots, 1, 0, \dots, 0]$ and each image being compared to the query was represented by the histogram: $[0, \dots, 0, 1, \dots, 1]$. Note that the full histogram dimension is $M \times N \times 256^3$, computing an $(M \times N \times 256^3)^2$ similarity matrix offline is not feasible. We can compute the similarity online for each pair of sparse histograms in $O((NM)^2)$ time. We now discuss how to do it more efficiently.

If we are comparing two images (as in Section 6.1) we can use a similarity matrix that gives far-away pixels zero similarity (see Eq. 10). Then, we can simply compare each pixel in one image to its corresponding $T \times T$ spatial neighbors in the second image. This reduces running time to $O(NMT^2)$. Using this technique, it is important to use a sparse representation for the bin-similarity matrix.

6 Results

We present results using the newly defined distances and state of the art distances, for image retrieval using SIFT-like descriptors and color image descriptors. In addition, we present results for shape classification using Inner Distance Shape Context (IDSC). More results for shape classification using SC, can be found in the appendix [20].

6.1 Image Retrieval Results

In this section we present results for image retrieval using the same benchmark as Pele and Werman [14]. We employed a database that contained 773 landscape images from the COREL database that were also used in Wang et al. [26]. The dataset has 10 classes¹:

¹ The original database contains some visually ambiguous classes such as Africa that also contains images of beaches in Africa. We used the filtered image dataset that was downloaded from: <http://www.cs.huji.ac.il/~ofirpele/FastEMD/>

People in Africa, Beaches, Outdoor Buildings, Buses, Dinosaurs, Elephants, Flowers, Horses, Mountains and Food. The number of images in each class ranges from 50 to 100. From each class we selected 5 images as query images (images 1, 10, ..., 40). Then we searched for the 50 nearest neighbors for each query image. We computed the distance of each image to the query image and its reflection and took the minimum. We present results for two types of image representations: SIFT-like descriptors and small L^*a*b^* images.

SIFT-like Descriptors. The first representation - SIFT is a $6 \times 8 \times 8$ SIFT descriptor [13] computed globally on the whole image. The second representation - CSIFT is a SIFT-like descriptor on a color-edge image. See [14] for more details.

We experimented with two new types of QC distances. The first is $QC_{0.5}^A$, which is a cross-bin generalization of $\sqrt{2\chi^2}$, which we call Quadratic-Chi-Squared (QCS). The second is $QC_{0.9}^A$, which has a larger normalization factor, which we call Quadratic-Chi-Normalized (QCN). We do not use QC_m^A with $m \geq 1$ due to discontinuity problems, see appendix [20] (practically, QC_1^A had slightly poorer results compared to $QC_{0.9}^A$). We also experimented with the Quadratic-Form (QF) distance which is QC_0^A . For all of these distances we used the bin-similarity matrix in Eq. 8. Let $M = 8$ be the number of orientation bins, as in Pele and Werman [14], the ground distance between bins (x_i, y_i, o_i) and (x_j, y_j, o_j) is:

$$d_T(i, j) = \min \quad ||(x_i, y_i) - (x_j, y_j)||_2 + \min(|o_i - o_j|, M - |o_i - o_j|) \quad , \quad T \quad (9)$$

We also used the identity matrix as a similarity matrix for all the above distances. We also compared to L_2 and χ^2 . $QF^I = L_2$, and nearest neighbors of χ^2 and QCS^I are the same.

We also compared to four EMD variants. The first was EMD_1^D with $D = d_T$ (Eq. 9) as in Pele and Werman [3]. The second was the L_1 norm which is equal to $EMD_{0.5}^D$ with D equals to the Kronecker delta multiplied by two. The third is $SIFT_{DIST}$ [3] which is the sum of EMD over all the spatial cells (each spatial cell contains one orientation histogram). The ground distance for the orientation histograms is: $\min(|o_i - o_j|, M - |o_i - o_j|, 2)$ (M is the number of orientation bins). The fourth was the $EMD-L_1$ [15] which is EMD with L_1 as the ground distance. We also tried non-thresholded ground distances (which produce non-sparse similarity matrices). However, the results were poor. This is in line with Pele and Werman's findings that cross-bin distances should be used with thresholded ground distances [14]. Finally, we compared to the Diffusion distance proposed by Ling and Okada [16] and to three probabilistic based distances: Bhattacharyya [17], Kullback-Leibler (KL) [18] and Jensen-Shannon (JS) [19] (we added Matlab's epsilon to all histogram bins when computing KL and JS throughout the paper, as they are not well defined if there is a zero bin, without doing so accuracy was very low).

For each distance measure, we present the descriptor (SIFT/CSIFT) with which it performed best. The results for all the pairs of descriptors and distance measures can be found in the appendix[20]. The results are presented in Fig. 3(a) and show that $QCN^{1-\frac{d_{T=2}}{2}}$ (QCN with the similarity matrix: $A_{ij} = 1 - \frac{d_{T=2}(i,j)}{2}$) outperformed all other methods. $EMD_1^{d_{T=2}}$ ranked second. The computation of $QCN^{1-\frac{d_{T=2}}{2}}$ was 266

times faster than $\text{EMD}_1^{dT=2}$, see Table 2 in page 760. QCN^I ranked third, which shows the importance of the normalization factor.

All cross-bin distances that use thresholded ground distances outperformed their bin-by-bin versions. The figure also shows that χ^2 and QF improve upon L_2 . QCN and QCS which are mathematically sound combinations of χ^2 and QF outperformed both.

L*a*b* Images. Our second type of image representation is a small L*a*b* image. We resized each image to 32×48 and converted them to L*a*b* space. The state of the art color distance is Δ_{00} - CIEDE2000 on L*a*b* color space[24,27]. As it is meaningful only for small distances we threshold it (as in [2,25,14]).

Again, we experimented with QCS, QCN and QF distances using the bin-similarity matrix in Eq. 8. The ground distance between two pixels $(x_i, y_i, L_i, a_i, b_i)$, $(x_j, y_j, L_j, a_j, b_j)$:

$$s(i, j) = \|(x_i, y_i) - (x_j, y_j)\|_2$$

$$\text{dc}_{T_1, T_2}(i, j) = \begin{cases} \min((s(i, j) + \Delta_{00}((L_i, a_i, b_i), (L_j, a_j, b_j))), T_1) & \text{if } s(i, j) \leq T_2 \\ T_1 & \text{otherwise} \end{cases} \quad (10)$$

This distance is similar to the one used by [14], except that distances with spatial difference larger than the threshold T_2 are set to the maximum threshold T_1 . This was done to accelerate the online computation of the bin-similarity matrix. The accuracy using this distance is the same as using the distance from Pele and Werman [14]. See appendix [20]. We also used EMD with dc_{T_1, T_2} (Eq. 10) as a ground distance. Let I_1, I_2 be the two L*a*b* images. We also used the following distances:

$$L_1 \Delta_{00} = \sum_{x, y} (\Delta_{00}(I_1(x, y), I_2(x, y))) \quad L_1 \Delta_{00}^T = \sum_{x, y} (\min(\Delta_{00}(I_1(x, y), I_2(x, y)), T))$$

$$L_2 \Delta_{00} = \sum_{x, y} (\Delta_{00}(I_1(x, y), I_2(x, y)))^2 \quad L_2 \Delta_{00}^T = \sum_{x, y} (\min(\Delta_{00}(I_1(x, y), I_2(x, y)), T))^2$$

QCN^I , χ^2 , L_2 , L_1 , $\text{SIFT}_{\text{DIST}}$ [3], $\text{EMD}-L_1$ [15], the Diffusion[16], Bhattacharyya [17], KL [18] and JS [19] distances cannot be applied to L*a*b* images as they are either bin-to-bin distances or applicable only to Manhattan networks.

We present results in Fig. 3. As shown, $\text{QCS}^{1 - \frac{\text{dc}_{T_1=20, T_2=5}}{20}}$ and $\text{EMD}_1^{\text{dc}_{T_1=20, T_2=5}}$ [14] distances ranked first. $\text{QCS}^{1 - \frac{\text{dc}_{T_1=20, T_2=5}}{20}}$ ran 300 times faster (see Table 2). However, since the computation of the bin-similarity matrix cannot be offline here, the real gain is a factor of 17. The $\text{QF}^{1 - \frac{\text{dc}_{T_1=20, T_2=5}}{20}}$ distance ranked last, which shows the importance of the normalization factor of the QC histogram members.

Although a QC distance alleviates quantization problems, EMD does it better, instead of matching everything to everything it finds the optimal matching. EMD however, does not reduce the effect of large bins. We conjecture that a variant of EMD which will reduce the effect of large bins will have an excellent performance.

6.2 Shape Classification Results

In this section we present results for shape classification using the same framework as Ling et al. [11,15,28]. We test for shape classification with the Inner Distance Shape

Table 1. Shape classification results. $\text{QCN}^{1-\frac{\text{dsc}_T=2}{2}}$ outperformed all other distances.

	Top 1	Top 2	Top 3	Top 4	AUC%
$\text{QCN}^{1-\frac{\text{dsc}_T=2}{2}}$	39	38	38	34	0.950
QCN^I	40	37	36	33	0.940
$\text{QCS}^{1-\frac{\text{dsc}_T=2}{2}}$	39	35	38	28	0.912
QCS^I	40	34	37	27	0.907
χ^2	40	36	36	21	0.902
$\text{QF}^{1-\frac{\text{dsc}_T=2}{2}}$	40	34	39	19	0.897
L_2	39	35	35	18	0.873

	Top 1	Top 2	Top 3	Top 4	AUC%
$\widehat{\text{EMD}}_1^{\text{dsc}_T=2}$	39	36	35	27	0.902
L_1	39	35	35	25	0.890
$\text{SIFT}_{\text{DIST}}[3]$	38	37	27	22	0.848
$\text{EMD-}L_1[15]$	39	35	38	30	0.917
$\text{Diffusion}[16]$	39	35	34	23	0.880
$\text{Bhattacharyya}[17]$	40	37	32	23	0.895
$\text{KL}[18]$	40	38	36	29	0.938
$\text{JS}[19]$	40	35	37	21	0.900

Context (IDSC) [11]. The original Shape Context (SC) descriptor was proposed by Belongie et al. [10]. Belongie et al. [10] and Ling and Jacobs [11] used the χ^2 distance for comparing shape context histograms. Ling and Okada [15] showed that replacing χ^2 with $\text{EMD-}L_1$ improves results. We show that QC members yields the best results.

We tested on the articulated shape data set [11,28], that contains 40 images from 8 different objects. Each object has 5 images articulated to different degrees. The dataset is very challenging because of the similarity between different objects. The original SC had a very poor performance on this dataset, see appendix [20].

Again, we experimented with QCS, QCN and QF distances with the bin-similarity matrix in Eq. 8. The ground distance between two bins $(i, o_i), (j, o_j)$ was (M is the number of orientation bins):

$$\text{dsc}_T(i, j) = \min(|d_i - d_j| + \min(|o_i - o_j|, M - |o_i - o_j|), T) \quad (11)$$

We also used the identity matrix as a similarity matrix, and thus we also compare to L_2 . χ^2 and QCS^I distances are not equivalent here as the distance is not used for nearest neighbors. We refer the reader to Belongie et al. paper to see its usage [10]. Practically, QCS^I slightly outperformed χ^2 in this task, see Table 1.

We also compared to four EMD variants: $\widehat{\text{EMD}}_1^D$ with $D = \text{dsc}_T$ (Eq. 11), the L_1 norm, $\text{SIFT}_{\text{DIST}}[3]$ and $\text{EMD-}L_1[15]$. Finally, we compared to the Diffusion distance proposed by Ling and Okada [16] and to three probabilistic based distances: Bhattacharyya [17], Kullback-Leibler (KL) [18] and Jensen-Shannon (JS) [19].

To evaluate results, for each image, the four most similar matches are chosen from other images in the dataset. The retrieval result is summarized as the number of 1st, 2nd, 3rd and 4th most similar matches that come from the correct object. Table 1 shows the retrieval results. The $\text{QCN}^{1-\frac{\text{dsc}_T=2}{2}}$ outperformed all the other methods. QCN^I performance is again excellent, which shows the importance of the normalization factor.

Again all cross-bin distances outperformed their bin-by-bin versions. Again, χ^2 and QF improved upon L_2 . QCN and QCS which are mathematically sound combinations of χ^2 and QF outperformed both.

6.3 Running Time Results

All runs were conducted on a Pentium 2.8GHz. A comparison of the practical running time of all distances is given in Table 2. Clearly QCN and QCS distances are fast to

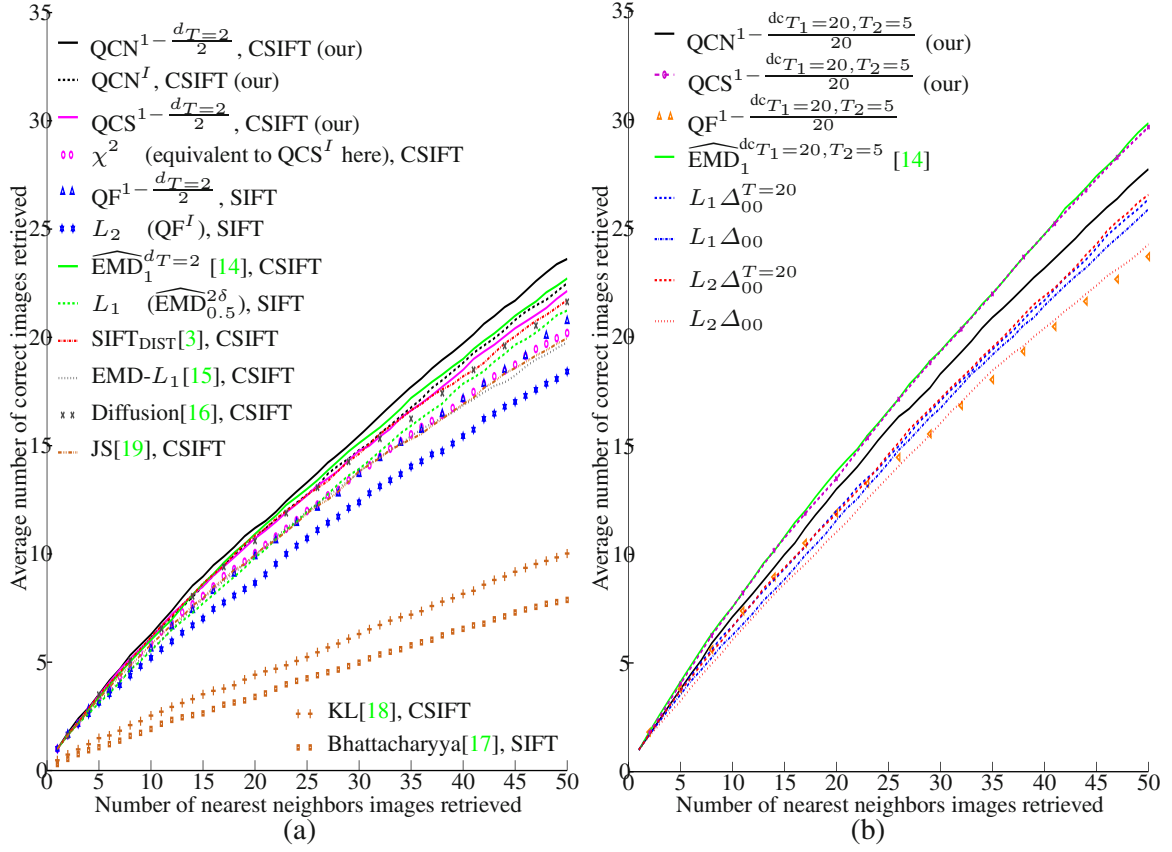


Fig. 3. Results for image retrieval.

(a) **SIFT-like descriptors.** For each distance measure, we present the descriptor (SIFT/CSIFT) with which it performed best. The results for all the pairs of descriptors and distance measures can be found in the appendix[20]. There are several key observations. First, the QC members performance is excellent. $\text{QCN}^{1 - \frac{dT=2}{2}}$ (QCN with the similarity matrix: $A_{ij} = 1 - \frac{T=2}{2} \delta_{i,j}$) outperformed all other distances. $\widehat{\text{EMD}}_1^{T=2}$ ranked second, but its computation was 266 times slower than $\text{QCN}^{1 - \frac{dT=2}{2}}$ computation (see Table 2). Second, all cross-bin versions of the distances (with d_T or a transformation of it) performed better than their bin-by-bin versions (with the identity matrix or the Kronecker delta function). Third, QCN ranked third, although its a bin-to-bin distance. This shows the importance of the normalization factor. Finally, χ^2 and QF improve upon L_2 . However, χ^2 does not take cross-bin relationships into account and QF does not reduce the effect of large bins. QCS and QCN histogram distances, which are mathematically sound combinations of χ^2 and QF have the two properties and outperformed both.

(b) **L*a*b* images results.** QCN, χ^2 , L_2 , L_1 , SIFT_{DIST}[3], EMD- L_1 [15], Diffusion[16], Bhattacharyya[17], KL[18] and JS[19] distances are not applicable here. $\text{QCS}^{1 - \frac{dcT_1=20, T_2=5}{20}}$ and $\widehat{\text{EMD}}_1^{dcT_1=20, T_2=5}$ [14] ranked first. $\text{QCS}^{1 - \frac{dcT_1=20, T_2=5}{20}}$ computation is 300 times faster than $\widehat{\text{EMD}}_1^{dcT_1=20, T_2=5}$ without taking the bin-similarity matrix computation into account and 17 times faster when it is taken into account (see Table 2). $\text{QF}^{1 - \frac{dcT_1=20, T_2=5}{20}}$ ranked last, which shows the importance of the normalization factor in QC members.

Table 2. (SIFT) 384-dimensional SIFT-like descriptors matching time (in *milliseconds*). The distances from left to right are the same as the distances in Fig. 3 (a) from up to down. (IDSC) 60-dimensional IDSC histograms matching time (in *microseconds*). The distances from left to right are the same as the distances in Table 1 from up to down. ($L^*a^*b^*$) 32×48 $L^*a^*b^*$ images matching time (in *milliseconds*). The distances from left to right are the same as the distances in Fig. 3 (b) from up to down. In parentheses is the time it takes to compute the distance and the bin-similarity matrix as it cannot be computed offline.

Descriptor	QCN ^{A2}	QCN ^I	QCS ^{A2}	QCS ^I	χ^2	QF ^{A2}	L_2	$\widehat{\text{EMD}}^{D_2}$ [14]	L_1	SIFT _{DIST} [3]
(SIFT)	0.15	0.1	0.07	0.014	0.013	0.05	0.011	40	0.011	0.07
(IDSC)	6.41	2.99	2.32	0.35	0.34	1.25	0.14	133.75	0.32	0.31

Descriptor	EMD- L_1 [15]	Diffusion[16]	JS[19]	KL[18]	Bhattacharyya[17]
(SIFT)	40	0.27	0.088	0.048	0.015
(IDSC)	20.57	3.15	1.40	8.53	17.17

Descriptor	QCN ^{A20}	QCS ^{A20}	QF ^{A20}	$\widehat{\text{EMD}}^{D_{20}}$ [14]	$L_1 \Delta_{00}^{T=20}$	$L_1 \Delta_{00}$	$L_2 \Delta_{00}^{T=20}$	$L_2 \Delta_{00}$
($L^*a^*b^*$)	20 (370)	19 (369)	11 (361)	6000 (6350)	3.2	3.2	3.2	3.2

compute. This is consistent with their linear time complexity. The only non-linear time distances are EMD [14] and EMD- L_1 [15] which are also practically much slower than the other methods. Our method can be easily parallelized, taking advantage of multi-core computers or the GPU.

7 Conclusions

We presented a new cross-bin distance family - the Quadratic-Chi (QC). QC distances have many desirable properties. Like the Quadratic-Form histogram distance they take into account cross-bin relationships. Like χ^2 they reduce the effect of large bins. We formalized two new cross-bin properties, *Similarity-Matrix-Quantization-Invariance* and *Sparseness-Invariance*. QC members were shown to have both. Finally, QC distance computation time is linear in the number of non-zero entries in the bin-similarity matrix. Experimentally, QC outperformed state of the art distances, while having a very short run-time.

There are several open questions that we still need to explore. The first is for which QC distances does the triangle inequality holds for. The second is whether we can change the Earth Mover's Distance so that it will also reduce the effect of large bins. Concave-cost network flow [29] seems to be the right direction for future work although it presents two major obstacles. First, the concave-cost network flow optimization is NP-hard [29]. However, there are available approximations [29,30]. Second, simply using concave-cost flow networks will result in a distance which is not *Similarity-Matrix-Quantization-Invariant*. We would also like to explore whether metric learning methods such as [31,32,33,34,35,36,37,38] can be generalized for the Quadratic-Chi histogram distance. Assent et al. [39] have suggested methods that accelerate database retrieval

that uses Quadratic-Form distances. Generalizing these methods for the Quadratic-Chi distances is of interest. Finally, other computer vision applications such as tracking can use the QC distances. The project homepage, including code (C++ and Matlab wrappers) is at: <http://www.cs.huji.ac.il/~ofirpele/QC/>.

References

1. Hafner, J., Sawhney, H., Equitz, W., Flickner, M., Niblack, W.: Efficient color histogram indexing for quadratic form distance functions. PAMI (1995)
2. Rubner, Y., Tomasi, C., Guibas, L.J.: The earth mover's distance as a metric for image retrieval. IJCV (2000)
3. Pele, O., Werman, M.: A linear time histogram metric for improved sift matching. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 495–508. Springer, Heidelberg (2008)
4. Snedecor, G., Cochran, W.: Statistical Methods, Ames, Iowa, 6th edn. (1967)
5. Cula, O., Dana, K.: 3D texture recognition using bidirectional feature histograms. IJCV (2004)
6. Zhang, J., Marszalek, M., Lazebnik, S., Schmid, C.: Local features and kernels for classification of texture and object categories: A comprehensive study. IJCV (2007)
7. Varma, M., Zisserman, A.: A statistical approach to material classification using image patch exemplars. PAMI (2009)
8. Xu, D., Cham, T., Yan, S., Duan, L., Chang, S.: Near Duplicate Identification with Spatially Aligned Pyramid Matching. In: CSVT (accepted)
9. Forssén, P., Lowe, D.: Shape Descriptors for Maximally Stable Extremal Regions. In: ICCV (2007)
10. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. PAMI (2002)
11. Ling, H., Jacobs, D.: Shape classification using the inner-distance. PAMI (2007)
12. Martin, D., Fowlkes, C., Malik, J.: Learning to detect natural image boundaries using local brightness, color, and texture cues. PAMI (2004)
13. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV (2004)
14. Pele, O., Werman, M.: Fast and robust earth mover's distances. In: ICCV (2009)
15. Ling, H., Okada, K.: An Efficient Earth Mover's Distance Algorithm for Robust Histogram Comparison. PAMI (2007)
16. Ling, H., Okada, K.: Diffusion distance for histogram comparison. In: CVPR (2006)
17. Bhattacharyya, A.: On a measure of divergence between two statistical populations defined by their probability distributions. BCMS (1943)
18. Kullback, S., Leibler, R.: On information and sufficiency. AMS (1951)
19. Lin, J.: Divergence measures based on the Shannon entropy. IT (1991)
20. Pele, O., Werman, M.: The quadratic-chi histogram distance family - appendices (2010), <http://www.cs.huji.ac.il/~ofirpele/publications/ECCV2010app.pdf>
21. Jacobs, D., Weinshall, D., Gdalyahu, Y.: Classification with nonmetric distances: Image retrieval and class representation. PAMI (2000)
22. D'Agostino, M., Dardanoni, V.: What's so special about Euclidean distance? SCW (2009)
23. Rubner, Y., Puzicha, J., Tomasi, C., Buhmann, J.: Empirical evaluation of dissimilarity measures for color and texture. CVIU (2001)
24. Luo, M., Cui, G., Rigg, B.: The Development of the CIE 2000 Colour-Difference Formula: CIEDE2000. CRA (2001)

25. Ruzon, M., Tomasi, C.: Edge, Junction, and Corner Detection Using Color Distributions. PAMI (2001)
26. Wang, J., Li, J., Wiederhold, G.: SIMPLIcity: Semantics-Sensitive Integrated Matching for Picture Libraries. PAMI (2001)
27. Sharma, G., Wu, W., Dalal, E.: The CIEDE2000 color-difference formula: implementation notes, supplementary test data, and mathematical observations. CRA (2005)
28. Ling, H.: Articulated shape benchmark and idsc code (2010),
<http://www.ist.temple.edu/~hbling/code/inner-dist-articu-distribution.zip>
29. Guisewite, G., Pardalos, P.: Minimum concave-cost network flow problems: Applications, complexity, and algorithms. AOR (1990)
30. Amiri, A., Pirkul, H.: New formulation and relaxation to solve a concave-cost network flow problem. JORS (1997)
31. Xing, E.P., Ng, A.Y., Jordan, M.I., Russell, S.: Distance metric learning with application to clustering with side-information. In: NIPS (2003)
32. Bar-Hillel, A., Hertz, T., Shental, N., Weinshall, D.: Learning distance functions using equivalence relations. In: ICML (2003)
33. Goldberger, J., Roweis, S., Hinton, G., Salakhutdinov, R.: Neighbourhood components analysis. In: NIPS (2005)
34. Globerson, A., Roweis, S.: Metric learning by collapsing classes. In: NIPS (2006)
35. Yang, L., Jin, R.: Distance metric learning: A comprehensive survey. MSU (2006)
36. Davis, J., Kulis, B., Jain, P., Sra, S., Dhillon, I.: Information-theoretic metric learning. In: ICML (2007)
37. Yu, J., Amores, J., Sebe, N., Radeva, P., Tian, Q.: Distance learning for similarity estimation. PAMI (2008)
38. Weinberger, K., Saul, L.: Distance metric learning for large margin nearest neighbor classification. JMLR (2009)
39. Assent, I., Wichterich, M., Seidl, T.: Adaptable Distance Functions for Similarity-based Multimedia Retrieval. DSN (2006)

Chapter 5

Interpolated-Discretized Metric Learning using Linear Programming

This chapter presents results which have not been
published yet

Interpolated-Discretized Metric Learning using Linear Programming

Abstract

Non-Mahalanobis distances such as χ^2 and robust estimators increase accuracy in many machine learning and computer vision applications. However, the distances are usually hand picked and cannot be learned efficiently from data. In this paper, we present a new non-Mahalanobis distance family, the *Interpolated-Discretized* (ID) distance family, and show how to efficiently learn its parameters. Once an ID distance is learned, computation of the distance is linear in the dimension and does not depend on the number of training examples (unlike kernel methods in which computing a distance scales linearly in the number of training examples). Many metric learning methods can be used with our ID distances. We demonstrate one possible way of learning ID distances using the Large Margin Nearest Neighbor (LMNN) framework [1]. With ID distances, the LMNN optimization problem is a linear program. Finally, we show that our method often outperforms other state-of-the-art metric learning approaches.

1 Introduction

Distance learning is of fundamental importance in machine learning. Learned distances can be used to improve algorithms which rely on distance computations such as nearest neighbor classification [1, 2], energy-based models [1, 3], supervised kernel machines (such as GPs or SVMs) and even unsupervised clustering algorithms [4].

Most of the work on metric learning [1, 2, 4, 5, 6, 7, 8, 9, 10, 11] has concentrated on learning a Mahalanobis distance, which is equivalent to learning a linear transformation of the data and then using the standard Euclidean distance. Let $\vec{x}^i, \vec{x}^j$ be two vectors, the squared Mahalanobis distance between them is parameterized by a positive-semidefinite (PSD) matrix, M :

$$\text{Mahalanobis}(\vec{x}^i, \vec{x}^j) = (\vec{x}^i - \vec{x}^j)^T M (\vec{x}^i - \vec{x}^j) \quad (1)$$

The PSD constraint on M is required for obtaining non-negative distances. Also, it implies that the M matrix can be expressed as $M = LL^T$ for some real matrix L . Thus, the distance can be computed as the squared Euclidean norm between linearly transformed vectors:

$$\text{Mahalanobis}(\vec{x}^i, \vec{x}^j) = \|L\vec{x}^i - L\vec{x}^j\|_2^2 \quad (2)$$

Note that, as defined above, the square root of a Mahalanobis measure is a pseudo-metric, as the distance between two different vectors might be zero. Computing the distance between never-seen examples using Mahalanobis distances scales quadratically in the dimension. The *Interpolated-Discretized* (ID) distance we introduce in the current work scales linearly in the dimension with just a small constant memory overhead.

Although much of the metric-learning work has focused on the Mahalanobis distance, this measure has some inherent limitations. The most important one is that it is a function only of the difference between input vectors $\vec{x}^i - \vec{x}^j$. Thus, it will give the same distance whenever two vectors have the same difference vector. For example, a Mahalanobis distance will not be able to differentiate between

($\vec{x}^i = 42, \vec{x}^j = 40$) and ($\vec{x}^i = 2, \vec{x}^j = 0$). To overcome this limitation several metric learning methods have been proposed [1, 3, 7, 9, 12]. We discuss these methods in section 4.

Another approach to overcoming the above limitation is to apply a function directly to $\vec{x}^i, \vec{x}^j$ such that the distance is not invariant to $\vec{x}^i - \vec{x}^j$. One particularly successful instance of this approach is the χ^2 distance, which has recently received considerable attention in the computer vision literature. It has been used in state-of-the-art: contour detection and segmentation algorithms [13], descriptors matching [14, 15], shape classification [15, 16] and image retrieval [15] to name a few. Another related approach is robust estimators which were successfully used optical flow computation [17]. Although successful, χ^2 and robust estimators are *hard-coded* and cannot be naturally learned or improved using data.

Another related shortcoming of the Mahalanobis distance is that it is invariant to where $\vec{x}^i$ and $\vec{x}^j$ are in space, as long as their difference vector is the same. This is undesirable in many cases, as we would often like to normalize the distances by the range of the input vectors, e.g., as in the χ^2 metric. In other words, the Mahalanobis distance is limited in terms of the bin to bin (or feature to feature) relationships it can capture.

In this work, we propose the *Interpolated-Discretized* (ID) distance. Our measure can overcome some of the key inherent limitations of the Mahalanobis distance, and is also more flexible than other approaches such as χ^2 . Specifically, an ID distance can model different distances over different parts of the space.

Our approach contains two key elements. The first is a discretization of each feature into C values such that the interaction between any two features is represented by an arbitrary $C \times C$ matrix. This allows us to model arbitrary relations between the two features. However, such a coarsely quantized representation neglects relevant information in the original value, and is thus undesirable. To overcome this, we employ a careful interpolation approach, which does take into account the original feature values.

The resulting ID measure is continuous in feature space but highly expressive due to its different treatment of different feature regimes. Before going into details, we provide an illustrative example in Fig. 1. The figure shows how the χ^2 measure can be well approximated via ID with only $C = 10$ clusters per feature. Note that in general, the goal is not to approximate a known distance such as χ^2 , but to learn to approximate the best *unknown* distance. A nice property of ID, that can also be seen in Fig. 1 is that for $C = 2$ it exactly equals the l_1 norm. This again highlights the fact that ID is continuous, but becomes more and more detailed as C is increased.

In the current work, we do not represent cross-bin dependencies. It is possible to extend our method such that ID distances will also be able to model relationships between features, so it will be able to learn both Mahalanobis type of distances and bin-to-bin distances and variants in between. This can be done by discretizing pairs of values. However, this will increase the computational complexity considerably. We decided to concentrate on bin-to-bin ID distance since it is interesting to check

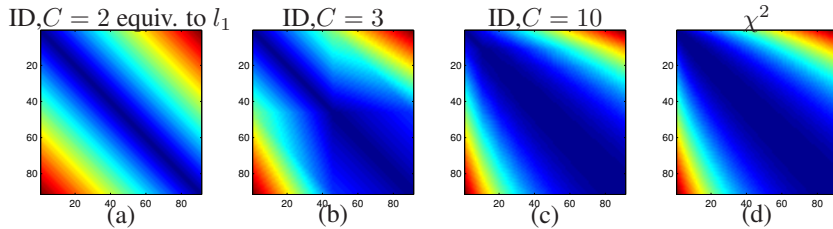


Figure 1: An illustration of a 1-dimensional χ^2 distance approximation using ID distances with different numbers of discretization points. Panel (d) shows the original χ^2 distance. In (a) we see that using just two points of discretization (i.e., $C = 2$ corresponding to the range $[0, 90]$), the ID distance in this case is equal to the l_1 norm. In (b), using three points of discretization $[0, 45, 90]$ (i.e., $C = 3$), the ID distance starts to resemble χ^2 , although the difference is noticeable. In (c), using $C = 10$ corresponding to the range $[0, 10, \dots, 90]$, the ID distance is almost identical to χ^2 . We emphasize that the goal is not to approximate a known distance such as χ^2 , but to learn to approximate the best *unknown* distance.

what can be achieved with those, especially since other works already showed that the diagonal Mahalanobis Metric Learning achieves competitive results [2, 18] and they are very fast to compute.

A useful property of an ID distance is that it is a linear combination of its parameters. Thus, many of the proposed metric learning optimization problems are especially suited for use with an ID distance. We demonstrate learning ID distance parameters using the Large Margin Nearest Neighbor (LMNN) framework [1]. LMNN with an ID distance is a linear program.

We evaluate our method on a variety of standard datasets and show that it often outperforms state-of-the-art metric learning approaches. We thus believe that ID distances are very useful distances which can be learned effectively from data.

2 The Interpolated-Discretized Distance Family

We begin by describing the ID distance between two scalars x^i, x^j . The vector case will easily follow. An ID distance between two scalar features is parameterized via a vector and a matrix. The first parameter is a sorted vector of C cluster centers of possible values of the feature: $[v_1, \dots, v_C]$. The second parameter is a matrix $M \in \mathbb{R}^{C \times C}$, where $M_{a,b}$ corresponds to the distance between v_a and v_b . See Fig. 2.

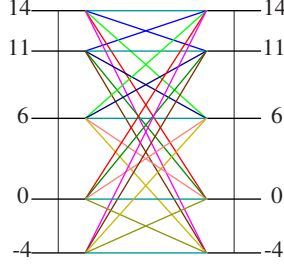


Figure 2: An Illustration of 1-dimensional ID distance parameters. Feature values are clustered into $C = 5$ cluster centers: $[-4, 0, 6, 11, 14]$. The full 5×5 matrix of distances between pairs of cluster centers values is illustrated as edges.

However, we would not like the distance to have the same value when x^i, x^j belong to a given pair of clusters. Rather, we would like the measure to smoothly vary with the feature values. One natural way to achieve this is via interpolation. Given two scalars, the two closest cluster centers are found for each. Next, four coefficients are computed. Each coefficient corresponds to one pair of cluster centers and is a normalized triangle area. See Fig. 3 for description and illustration of how the coefficients are computed. Finally, the ID distance is the sum of the coefficients multiplied by their corresponding entries from the matrix of the distances between pairs of cluster centers values:

$$\begin{aligned}
 \text{ID}(x^i, x^j) &= \sum_{t=1}^4 \alpha_{a(t), b(t)} M_{a(t), b(t)} \\
 a(1) &= a(2) = \underset{c}{\operatorname{argmax}} \{v_c \leq x^i\} \\
 a(3) &= a(4) = \underset{c}{\operatorname{argmin}} \{v_c \geq x^i\} \\
 b(1) &= b(3) = \underset{c}{\operatorname{argmax}} \{v_c \leq x^j\} \\
 b(2) &= b(4) = \underset{c}{\operatorname{argmin}} \{v_c \geq x^j\}
 \end{aligned} \tag{3}$$

In the above the coefficients $\alpha_{a(t), b(t)}$ correspond to the interpolation coefficient. See Fig. 3.

We now discuss properties of the ID distance. There are three conditions for a distance function, D , to be a semimetric. The first is *non-negativity* (i.e. $D(\vec{x}^i, \vec{x}^j) \geq 0$). The second is *identity* (i.e. $D(\vec{x}^i, \vec{x}^j) = 0$ iff $\vec{x}^i = \vec{x}^j$). The third is *symmetry* (i.e. $D(\vec{x}^i, \vec{x}^j) = D(\vec{x}^j, \vec{x}^i)$). A natural question is what properties does the M matrix need to satisfy in order for the resulting ID measure to be a

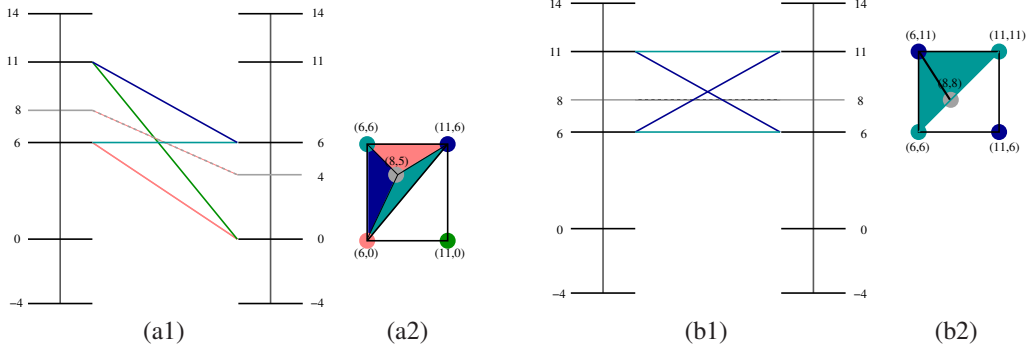


Figure 3: An illustration of how the coefficients are computed for $x^i = 8, x^j = 4$ where the cluster centers are: $[-4, 0, 6, 11, 14]$ (as in Fig. 2). In (a1), the closest cluster centers are $(6, 11)$ for $x^i = 8$ and $(0, 6)$ for $x^j = 4$. Thus, the four discrete distances are: $M_{3,2}, M_{3,3}, M_{4,2}$ and $M_{4,3}$ which correspond to the four cluster center pairs: $(6, 0), (6, 6), (11, 0)$ and $(11, 6)$. To compute the α coefficients (the interpolation step), we check in (a2) whether the point $(x^i = 8, x^j = 4)$ is above or below the line connecting $(6, 0)$ with $(11, 6)$. Then, the normalized triangle areas are the coefficients of the corresponding discrete distances. Note that since the point was above the line, the coefficient of $M_{4,2}$ is zero.

In (b1), (b2) we see a degenerate case where the interpolation is between $M_{3,3}$ and $M_{4,4}$. If we enforced the identity property on M , these will be zero and the resulting distance is also zero.

This interpolation has several advantages. First, the resulting distance will have the identity property if we enforce the distance between two identical cluster centers to be zero. Second, using this interpolation results in a distance which is always continuous and differentiable except at the discretization points. This means, for example, that an ID distance is not very sensitive to the number of clusters chosen. Finally, an ID distance is equivalent to a weighted l_1 norm if the number of cluster centers is two and the distance between identical cluster centers is set to zero.

semimetric. It's straightforward to see that the conditions on M are simply that it satisfies $M_{i,j} > 0$ for all $i \neq j$, as well as $M_{i,i} = 0$ for all i and $M_{i,j} = M_{j,i}$ for all i, j . All these can be easily enforced via linear constraints on the elements of M .

A semimetric is also a metric if it has the property of *subadditivity* (i.e. $D(\vec{x}^i, \vec{x}^j) \leq D(\vec{x}^i, \vec{x}^k) + D(\vec{x}^k, \vec{x}^j)$). We can transform a semimetric into a non-continuous metric by linearly mapping all strictly bigger than zero distances to the range $[1, 2]$. Thus it is very easy to transform an ID distance into a metric. An additional discussion about triangle inequality can be found in Jacobs et al. [19].

To generalize to N features we define a separate matrix, M^n for each feature.

3 Interpolated-Discretized Large Margin Nearest Neighbor Metric Learning

In this section we show how to learn an ID distance using the Large Margin Nearest Neighbor Metric Learning framework [1]. We start by describing the LMNN formulation.

Let $\{(\vec{x}^i, y^i)\}_{i=1}^T$ denote a set of T labeled examples with inputs $\vec{x}^i \in \mathbb{R}^N$ and discrete (but not necessarily binary) class label y^i . We use the binary matrix $y_{ij} \in \{0, 1\}$ to indicate whether or not the labels y^i and y^j match.

In addition to the class label y_i , for each input $\vec{x}^i$ we also specify k "target" neighbors, that is, k other inputs with the same label y^i that we wish to have minimal distance to $\vec{x}^i$. In the absence of any knowledge, the target neighbors can be the k closest nearest neighbors using the Euclidean distance. This is also what we use in this paper. We denote that $\vec{x}^j$ is a target neighbor of $\vec{x}^i$ as $j \rightsquigarrow i$.

The LMNN framework has a loss function with two terms. The first, tries to minimize the distance between all $\vec{x}^i$ and their corresponding target neighbors. The second, tries to push vectors with different labels away, such that the distance to them minus the distance to the target neighbor (the margin) will be bigger or equal to 1. Since this might be unfeasible, slack variables are added and

the second term is the sum of slack variables. The two terms are weighted with a parameter μ . Using ID distances the optimization of LMNN is:

$$\begin{aligned}
\min_{M, \xi} \quad & (1 - \mu) \sum_{i, j \rightsquigarrow i} \text{ID}(\vec{x}^i, \vec{x}^j) + \mu \sum_{i, j \rightsquigarrow i, l} (1 - y_{il}) \xi_{ijl} \quad \text{subject to:} \\
& \forall i, j \rightsquigarrow i, l \quad \text{ID}(\vec{x}^i, \vec{x}^l) - \text{ID}(\vec{x}^i, \vec{x}^j) \geq 1 - \xi_{ijl} \\
& \forall i, j \rightsquigarrow i, l \quad \xi_{ijl} \geq 0 \\
& \text{ID}(\vec{x}^i, \vec{x}^j) = \sum_{n=1}^N \sum_{t=1}^4 \alpha_{a(t), b(t)}^n M_{a(t), b(t)}^n \\
& \forall n, a, b \quad M_{a, b}^n \geq 0
\end{aligned} \tag{4}$$

This linear program allows to learn any bin-to-bin distance function. In order to learn a semi-metric, the following constraints should be added:

$$\begin{aligned}
& \forall n, a \quad M_{a, a}^n = 0 \\
& \forall n, a \neq b \quad M_{a, b}^n \geq \varepsilon \\
& \forall n, a, b \quad M_{a, b}^n = M_{b, a}^n
\end{aligned} \tag{5}$$

where ε is some strictly positive small scalar. We use $\varepsilon = 10^{-5}$. We can also use an upper bound on M^n entries as a regularizer. We used one as an upper bound.

As we discussed before, we can transform a semimetric into a non-continuous metric by linearly mapping all strictly bigger than zero distances to the range $[1, 2]$. Thus it is very easy to transform an ID distance into a metric.

Finally, we can add monotone constraints as a regularizer:

$$\forall n, a < b < c, \quad M_{a, b}^n \leq M_{a, c}^n \quad M_{b, c}^n \leq M_{a, c}^n \tag{6}$$

4 Related Work

Our method builds in a novel direction on the success of previous metric learning approaches. As Weinberger and Saul [1] conjectured, more adaptive transformations of the input space can lead to improved performance. Our method allows to enlarge the number of the learned parameters, while the computation of the distance between two never-seen examples is only linear in the dimension.

Chopra et al. [3] proposed to learn a convolutional neural net as a non-linear transformation before applying the ℓ_2 norm. They showed excellent results on image data. Babenko et al. [12] suggested a boosting framework for learning non-Mahalanobis metrics. They also presented excellent results on image data. These methods are non-convex and thus they might suffer from local minimas and training is sensitive to parameters.

Kernel methods were also proposed in order to learn a Mahalanobis distance over non-linear transformations of the data [1, 7, 9]. Computing such a distance between two vectors scales linear in the number of training examples, which makes it impractical for large datasets. Computing our ID distances does not depend on the number of training examples.

A family of non-Mahalanobis distances recently proposed is the *Quadratic-Chi* (QC) [15]. The QC family generalizes both the Mahalanobis distance and the χ^2 distance. A QC distance have parameters that can be learned. However, a serious limitation is that it can only model χ^2 -like distances. In addition, it is applicable only to non-negative vectors. Finally, it is non-convex with respect to its parameters, so learning them is hard.

Rosales and Fung [5] also propose learning metrics via linear programming. However, while we learn a non-Mahalanobis distance, their method learns a subfamily of Mahalanobis distance. That is, their method is restricted to learning a Mahalanobis distance which is parameterized with a diagonal dominant matrix.

5 Results

We evaluated our approach on ten UCI data sets of varying size and difficulty. Table 1 summarizes the data sets properties. Except for Spectf all experimental results are averaged over several runs of randomly generated stratified 70/30 splits of the data. The Spectf data set has a predefined training/test split. All experiments are averaged over 100 runs, except for the Steel database where it is averaged over 20 runs and for the Cardio dataset where it is averaged over 5 runs.

	Iris	Wine	Ion	Sonar	Transf	Spectf	Breast	Glass	Steel	Cardio
examples	150	178	351	208	748	267	106	214	1941	2126
classes	3	3	2	2	2	2	6	6	7	10
dimension	4	13	34	60	9	44	9	9	27	21

Table 1: Properties of data sets

We applied kNN classifiers with different distances. We compared using our learned ID distances to using Mahalanobis distance learned with ITML [9] and with the original LMNN [1].

All methods were used with target neighbors $k = 3$. ITML was used with the default parameters in the distribution of the code. That is, the slack variable parameter γ which inversely correspond to the regularization magnitude was tuned using cross-validation over the values $\{0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000, 10000\}$. The LMNN and ID-LMNN μ parameter was set as 0.5 (the default in LMNN code). Our ID-LMNN approach was tested with two different numbers of clusters C . First, $C = 2$ which is equivalent to learning weighted l_1 norm using the LMNN framework. Second, $C = 10$ which allows to capture the non-linearities in the data.

We used Mosek [20] as the LP solver, except for Steel and Cardio where we used a subgradient solver as the problems do not fit in memory. We are working on accelerating the subgradient method using active set method as in LMNN and using stochastic subgradient methods. We believe that it will allow us to work on much larger datasets.

The results are presented in Fig. 4. There are several observations. First, for almost all databases, metric learning is better than naive Euclidean distance. Second, learned ID distance with $C = 2$, which is equivalent to a learned weighted l_1 distance, has an excellent performance. Learned diagonal Mahalanobis distances (equivalent to weighted l_2) is known to have competitive results [2, 18]. In our experiments, learned weighted l_1 outperforms the full Mahalanobis distances in many cases.

Finally, learning our proposed ID distances with $C = 10$ is a strong competitor on all datasets and outperformed other methods on most of the datasets, especially the larger datasets and those with a large number of classes.

6 Conclusions

We presented a new distance family - the *Interpolated-Discretized* (ID) distances. ID distances have many desirable properties. First, they can approximate any bin-to-bin distance. Second, as they are linear combination of parameters, they can be learned efficiently. Third, using linear constraints we can enforce them to be a metric without needing more complex semi-definite requirements on the parameters as in Mahalanobis distances. Finally, we have shown that experimentally learned ID distances have excellent performance and they often outperform state-of-the-art learned distances.

We note that on data with a large quantization problems, such as image pixels values, our method is likely not to perform well. However, modern computer vision methods do not use image pixels values as a feature, but use state-of-the-art descriptors such as SIFT [21] which do not suffer too much from quantization problems.

Future work will include the generalization of ID distance to model “cross-bin” relationships. This can be done by discretizing pairs of values. However, this will be much more time demanding and this issue need to be resolved.

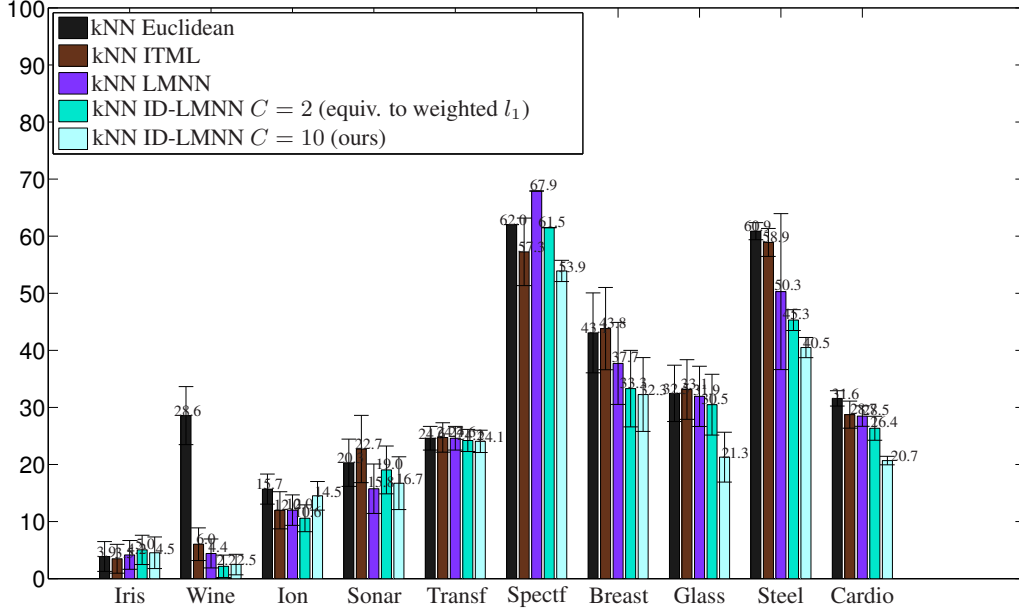


Figure 4: Test errors with standard deviations for kNN classification using several distances. In our experiments, learned weighted l_1 outperforms the full Mahalanobis distances in many cases. Our proposed ID distances with $C = 10$ is a strong competitor on all datasets and outperformed other methods on most of the datasets, especially the larger datasets and those with a large number of classes.

References

- [1] K.Q. Weinberger and L.K. Saul. Distance metric learning for large margin nearest neighbor classification. *JMLR*, 2009.
- [2] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighbourhood components analysis. *NIPS*, 2005.
- [3] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, 2005.
- [4] E.P. Xing, A.Y. Ng, M.I. Jordan, and S. Russell. Distance metric learning with application to clustering with side-information. *NIPS*, 2003.
- [5] R. Rosales and G. Fung. Learning sparse metrics via linear programming. In *ICKDM*, 2006.
- [6] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning distance functions using equivalence relations. In *ICML*, 2003.
- [7] A. Globerson and S. Roweis. Metric learning by collapsing classes. *NIPS*, 2006.
- [8] L. Yang and R. Jin. Distance metric learning: A comprehensive survey. *MSU*, 2006.
- [9] J.V. Davis, B. Kulis, P. Jain, S. Sra, and I.S. Dhillon. Information-theoretic metric learning. In *ICML*, 2007.
- [10] D. Ramanan and S. Baker. Local distance functions: A taxonomy, new algorithms, and an evaluation. *PAMI*, 2010.
- [11] B. McFee and G.R.G. Lanckriet. Learning multi-modal similarity. *JMLR*, 2011.
- [12] B. Babenko, S. Branson, and S. Belongie. Similarity metrics for categorization: from monolithic to category specific. In *ICCV*, 2009.
- [13] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *PAMI*, 2010.
- [14] P.E. Forssén and D.G. Lowe. Shape Descriptors for Maximally Stable Extremal Regions. In *ICCV*, 2007.
- [15] O. Pele and M. Werman. The quadratic-chi histogram distance family. In *ECCV*, 2010.

- [16] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *PAMI*, 2002.
- [17] A. Bab-Hadiashar and D. Suter. Optic flow calculation using robust statistics. In *CVPR*, 1997.
- [18] M.P. Kumar, PHS Torr, and A. Zisserman. An invariant large margin nearest neighbour classifier. In *ICCV*, 2007.
- [19] D.W. Jacobs, D. Weinshall, and Y. Gdalyahu. Classification with nonmetric distances: Image retrieval and class representation. *PAMI*, 2000.
- [20] The mosek optimization software. <http://www.mosek.com/>.
- [21] D.G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.

Chapter 6

Conclusions

We presented several new distance functions and efficient algorithms for their computation. The distance functions were designed to be robust to common image noise and deformation. On the other hand, the distance functions were designed to be as distinctive as possible. The distances which balance between robustness and distinctiveness improved on the state-of-the-art of several computer vision applications such as image retrieval and shape classification. Improved results using our distance functions were also reported by other researchers, both from the computer vision community and outside of it (*e.g.* [54–57]).

Our first conjecture was that real-world noise has two main components. First, limited deformation noise. Second, not limited outlier noise. For example, deformation noise can change blue to some other shade of blue; On the other hand, outlier noise (*e.g.* occlusion) can change blue to any other color. We proposed using saturated cross-bin distances, that is, distances where different outliers receive the same value.

We also presented $\widehat{EMD}$, a new generalization of the transportation distance for non-normalized histograms. Unlike the classic generalization (EMD), $\widehat{EMD}$ is a metric even when the histograms are non-normalized. Additionally, $\widehat{EMD}$ is often better than EMD in practice. We have shown that these distances outperformed state-of-the-art distance functions in many computer vision applications.

A second observation was that image histograms have a unique statistical behavior since large values usually indicates large variance. χ^2 was proposed as a distance function that overcomes this problem [8, 9]. χ^2 is the second order Taylor expansion of the Jensen Shannon (JS) divergence [7] and thus χ^2 and JS results are almost identical [8–10]. We suggested a new family of histogram distances, called *Quadratic-Chi* (QC). Members of this family are robust to quantization effects and have a χ^2 -like normaliza-

tion. We also proposed two new cross-bin distances properties *Similarity-Matrix-Quantization-Invariance* and *Sparseness-Invariance* and a proof that $\widehat{EMD}$ and QC distances have these properties. We demonstrated excellent experimental results, also showing the practical importance of the two new properties.

Finally, we described a novel method for non-Mahalanobis metric learning, *Interpolated-Discretized Metric Learning*. This was motivated by the success of our non-Mahalanobis distance, the QC. A QC distance has parameters that can be learned. However, a serious limitation is that it can only model χ^2 -like distances. In addition, it is applicable only to non-negative vectors. Finally, it is non-convex with respect to its parameters, so learning them is hard. To overcome these problems we suggested a new non-Mahalanobis distance family, the *Interpolated-Discretized* (ID) distance family, and showed how to efficiently learn its parameters. An ID distance can approximate any *unknown* distance. Many metric learning methods can be used with our ID distances. We demonstrated one possible way of learning ID distances using the Large Margin Nearest Neighbor (LMNN) framework [2]. With ID distances, the LMNN optimization problem is a linear program. Finally, we showed that our method often outperforms other state-of-the-art metric learning approaches.

Future work

The work described in this thesis can be extended in several directions:

- Several algorithmic improvements are planned for the algorithm presented in chapter 3.
- The algorithms speed can be further improved using techniques such as Bayesian sequential hypothesis testing [58], metric trees [59–61] and approximate nearest neighbor [62, 63].
- An interesting question is whether we can change the Earth Mover’s Distance so that it will also reduce the effect of large values. Concave-cost network flow [64] seems to be the right direction for future work although it presents two major obstacles. First, the concave-cost network flow optimization is NP hard [64]. However, there are available approximations [64, 65]. Second, simply using concave-cost flow networks will result in a distance which is not *Similarity-Matrix-Quantization-Invariant*.

- Generalization of the *Interpolated-Discretized* distance to model cross-bin relationships. This can be done by discretizing pairs of values. However, this will be much more time demanding and this issue need to be resolved.

Bibliography

- [1] Y. Rubner, C. Tomasi, and L. J. Guibas, “The earth mover’s distance as a metric for image retrieval,” *IJCV*, 2000.
- [2] K. Weinberger and L. Saul, “Distance metric learning for large margin nearest neighbor classification,” *JMLR*, 2009.
- [3] J. Lin, “Divergence measures based on the shannon entropy,” *TIT*, 1991.
- [4] F. Ôsterreicher and I. Vajda, “A new class of metric divergences on probability spaces and its statistical applications,” *AIMS*, 2003.
- [5] D. M. Endres and S. J.E., “A new metric for probability distributions,” *IT*, 2003.
- [6] F. Topsøe, “Jenson-shannon divergence and norm-based measures of discrimination and variation,” *Preprint*, 2003.
- [7] F. Topsoe, “Some inequalities for information divergence and related measures of discrimination,” *IT*, 2000.
- [8] J. Puzicha, T. Hofmann, and J. Buhmann, “Non-parametric similarity measures for unsupervised texture segmentation and image retrieval,” in *CVPR*, 1997.
- [9] Y. Rubner, J. Puzicha, C. Tomasi, and J. Buhmann, “Empirical evaluation of dissimilarity measures for color and texture,” *CVIU*, 2001.
- [10] O. Pele and M. Werman, “The quadratic-chi histogram distance family,” in *ECCV*, 2010.
- [11] G. Snedecor and W. Cochran, “Statistical Methods, ed 6. Ames, Iowa,” 1967.

- [12] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, “Contour detection and hierarchical image segmentation,” *PAMI*, 2010.
- [13] P. Forssén and D. Lowe, “Shape Descriptors for Maximally Stable Extremal Regions,” in *ICCV*, 2007.
- [14] O. Cula and K. Dana, “3D texture recognition using bidirectional feature histograms,” *IJCV*, 2004.
- [15] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid, “Local features and kernels for classification of texture and object categories: A comprehensive study,” *IJCV*, 2007.
- [16] M. Varma and A. Zisserman, “A statistical approach to material classification using image patch exemplars,” *PAMI*, 2009.
- [17] A. Vedaldi and A. Zisserman, “Efficient additive kernels via explicit feature maps,” in *CVPR*, 2010.
- [18] D. Xu, T. Cham, S. Yan, L. Duan, and S. Chang, “Near Duplicate Identification with Spatially Aligned Pyramid Matching,” *CSVT*, 2010.
- [19] S. Belongie, J. Malik, and J. Puzicha, “Shape matching and object recognition using shape contexts,” *PAMI*, 2002.
- [20] H. Ling and D. Jacobs, “Shape classification using the inner-distance,” *PAMI*, 2007.
- [21] J. Hafner, H. Sawhney, W. Equitz, M. Flickner, and W. Niblack, “Efficient color histogram indexing for quadratic form distance functions,” *PAMI*, 1995.
- [22] P. Mahalanobis, “On the generalized distance in statistics,” in *PNIS*, 1936.
- [23] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov, “Neighbourhood components analysis,” *NIPS*, 2005.
- [24] E. Xing, A. Ng, M. Jordan, and S. Russell, “Distance metric learning with application to clustering with side-information,” *NIPS*, 2003.
- [25] R. Rosales and G. Fung, “Learning sparse metrics via linear programming,” in *ICKDM*, 2006.

- [26] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall, “Learning distance functions using equivalence relations,” in *ICML*, 2003.
- [27] A. Globerson and S. Roweis, “Metric learning by collapsing classes,” *NIPS*, 2006.
- [28] L. Yang and R. Jin, “Distance metric learning: A comprehensive survey,” *MSU*, 2006.
- [29] J. Davis, B. Kulis, P. Jain, S. Sra, and I. Dhillon, “Information-theoretic metric learning,” in *ICML*, 2007.
- [30] D. Ramanan and S. Baker, “Local distance functions: A taxonomy, new algorithms, and an evaluation,” *PAMI*, 2010.
- [31] B. McFee and G. Lanckriet, “Learning multi-modal similarity,” *JMLR*, 2011.
- [32] S. Chopra, R. Hadsell, and Y. LeCun, “Learning a similarity metric discriminatively, with application to face verification,” in *CVPR*, 2005.
- [33] B. Babenko, S. Branson, and S. Belongie, “Similarity metrics for categorization: from monolithic to category specific,” in *ICCV*, 2009.
- [34] G. Monge, “Déblai et remblai,” *Mémoires de l’Académie des Sciences*, 1781.
- [35] L. Kantorovitch, “On the translocation of masses,” *MS*, 1958.
- [36] L. Wasserstein, “Markov processes over denumerable products of spaces, describing large systems of automata,” *PIT*, 1969.
- [37] C. Mallows, “A note on asymptotic joint normality,” *AMS*, 1972.
- [38] H. Shen and A. Wong, “Generalized texture representation and metric,” *CVGIP*, 1983.
- [39] M. Werman, S. Peleg, and A. Rosenfeld, “A distance metric for multidimensional histograms.,” *CVGIP*, 1985.
- [40] M. Werman, S. Peleg, R. Melter, and T. Kong, “Bipartite graph matching for points on a line or a circle,” *JOA*, 1986.

- [41] S. Peleg, M. Werman, and H. Rom, “A unified approach to the change of resolution: Space and gray-level,” *PAMI*, 1989.
- [42] J. Orlin, “A faster strongly polynomial minimum cost flow algorithm,” in *STOC*, 1988.
- [43] H. Ling and K. Okada, “An Efficient Earth Mover’s Distance Algorithm for Robust Histogram Comparison,” *PAMI*, 2007.
- [44] J. Gudmundsson, O. Klein, C. Knauer, and M. Smid, “Small Manhattan Networks and Algorithmic Applications for the Earth Mover’s Distance,” in *EWCG*, 2007.
- [45] P. Indyk and N. Thaper, “Fast image retrieval via embeddings,” in *IWSCTV*, 2003.
- [46] K. Grauman and T. Darrell, “The pyramid match kernel: Efficient learning with sets of features,” *JMLR*, 2007.
- [47] S. Shirdhonkar and D. Jacobs, “Approximate earth mover’s distance in linear time,” in *CVPR*, 2008.
- [48] S. Khot and A. Naor, “Nonembeddability theorems via Fourier analysis,” *Mathematische Annalen*, 2006.
- [49] A. Andoni, P. Indyk, and R. Krauthgamer, “Earth mover distance over high-dimensional spaces,” in *SODA*, 2008.
- [50] K. Grauman and T. Darrell, “Fast contour matching using approximate earth mover’s distance,” in *CVPR*, 2004.
- [51] T. Goh, R. West, and K. Okada, “Robust detection of semantically equivalent visually dissimilar objects,” in *SLAM*, 2008.
- [52] O. Pele and M. Werman, “A linear time histogram metric for improved sift matching,” in *ECCV*, 2008.
- [53] O. Pele and M. Werman, “Fast and robust earth mover’s distances,” in *ICCV*, 2009.
- [54] R. Gupta, H. Patil, and A. Mittal, “Robust order-based methods for feature description,” in *CVPR*, 2010.
- [55] M. Rubinstein, D. Gutierrez, O. Sorkine, and A. Shamir, “A comparative study of image retargeting,” in *SIGGRAPH Asia*, 2010.

- [56] O. Macindoe and W. Richards, “Graph comparison using fine structure analysis,” in *ICSC*, 2010.
- [57] T. Xu, H. Wang, and L. Sun, “Wood recognition based on histogram matching algorithms,” *Applied Mechanics and Materials*, vol. 58, pp. 1744–1748, 2011.
- [58] O. Pele and M. Werman, “Robust real time pattern matching using bayesian sequential hypothesis testing,” *PAMI*, 2008.
- [59] P. Yianilos, “Data structures and algorithms for nearest neighbor search in general metric spaces,” in *SODA*, 1993.
- [60] P. Ciaccia, M. Patella, and P. Zezula, “M-tree: An Efficient Access Method for Similarity Search in Metric Spaces,” in *ICVLDB*, 1997.
- [61] A. Beygelzimer, S. Kakade, and J. Langford, “Cover trees for nearest neighbor,” in *ICML*, 2006.
- [62] S. Arya, D. Mount, N. Netanyahu, R. Silverman, and A. Wu, “An optimal algorithm for approximate nearest neighbor searching fixed dimensions,” *JACM*, 1998.
- [63] M. Muja and D. Lowe, “Fast approximate nearest neighbors with automatic algorithm configuration,” in *VISSAPP*, 2009.
- [64] G. Guisewite and P. Pardalos, “Minimum concave-cost network flow problems: Applications, complexity, and algorithms,” *AOR*, 1990.
- [65] A. Amiri and H. Pirkul, “New formulation and relaxation to solve a concave-cost network flow problem,” *JORS*, 1997.

פונקציות מרחק: תיאוריה, אלגוריתמים ואפליקציות

חיבור לשם קבלת תואר
"דוקטור לפילוסופיה"

מאת
אופיר פלא

הוגש לסינט של האוניברסיטה העברית
יולי 2011

עבודה זו נעשתה בהדרכתו של
פרופ' מיכאל ורמן

תקציר

אלגוריתמים רבים בתחום הראייה הממוחשבת ובתחום הלמידה החישובית כול-לים בתוכם חישוב מרחקים. לדוגמה, שערך תנועת מצלמה מתמונה אחת לאחרת, כולל שלב של התאמת מתארים (מתארים הם וקטורים של מספרים המתאר-ים איזור מסויים בתמונה). השלב הראשון הוא בחירת איזורים קטנים בשתי התמונות. השלב השני הוא חישוב מתאר עבור כל אחד מהאיזורים הללו. על מנת להחליט אם שני מתארים (משתי תמונות שונות) הם למעשה תמונה של אותו האיזור התלת מימדי בעולם, המתארים מושווים בעזרת פונקציה המחשבת מרחק ביניהם. בחירת פונקציה המרחק משפיעה על דיוק ומהירות השיטה. שימוש אחר לפונקציות מרחק הוא חיפוש תמונות דומות. המרחק יקבע את איכות התמונות הדומות שיוחזרו למשתמש וכמה זמן ייקח להחזיר אותן. בתחום הל-מידה חישובית, איכות תיוג על פי k השכנים הקרובים ביותר נקבעת על ידי פונקציה המרחק המשמשת לקביעת השכנים.

תיזה זו מציגה מספר פונקציות מרחק חדשות וחוקרת את תכונותיהן. דגש מיוחד מושם על ישימות מעשית המבוססת על תובנות תיאורטיות. התיזה מציגה אלגוריתמים יעילים לחישוב פונקציות המרחק. פונקציות המרחק המוצעות אדישות לרעש בתמונות, מצד שני, הן בעלות ערכים גבוהים עבור שתי מתארים שונים (לדוגמא שתי מתארים שלא מתארים את אותו האיזור בעולם). חוקרים אחרים השתמשו בשיטות המוצעות בתיזה זו על מנת לפתור בעיות מראייה ממוחשבת ומתחומים אחרים וקיבלו תוצאות טובות יותר משיטות אחרות. הצלחת השי-טות בתחומים אחרים נובעת כנראה מכך שתכונות הרעש בתחומים הללו דומות לתכונות הרעש של תמונות.

החלק הראשון והחלק השני של תיזה זו מוקדשים ל- Earth Mover's Distance (EMD) [1]. מרחק זה הוא הכללה של מרחק ה- transportation להיסטוגרמות לא מנורמלות. אנו מציגים הכללה חדשה למרחק ה- transportation להיסטוגרמות לא מנורמלות, $\widehat{EMD}$. שלא כמו ה- EMD הקלאסי, $\widehat{EMD}$ הוא מטריקה אפילו עבור היסטוגרמות לא מנורמלות. בנוסף לכך, שימוש ב- $\widehat{EMD}$ נותן בדרך כלל תוצאות ניסיוניות טובות יותר משימוש ב- EMD. אנחנו גם מראים כי $\widehat{EMD}$ הינו הכללה טבעית של מרחק ה- L_1 .

מתארים שהם היסטוגרמות של זווית משמשות למספר רב מאד של משימות ראייה ממוחשבת. רב המרחקים שמשמשים להשוואת שתי מתארים כאלו לא לוקחים בחשבון את המבנה המעגלי. לעומת זאת חישוב המרחקים שכן לוקחים את המבנה המעגלי בחשבון הוא ארוך מדי. אנו מציגים אלגוריתם לינארי המשווה בין היסטוגרמות של זווית ולוקח בחשבון את המבנה המעגלי. שיטה זו משיגה תוצאות טובות יותר מהשיטות הקודמות על בוחן ביצועים שהשתמשו בו רבים וזמין לכולם.

בחלק השני של התיזה אנו מציגים משפחה של Earth Mover's Distances אדישה לרעש וניתן להשתמש בה לכל סוג של היסטוגרמות (לא רק היסטוגרמות של זווית). פיתחנו אלגוריתם המחשב את המרחקים הללו בסדר גודל מהר יותר

מאשר חישוב מרחקים דומים. שיטה זו משיגה תוצאות מצויינות בחיפוש תמונות דומות.

בחלק השלישי של התיזה אנו מציגים משפחה חדשה של מרחקים להיסטו-גרמות – Quadratic-Chi (QC). מרחקי QC הם מרחקי תבנית ריבועית עם נרמול דומה לנרמול של מרחק ה- χ^2 . נרמול זה מוריד את השפעת התאים עם הערכים הגבוהים ונורמליזציה זו הוכחה כמועילה במקרים רבים. עם זאת, χ^2 רגיש לקוונ-טיזציה (שנגרמת לדוגמא על ידי שינויי תאורה ועיוותי צורה). הפרמטרים של התבנית הריבועית ממדלים את הקשר בין התאים השונים של ההיסטוגרמה (לדוגמא אדום וכתום) וכך בעיית הקוונטיזציה משוככת. אנו מציגים שתי תכונות חדשות למרחקים הממדלים קשר בין תאי ההיסטוגרמה: *Similarity-Matrix-Quantization-Invariance* ו-*Sparseness-Invariance* ומראים כי לכל מרחקי ה-QC יש את התכונות הללו. אנו מראים גם כי תכונות אילו חשובות מבחינה מעשית. חישוב מרחק QC נעשה בזמן לינארי. כמו כן קל למקבל את חישוב המרחק. מרחקי QC משיגים תוצאות טובות יותר ממה שהיו פונקציות המרחק הטובות ביותר עד כה וזמן חישובם קצר.

בפרק הרביעי של התיזה אנו מתארים שיטה חדשנית למידת מרחקים שקראנו לה - *"Interpolated-Discretized Metric Learning"*. הצלחת מרחקי ה-QC נתנה מוטיבציה לחלק זה של העבודה. אמנם למרחקי QC ישנם פרמטרים שניתן ללמוד אותם, אך מגבלה חמורה היא שהמרחקים הללו מתאימים רק לוקטורים לא שליליים. בנוסף, QC לא קמור ביחס לפרמטרים שלו ולכן קשה ללמוד אותם. על מנת להתגבר על בעיות אלו אנו מציעים משפחה חדשה של מרחקים לה קראנו - *"Interpolated-Discretized"* או בקיצור ID. אנו מראים כיצד ניתן ללמוד את הפרמטרים של מרחק כזה בצורה יעילה. מרחק ID יכול לקרב כל מרחק. קל לשנות את רוב שיטות למידת המרחקים הקיימות כיום כך שילמדו מרחקי ID. אנו מדגימים דרך אחת אפשרית ללימוד מרחקי ID על ידי שימוש בשיטה Large Margin Nearest Neighbor [2]. עם מרחקי ID בעיית האופטימיזציה של שיטה זו היא בעיית תכנון לינארי. אנו מראים כי במקרים רבים שיטת למידת המרחקים החדשה שלנו משיגה תוצאות טובות יותר משיטות למידת מרחקים שהיו השיטות הטובות ביותר עד כה.

Distance Functions: Theory, Algorithms and Applications - Appendices

Thesis submitted for the degree of
“Doctor of Philosophy”

By
Ofir Pele

Submitted to the Senate of the Hebrew University
July, 2011

This work was carried out under the supervision of:
Prof. Michael Werman

Contents

1	A Linear Time Histogram Metric for Improved SIFT Matching - Appendices	1
2	Fast and Robust Earth Mover's Distances - Appendices	47
3	The Quadratic-Chi Histogram Distance Family - Appendices	68

Chapter 1

A Linear Time Histogram Metric for Improved SIFT Matching - Appendices

Appendix A. Linear Time EMD_{MOD} Algorithm

This appendix presents a linear time algorithm for the computation of the Modulo Earth Mover's Distance between normalized cyclic histograms. The algorithm is an adaption of Werman et al. algorithm for bipartite graph matching for points on a circle [22].

Let $A = \{0, \dots, N-1\}$ be N points, equally spaced, on a circle. Let Q and P be two normalized histograms of such points. The Modulo Earth Mover's Distance (EMD_{MOD}) between Q and P is defined as:

$$\text{EMD}_{\text{MOD}}(Q, P) = \min_{F=\{f_{ij}\}} \sum_{i,j} f_{ij} D_{\text{MOD}}(i, j) \quad s.t \quad (6)$$

$$\sum_j f_{ij} \leq P_i, \quad \sum_i f_{ij} \leq Q_j, \quad \sum_{i,j} f_{ij} = \sum_i P_i = \sum_j Q_j, \quad f_{ij} \geq 0 \quad (7)$$

where $D_{\text{MOD}}(i, j)$ is the modulo L_1 distance:

$$D_{\text{MOD}}(i, j) = \min(|i - j|, N - |i - j|) \quad (8)$$

The linear time EMD_{MOD} pseudo code is given in Alg. 1. Lines 2-8 compute the F function from [22] for masses instead of points; *i.e.* $F[i]$ is the total mass in bins smaller or equal to i in Q , minus the total mass in bins smaller or equal to i in P . As was noted by Cabrelli and Molter [37] the cutting point of the circle for the match can be found with a weighted median algorithm when the points on the circles are sorted. In histograms the points are sorted and equally spaced; thus it is enough to find the median. Lines 10-16 in Alg. 1 cut the circles into two lines. Finally, lines 17-31 match groups of points instead of single points.

Algorithm 1 $\text{EMD}_{\text{MOD}}(Q, P, N)$

```

1:  $D \leftarrow 0$ 
2:  $cQ \leftarrow Q[0]$ 
3:  $cP \leftarrow P[0]$ 
4:  $F[0] \leftarrow Q[0] - P[0]$ 
5: for  $i = 1$  to  $N - 1$  do
6:    $cQ \leftarrow cQ + Q[i]$ 
7:    $cP \leftarrow cP + P[i]$ 
8:    $F[i] \leftarrow cQ - cP$ 
9:  $i \leftarrow ((\text{index of the median of } F) + 1) \bmod N$ 
10: for  $t = 0$  to  $N - 1$  do
11:    $I[t] \leftarrow i$ 
12:    $i \leftarrow (i + 1) \bmod N$ 
13:  $tQ \leftarrow 0$ 
14:  $tP \leftarrow 0$ 
15:  $iQ \leftarrow I[tQ]$ 
16:  $iP \leftarrow I[tP]$ 
17: while true do
18:   while  $Q[iQ] = 0$  do
19:      $tQ \leftarrow tQ + 1$ 
20:     if  $tQ = N$  then
21:       return  $D$ 
22:    $iQ \leftarrow I[tQ]$ 
23:   while  $P[iP] = 0$  do
24:      $tP \leftarrow tP + 1$ 
25:     if  $tP = N$  then
26:       return  $D$ 
27:    $iP \leftarrow I[tP]$ 
28:    $f \leftarrow \min(Q[iQ], P[iP])$ 
29:    $Q[iQ] \leftarrow Q[iQ] - f$ 
30:    $P[iP] \leftarrow P[iP] - f$ 
31:    $D \leftarrow f \times \min(|iQ - iP|, N - |iQ - iP|)$ 

```

Appendix B. $\widehat{EMD}$ Metric Proof

In this appendix we prove that if $\alpha \geq 0.5$ and the ground distance is a metric, $\widehat{EMD}$ (see Eq. 3 in page 3) is a metric. Non-negativity and symmetry hold trivially in all cases, so we only need to prove that the triangle inequality holds.

Theorem 1. *Let A, B, C be three vectors in $(\mathbb{R}^+)^N$ and let $\widehat{EMD}$ be with a metric ground distance and $\alpha \geq 0.5$, then:*

$$\widehat{EMD}_\alpha(A, B) + \widehat{EMD}_\alpha(B, C) \geq \widehat{EMD}_\alpha(A, C) \quad (9)$$

Proof. Given an $N \times N$ distance matrix d_{ij} , let $\tilde{d}_{ij}$ be an $(N+1) \times (N+1)$ distance matrix such that:

$$\tilde{d}_{ij} = \begin{cases} d_{ij} & 1 \leq i, j \leq N \\ \alpha \max_{i,j} \{d_{ij}\} & i = N+1, j \neq N+1 \\ \alpha \max_{i,j} \{d_{ij}\} & i \neq N+1, j = N+1 \\ 0 & i = N+1, j = N+1 \end{cases} \quad (10)$$

Let $S = \max(\sum_{i=1}^N A_i, \sum_{i=1}^N B_i, \sum_{i=1}^N C_i)$. We define three vectors in $\mathbb{R}^{N+1}$:

$$\tilde{A} = [A, S - \sum_{i=1}^N A_i] \quad (11)$$

$$\tilde{B} = [B, S - \sum_{i=1}^N B_i] \quad (12)$$

$$\tilde{C} = [C, S - \sum_{i=1}^N C_i] \quad (13)$$

The sum of each of the three vectors equals S . In addition:

$$\widehat{EMD}_\alpha(A, B) = EMD(\tilde{A}, \tilde{B}) \times S \quad (14)$$

$$\widehat{EMD}_\alpha(B, C) = EMD(\tilde{B}, \tilde{C}) \times S \quad (15)$$

$$\widehat{EMD}_\alpha(A, C) = EMD(\tilde{A}, \tilde{C}) \times S \quad (16)$$

where $\widehat{EMD}$ is with the ground distance d_{ij} and EMD is with the ground distance $\tilde{d}_{ij}$. Rubner et al. [1] proved that the triangle inequality holds for EMD when the ground distance is a metric and the histograms are normalized. $\tilde{A}, \tilde{B}, \tilde{C}$ are normalized histograms. If the ground distance d_{ij} is a metric and $\alpha \geq 0.5$, the ground distance $\tilde{d}_{ij}$ is a metric (an enumeration of all cases proves this). Thus we get:

$$EMD(\tilde{A}, \tilde{B}) + EMD(\tilde{B}, \tilde{C}) \geq EMD(\tilde{A}, \tilde{C}) \quad (17)$$

$$\Downarrow (S \geq 0) \quad (18)$$

$$(EMD(\tilde{A}, \tilde{B}) \times S) + (EMD(\tilde{B}, \tilde{C}) \times S) \geq (EMD(\tilde{A}, \tilde{C}) \times S) \quad (19)$$

$$\Downarrow (\text{Eqs. 14, 15, 16}) \quad (20)$$

$$\widehat{EMD}_\alpha(A, B) + \widehat{EMD}_\alpha(B, C) \geq \widehat{EMD}_\alpha(A, C) \quad (21)$$

Appendix C. A Linear Time Histogram Metric for Improved SIFT Matching Additional Results

Ofir Pele¹ and Michael Werman¹

School of Computer Science and Engineering
The Hebrew University of Jerusalem
{ofirpele,werman}@cs.huji.ac.il

This document contains all the 1-precision vs. recall graph results for the experiments in the paper: “A Linear Time Histogram Metric for Improved SIFT Matching” [1].

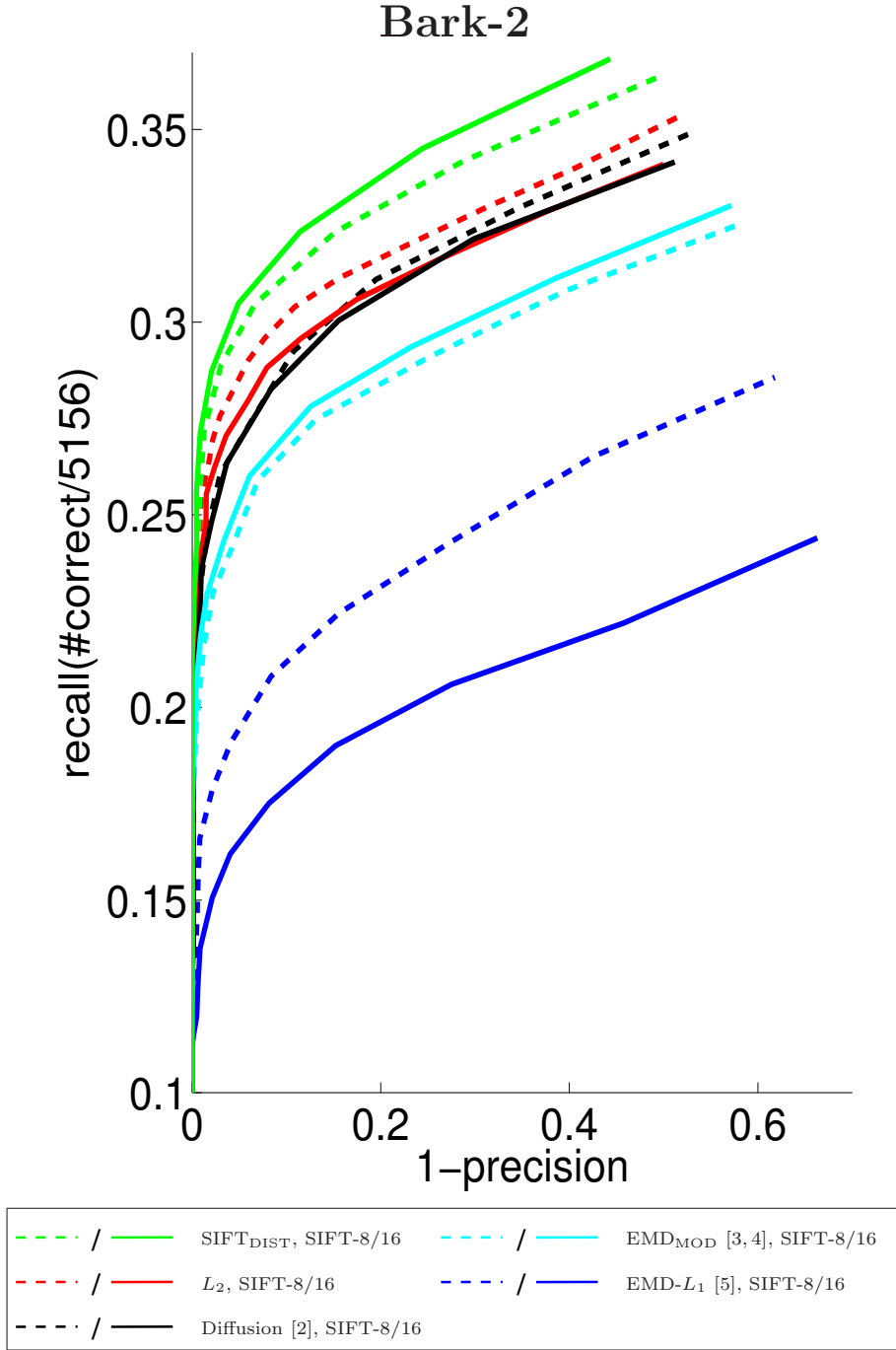


Fig. 1. Results on the Mikolajczyk and Schmid dataset [6]. Should be viewed in color.

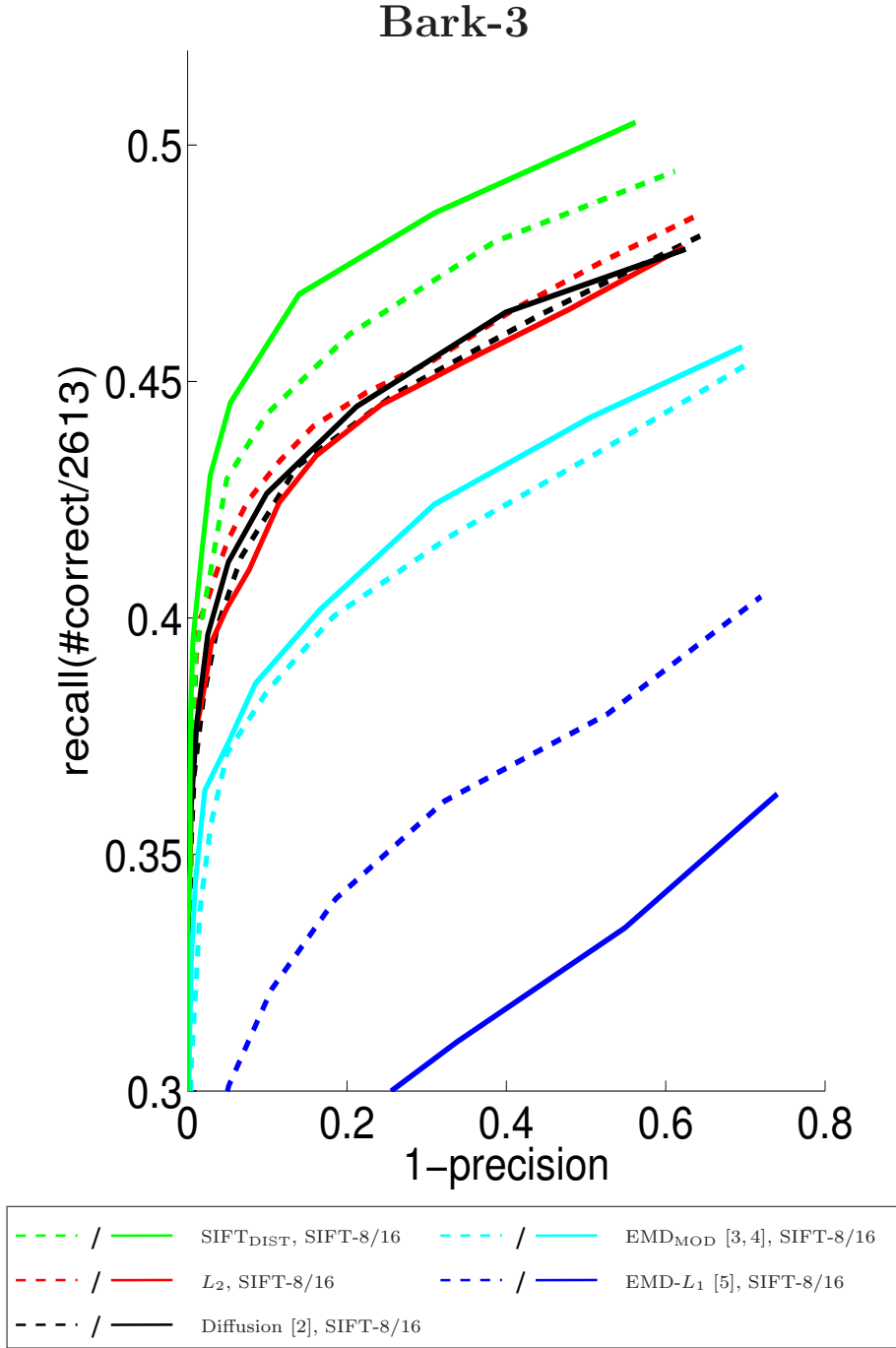


Fig. 2. Results on the Mikolajczyk and Schmid dataset [6]. Should be viewed in color.

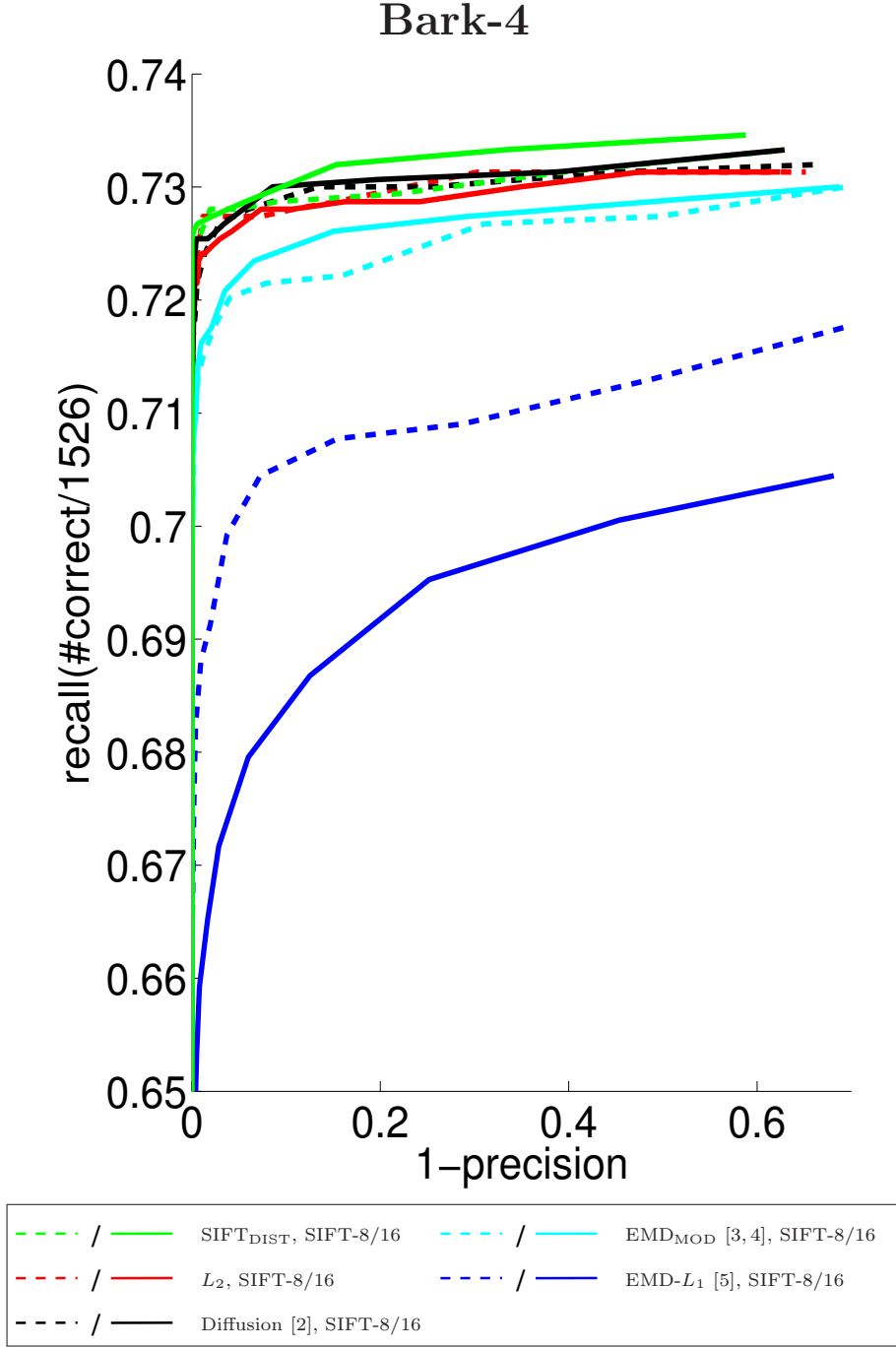


Fig. 3. Results on the Mikolajczyk and Schmid dataset [6]. Should be viewed in color.

Bark-5

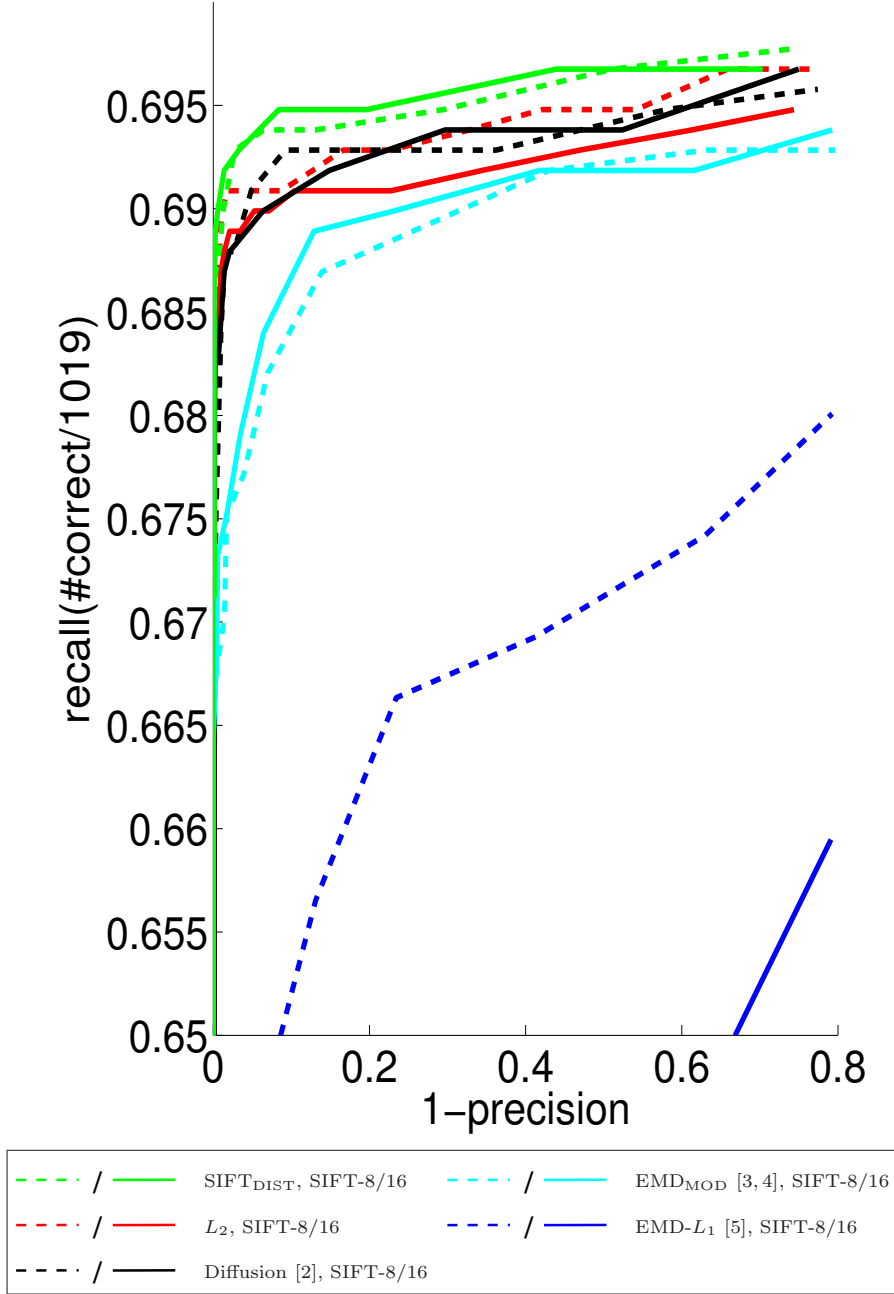


Fig. 4. Results on the Mikolajczyk and Schmid dataset [6]. Should be viewed in color.

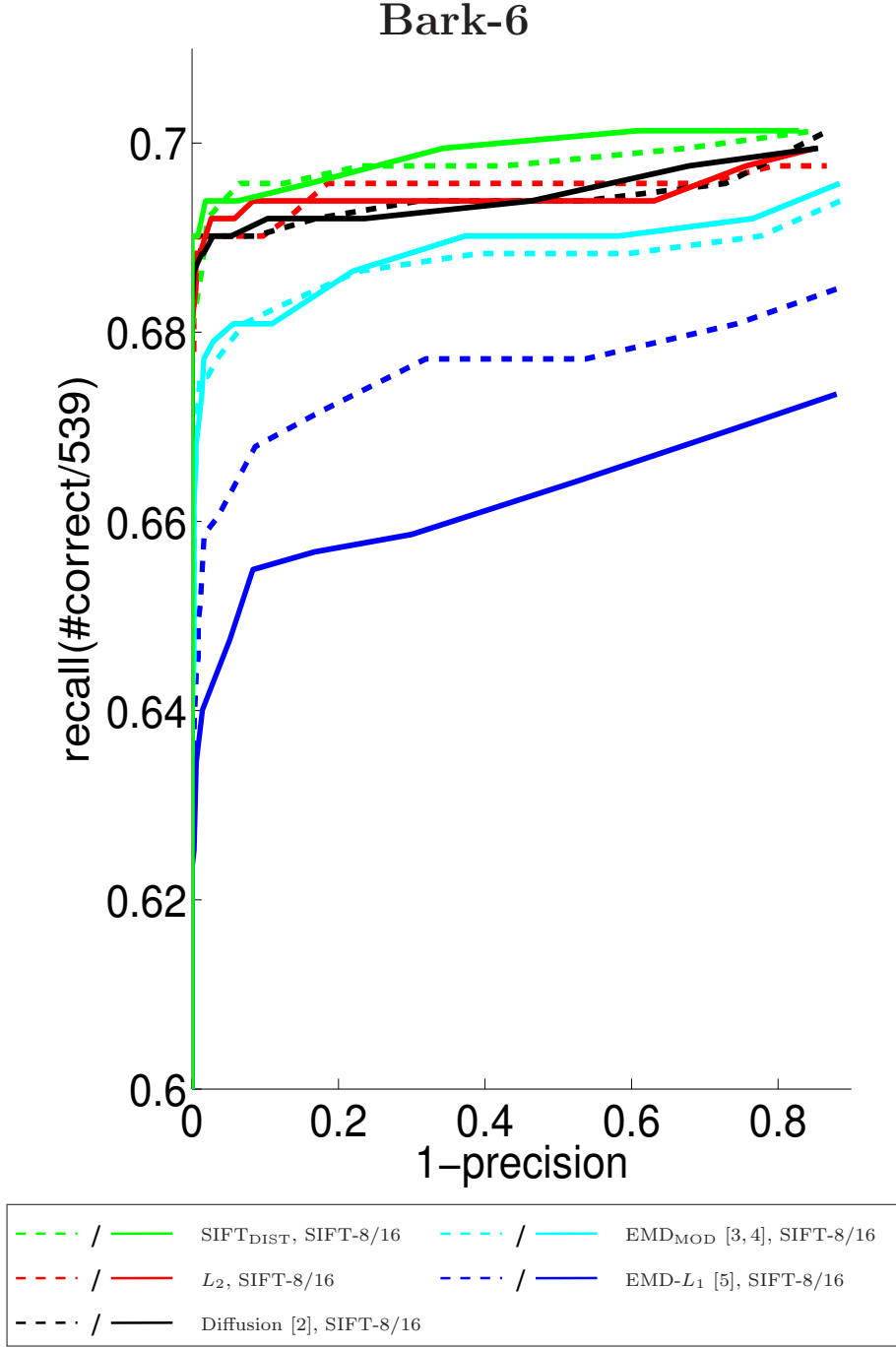


Fig. 5. Results on the Mikolajczyk and Schmid dataset [6]. Should be viewed in color.

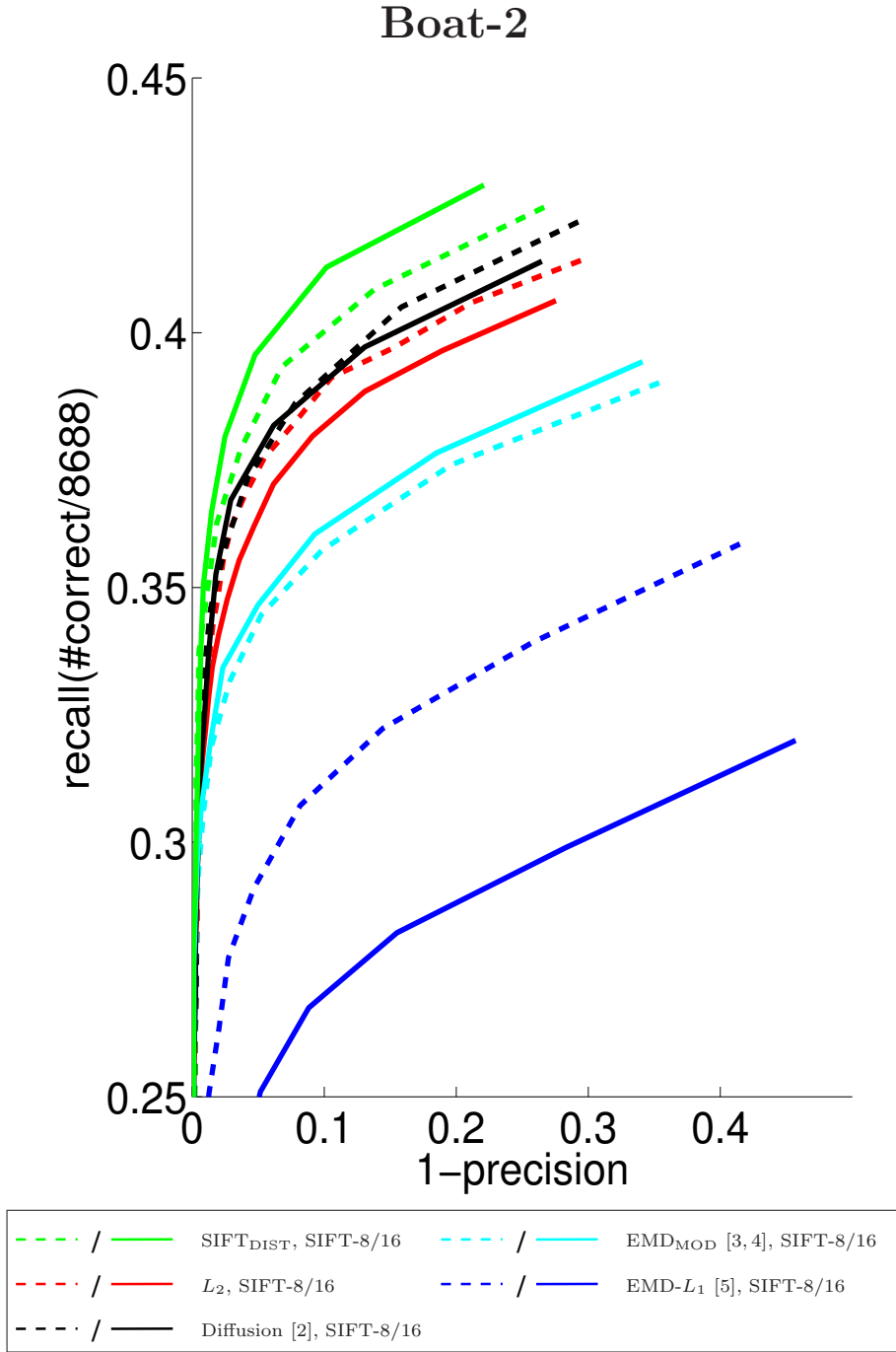


Fig. 6. Results on the Mikolajczyk and Schmid dataset [6]. Should be viewed in color.

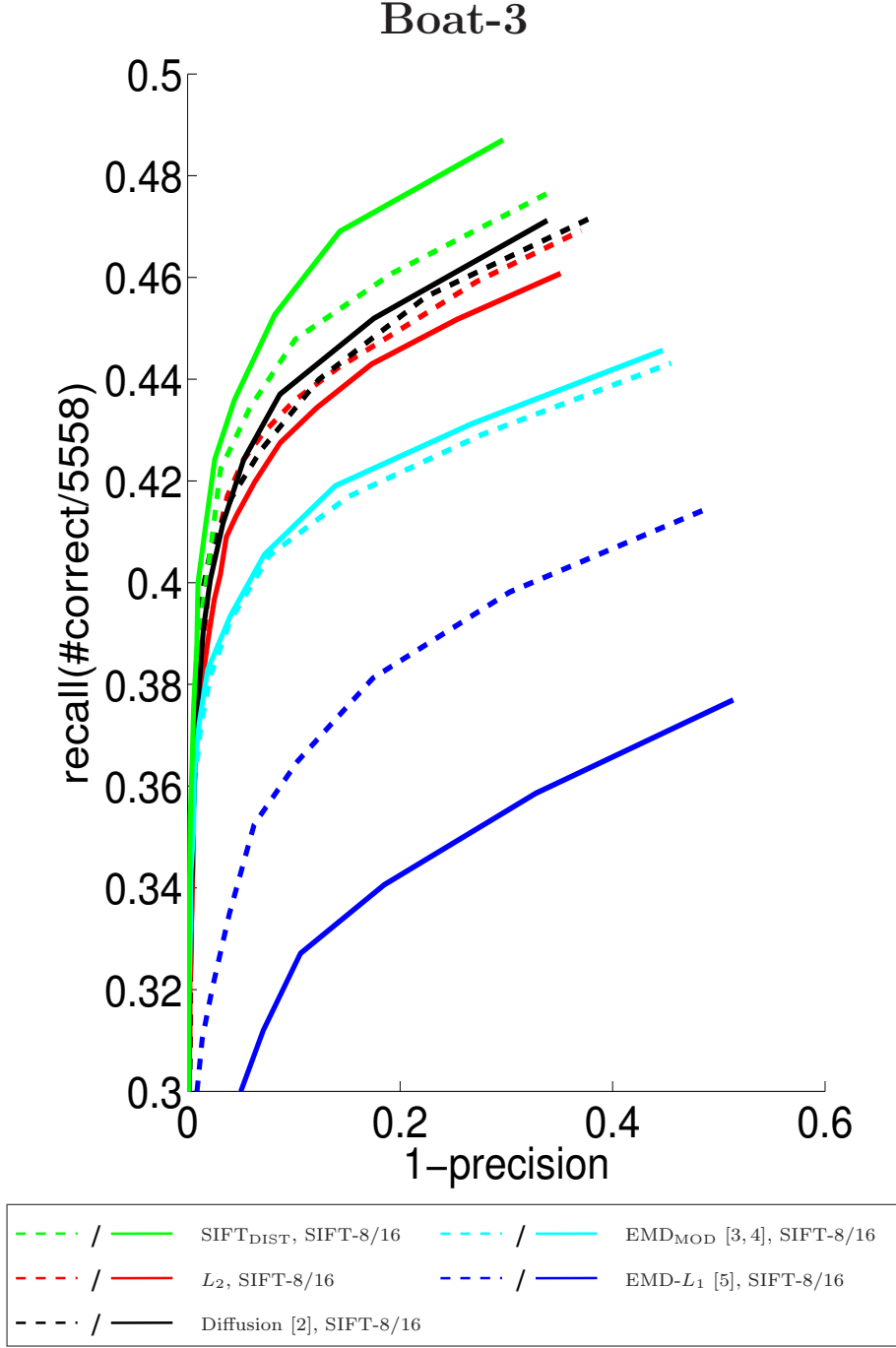


Fig. 7. Results on the Mikolajczyk and Schmid dataset [6]. Should be viewed in color.

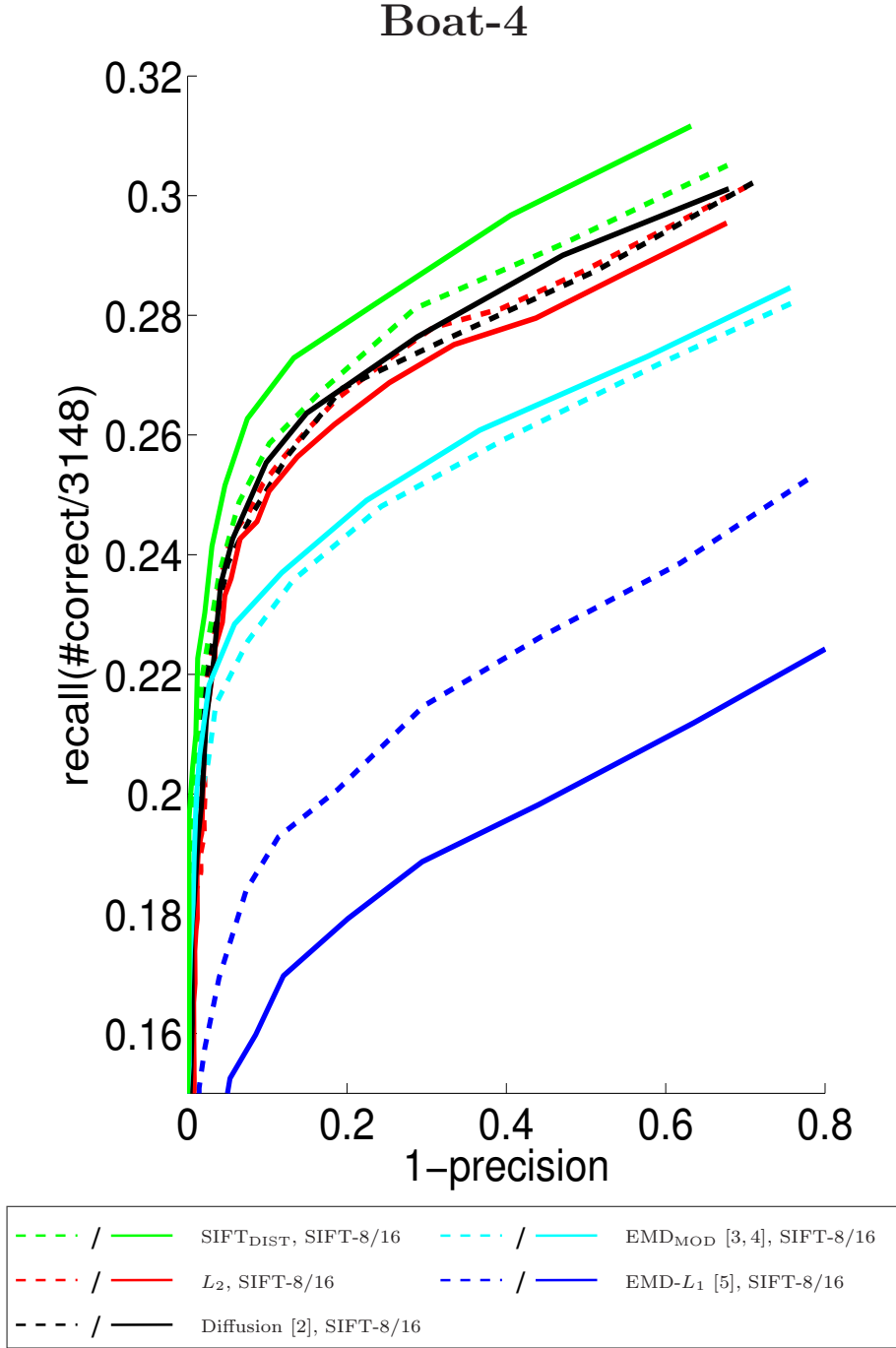


Fig. 8. Results on the Mikolajczyk and Schmid dataset [6]. Should be viewed in color.

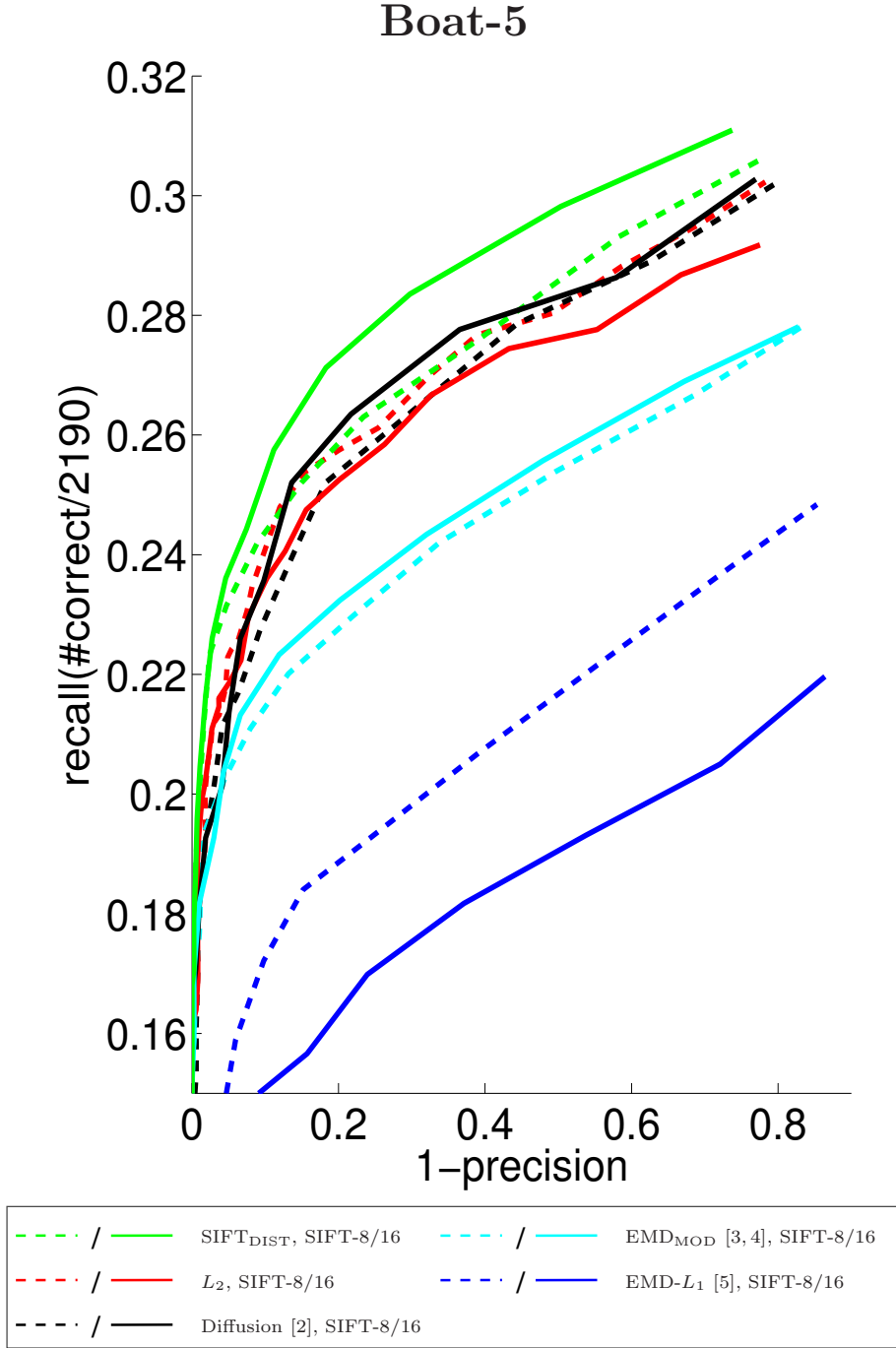


Fig. 9. Results on the Mikolajczyk and Schmid dataset [6]. Should be viewed in color.

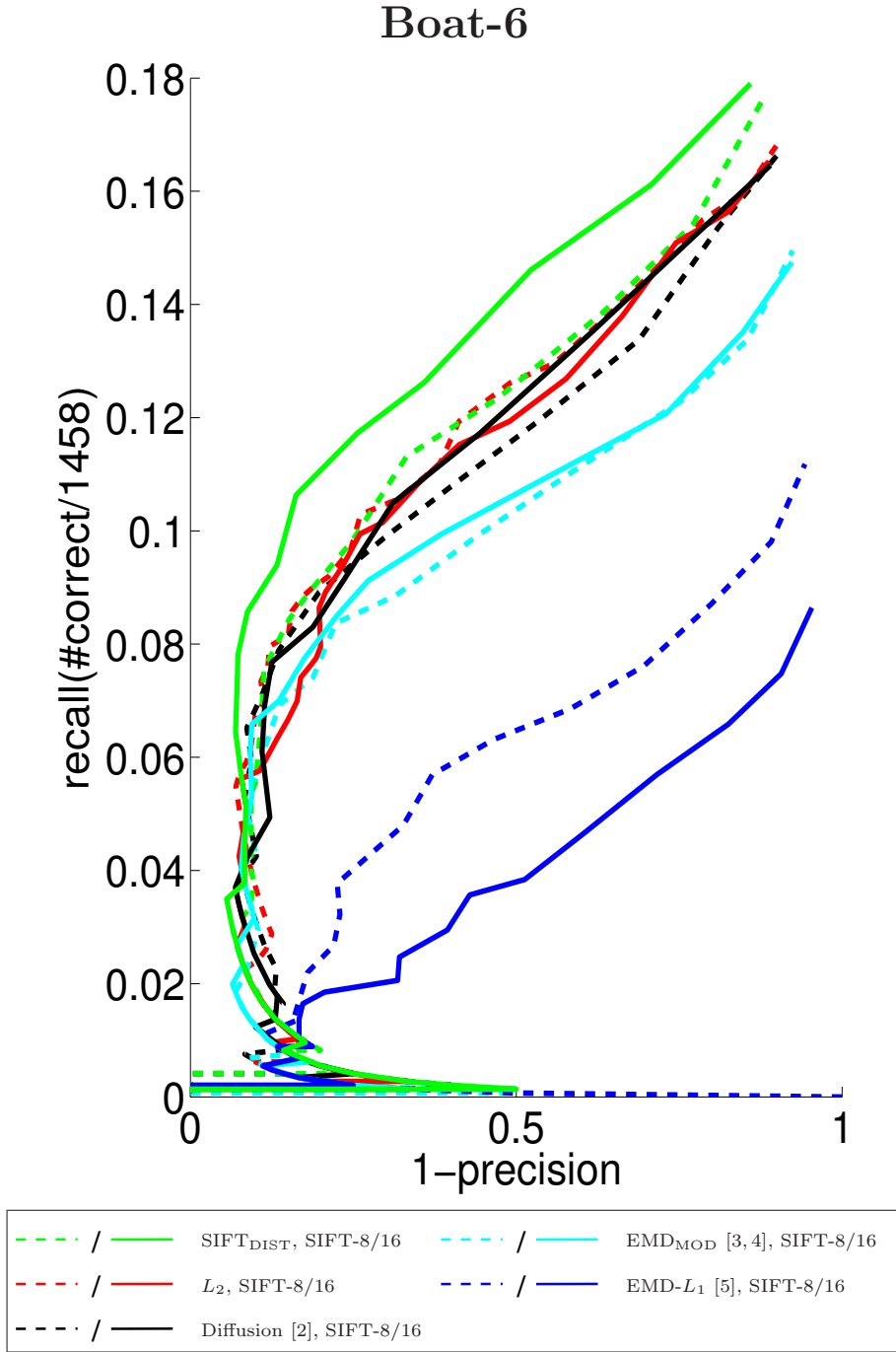


Fig. 10. Results on the Mikolajczyk and Schmid dataset [6]. Should be viewed in color.

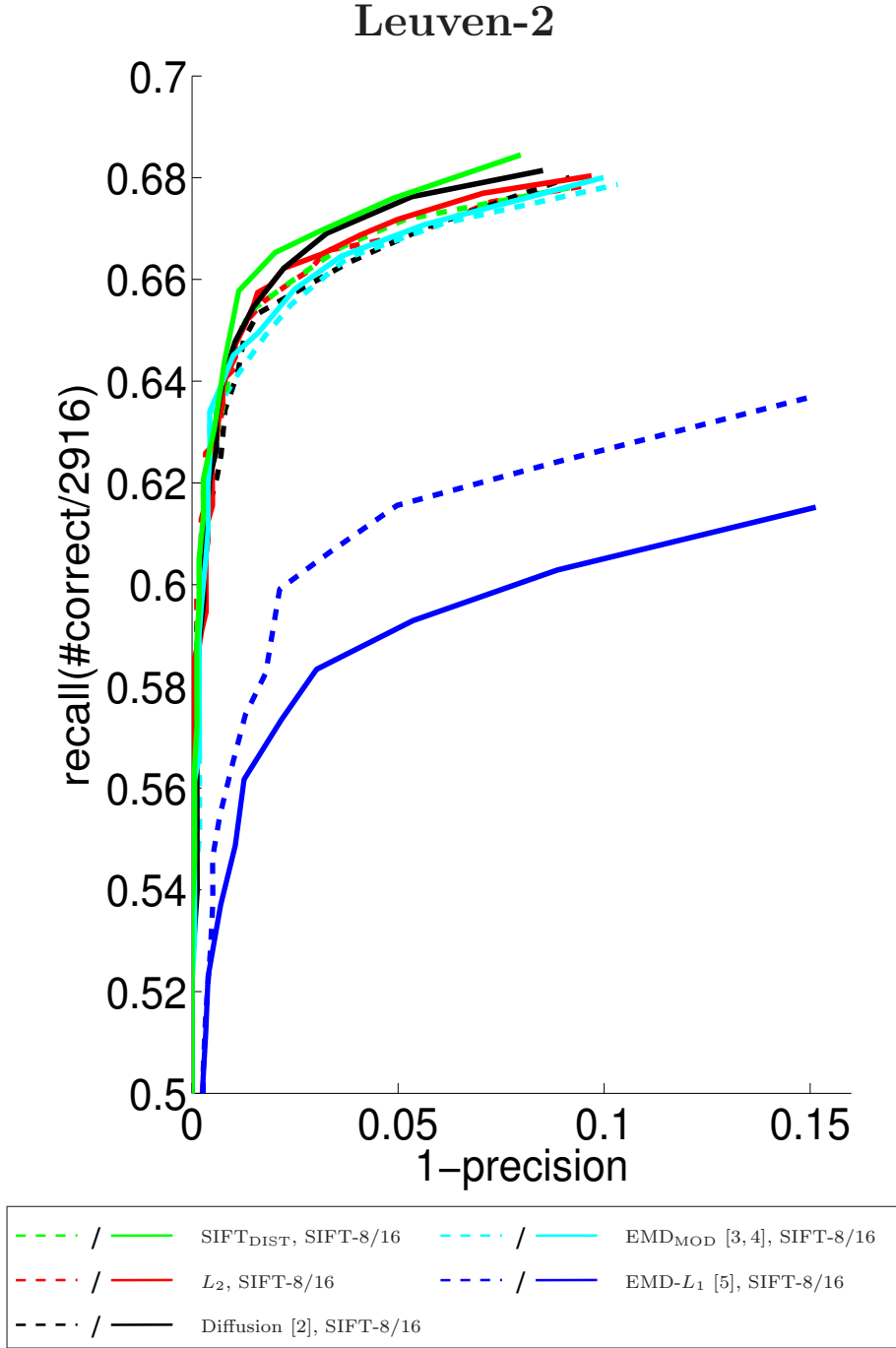


Fig. 11. Results on the Mikolajczyk and Schmid dataset [6]. Should be viewed in color.

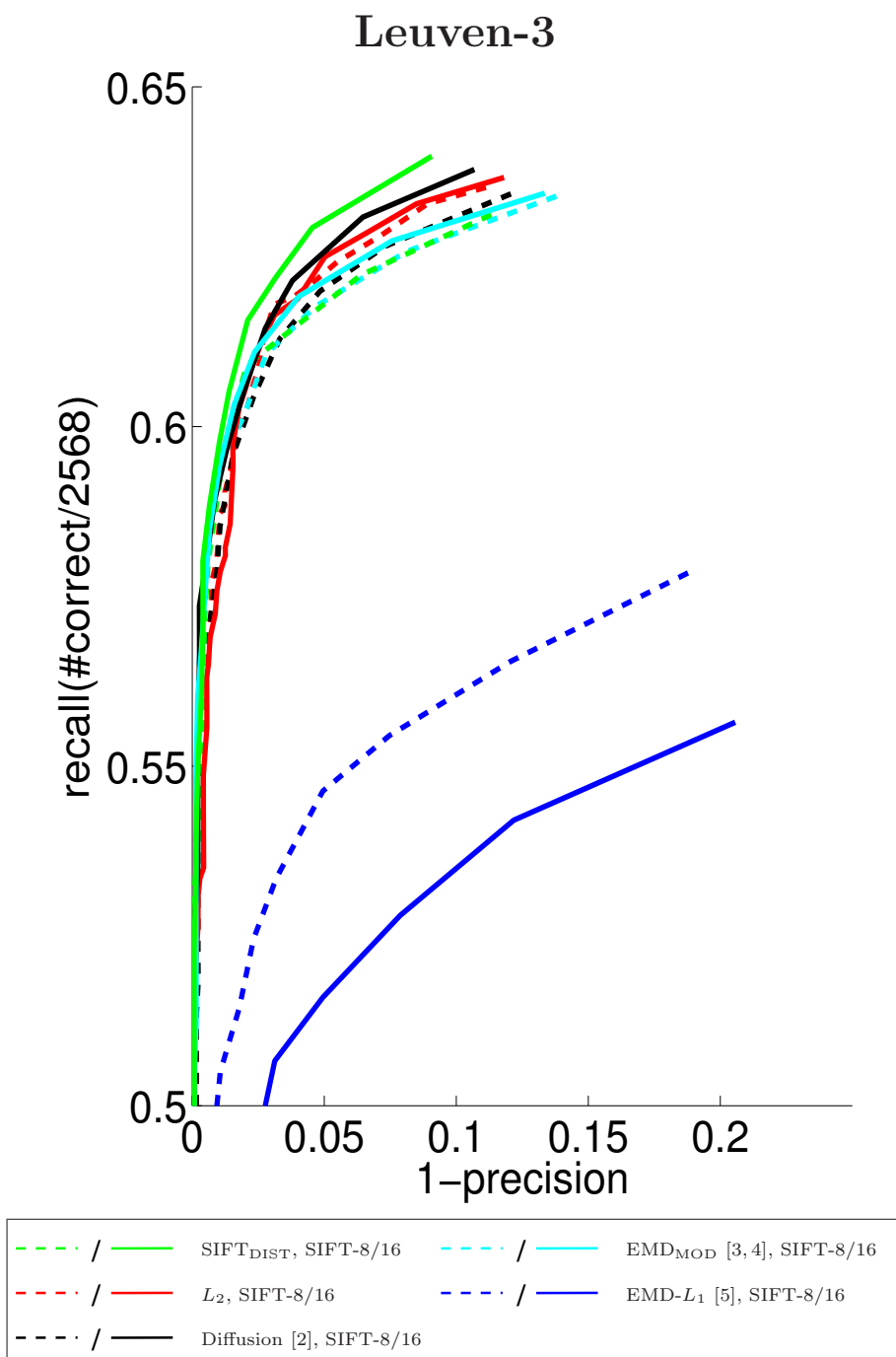


Fig. 12. Results on the Mikolajczyk and Schmid dataset [6]. Should be viewed in color.

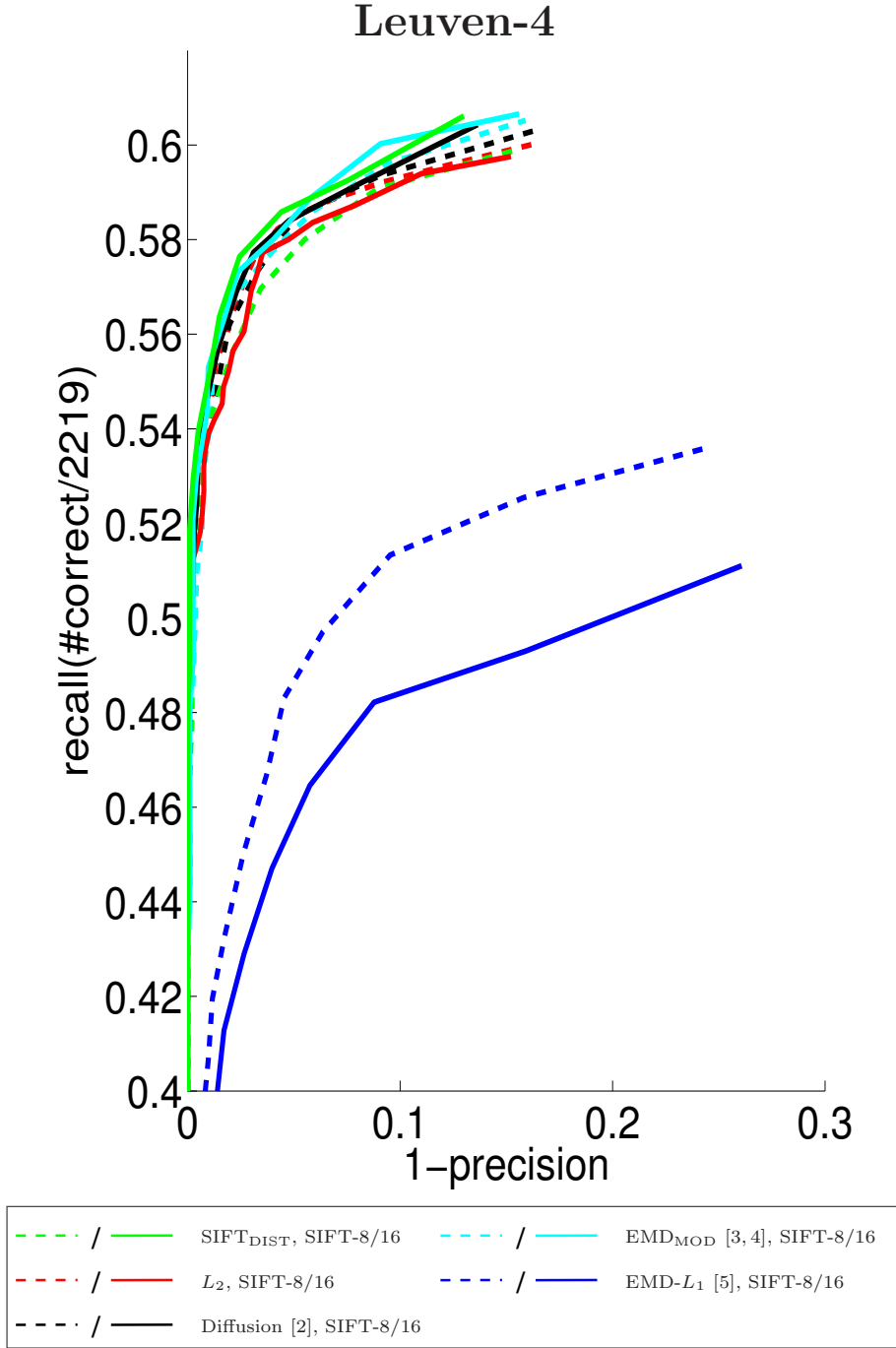


Fig. 13. Results on the Mikolajczyk and Schmid dataset [6]. Should be viewed in color.

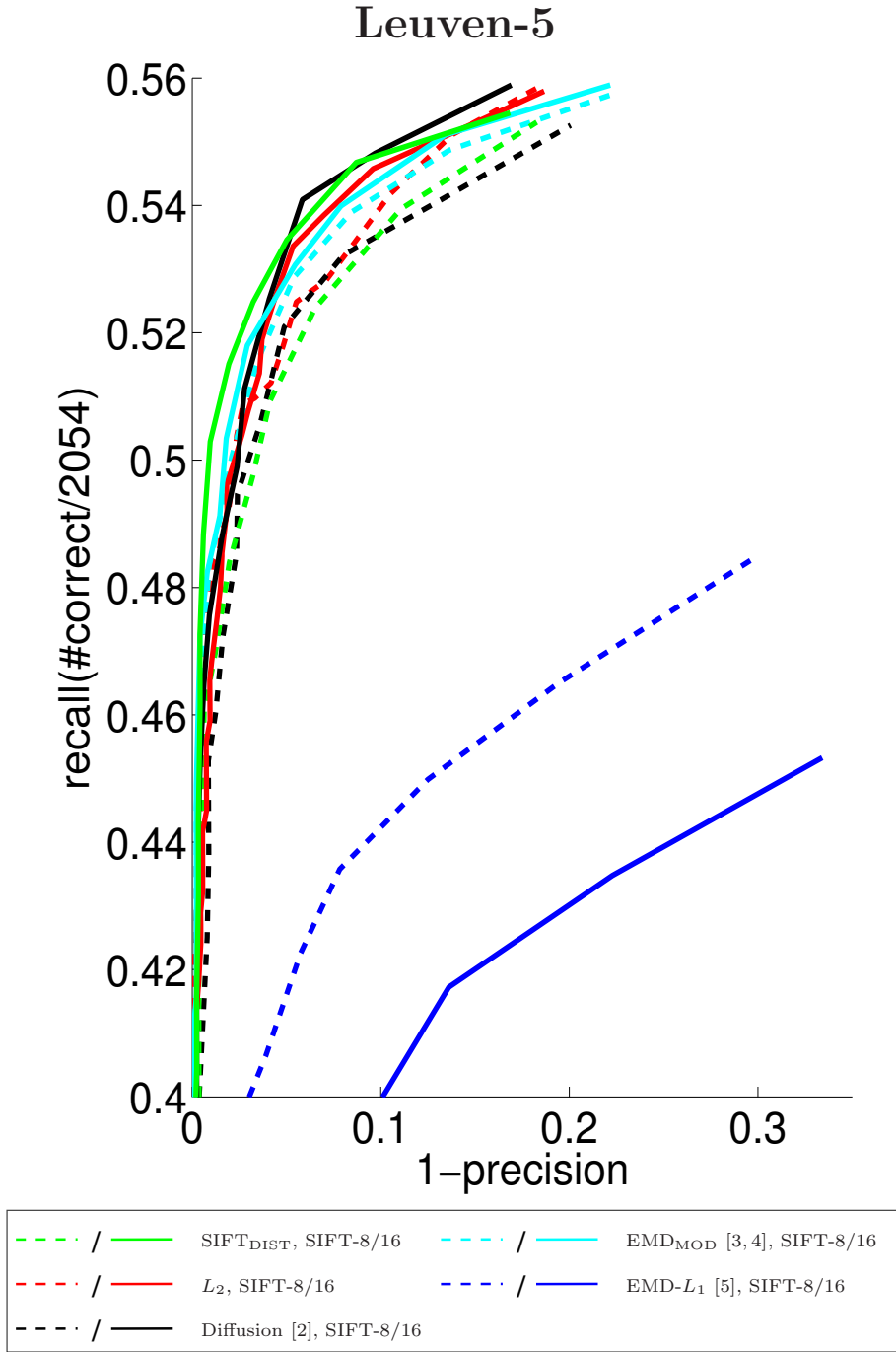


Fig. 14. Results on the Mikolajczyk and Schmid dataset [6]. Should be viewed in color.

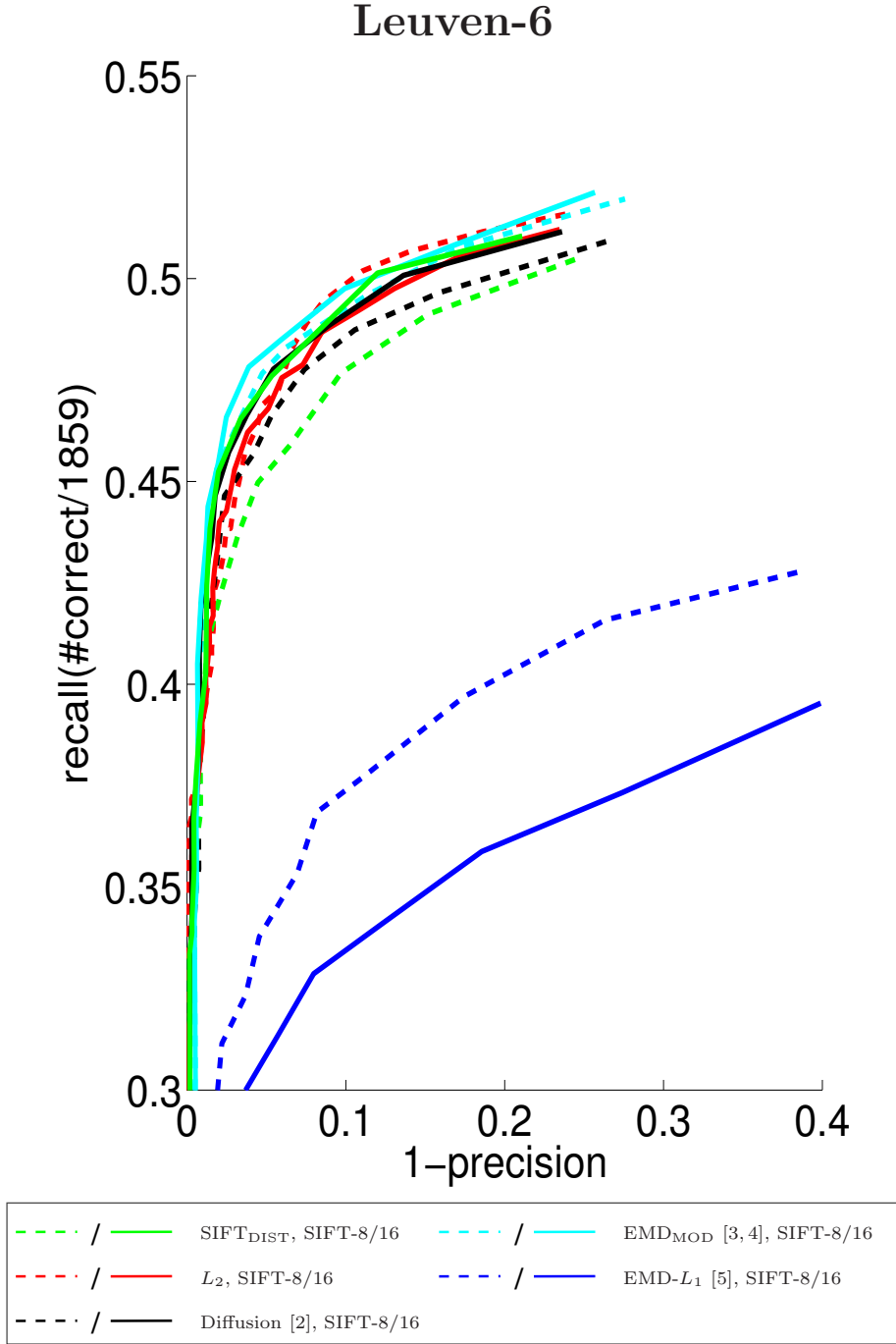


Fig. 15. Results on the Mikolajczyk and Schmid dataset [6]. Should be viewed in color.

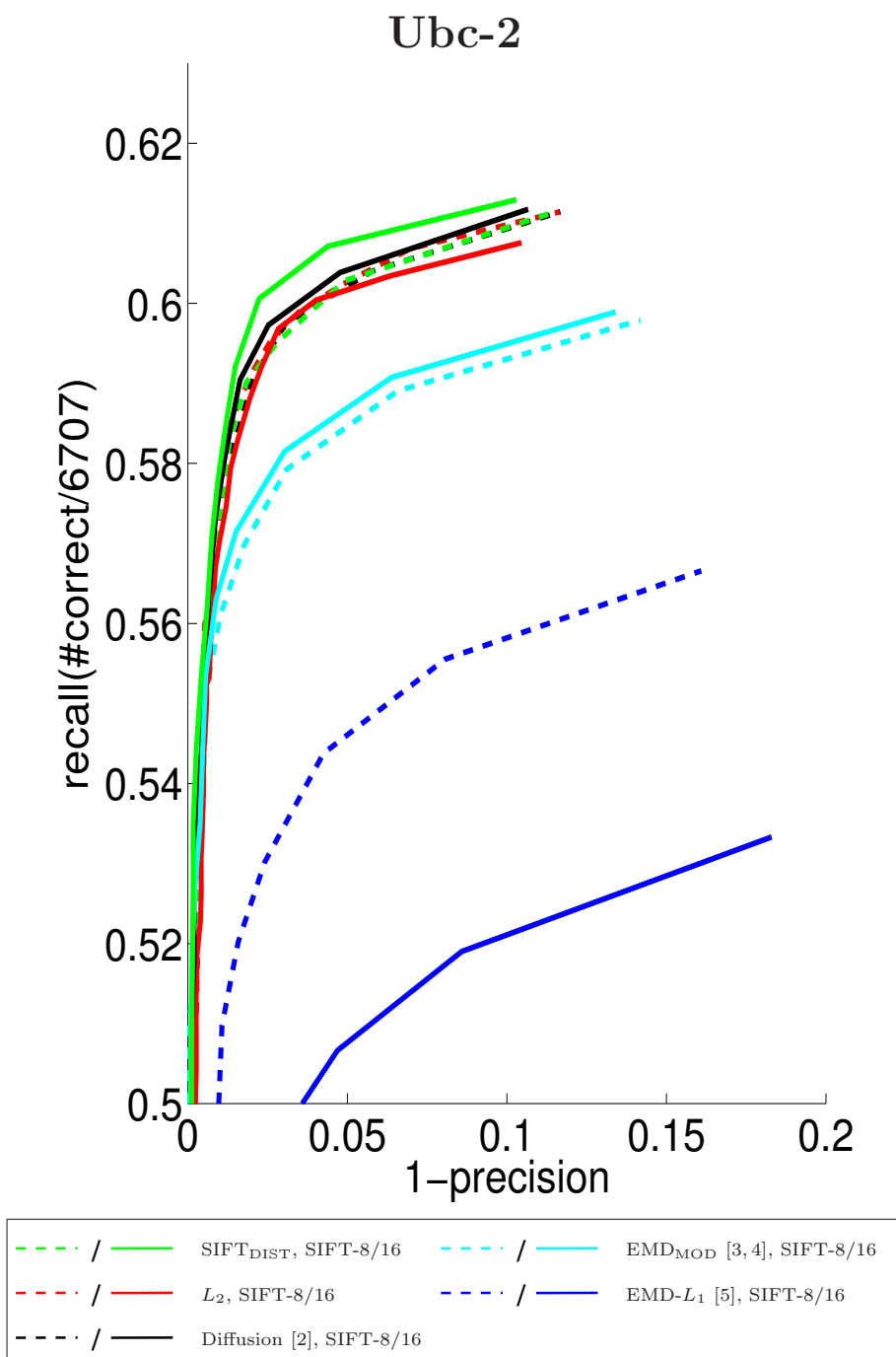


Fig. 16. Results on the Mikolajczyk and Schmid dataset [6]. Should be viewed in color.

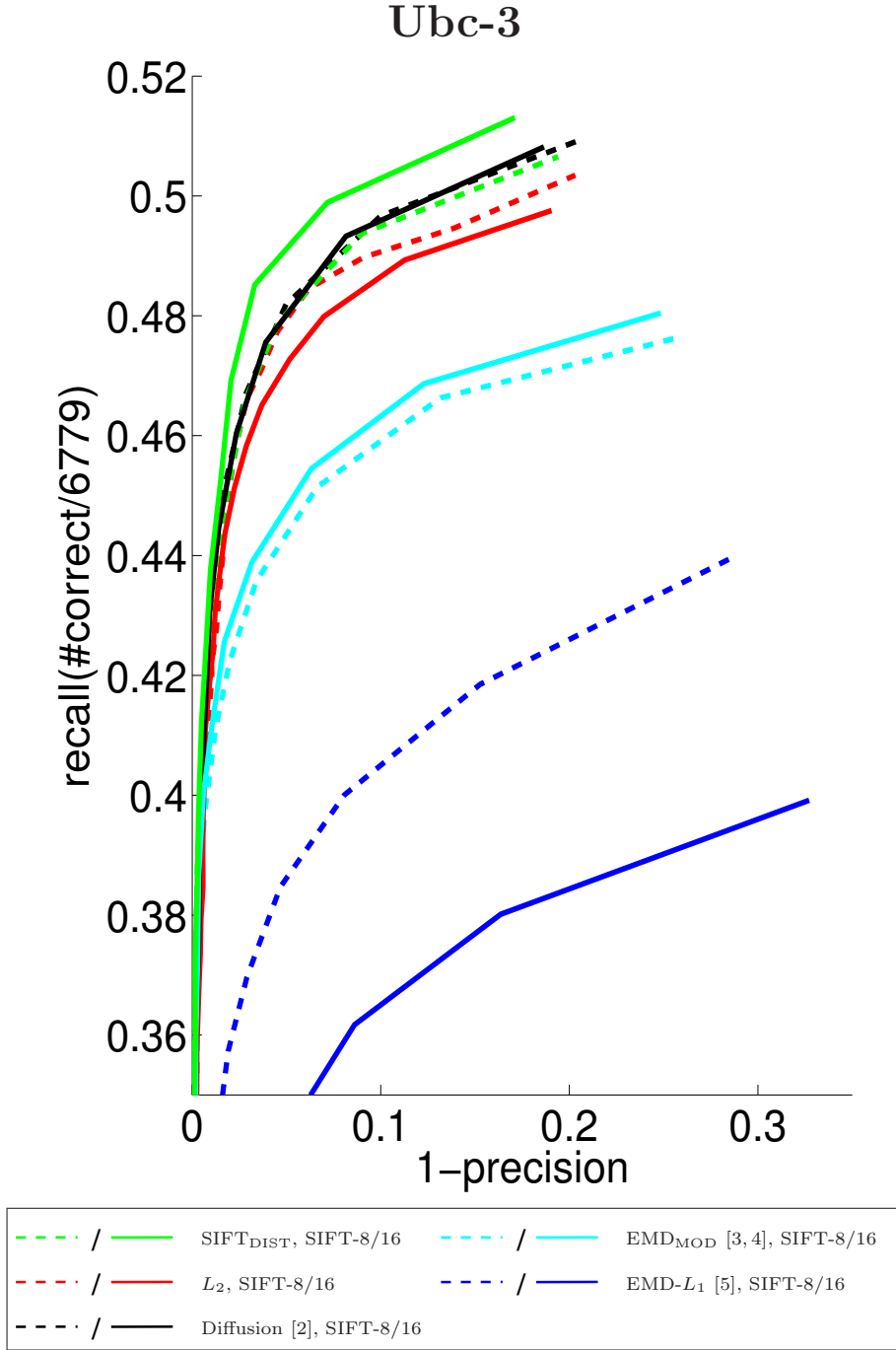


Fig. 17. Results on the Mikolajczyk and Schmid dataset [6]. Should be viewed in color.

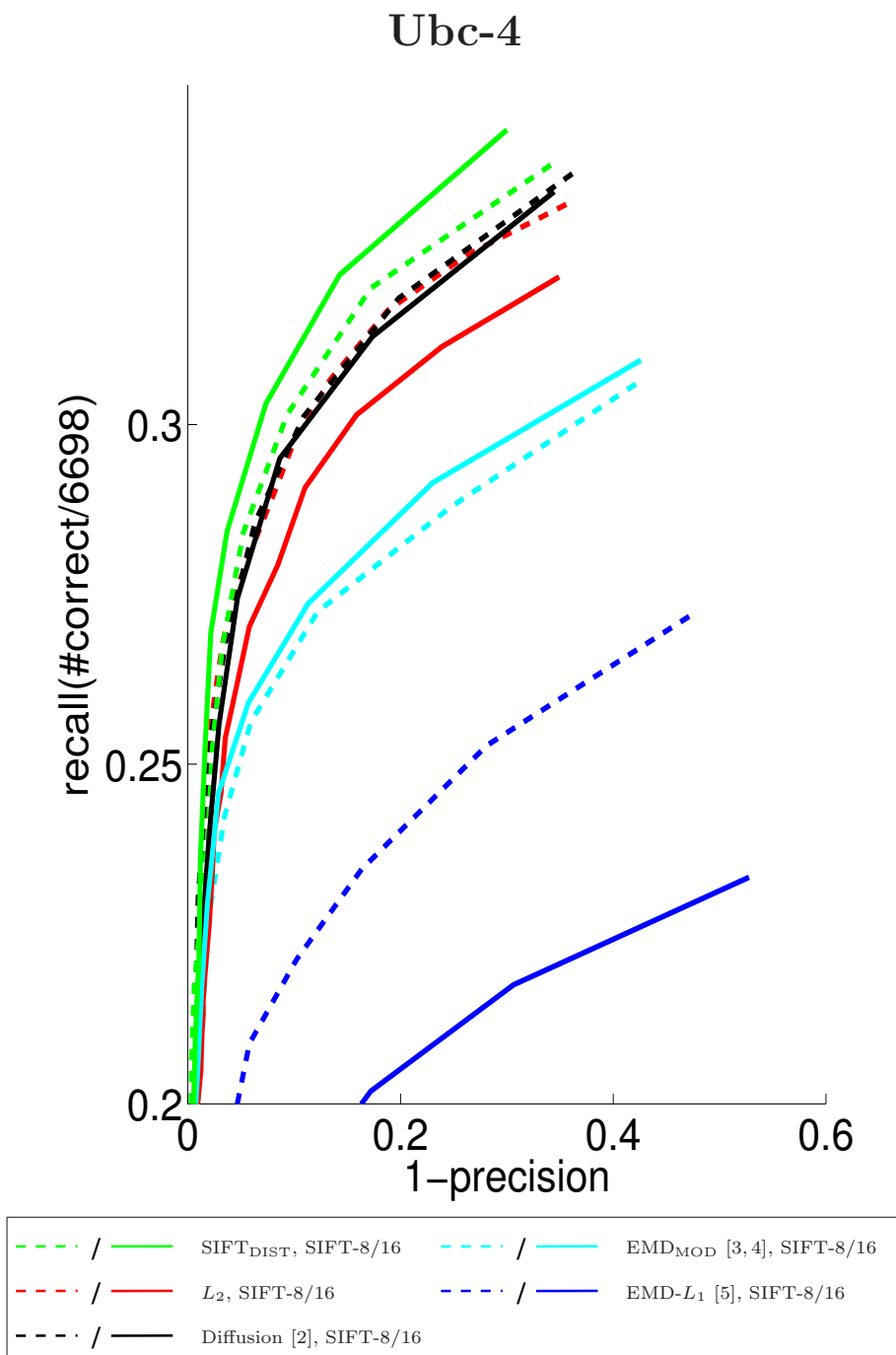


Fig. 18. Results on the Mikolajczyk and Schmid dataset [6]. Should be viewed in color.

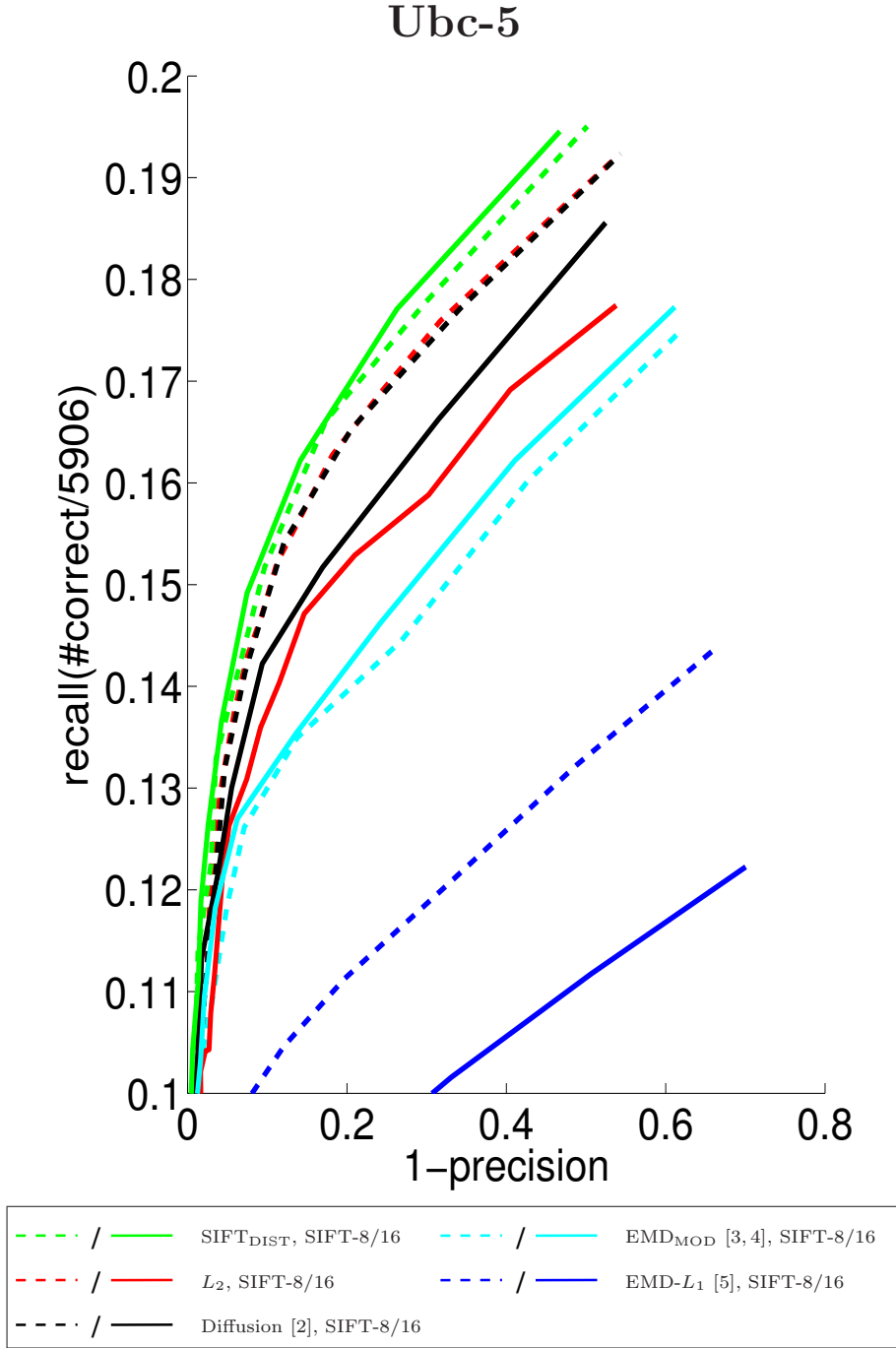


Fig. 19. Results on the Mikolajczyk and Schmid dataset [6]. Should be viewed in color.

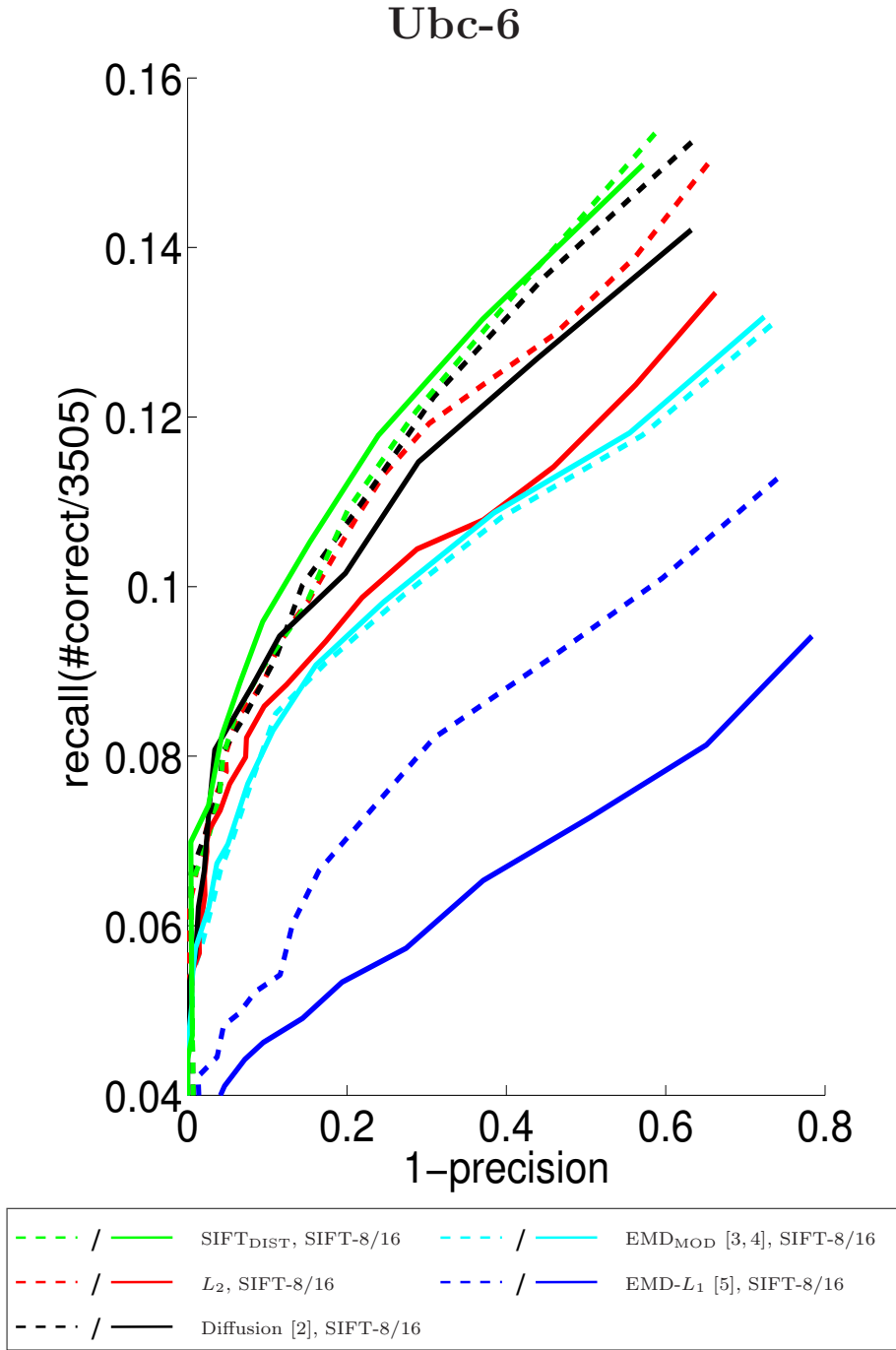


Fig. 20. Results on the Mikolajczyk and Schmid dataset [6]. Should be viewed in color.

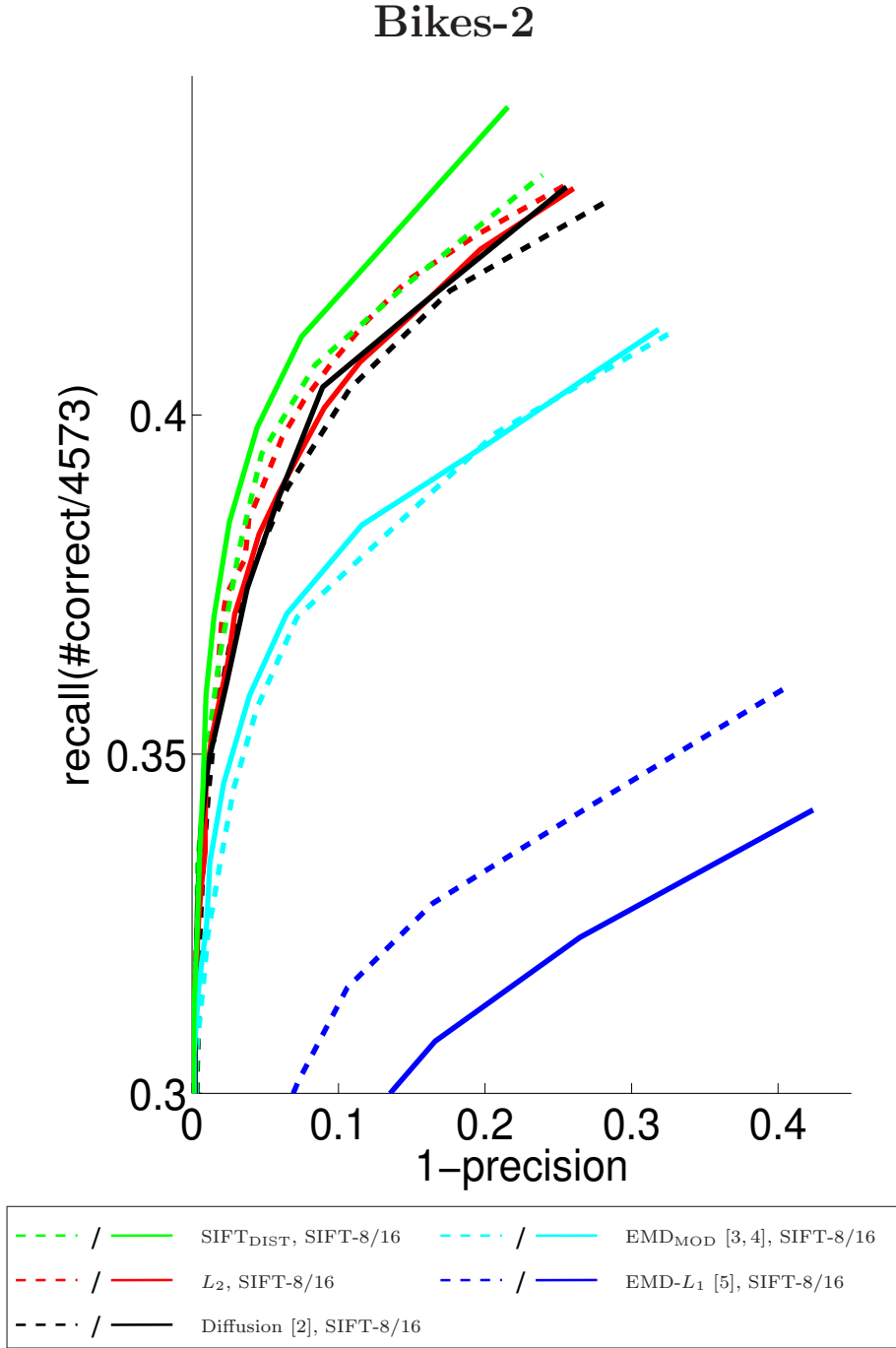


Fig. 21. Results on the Mikolajczyk and Schmid dataset [6]. Should be viewed in color.

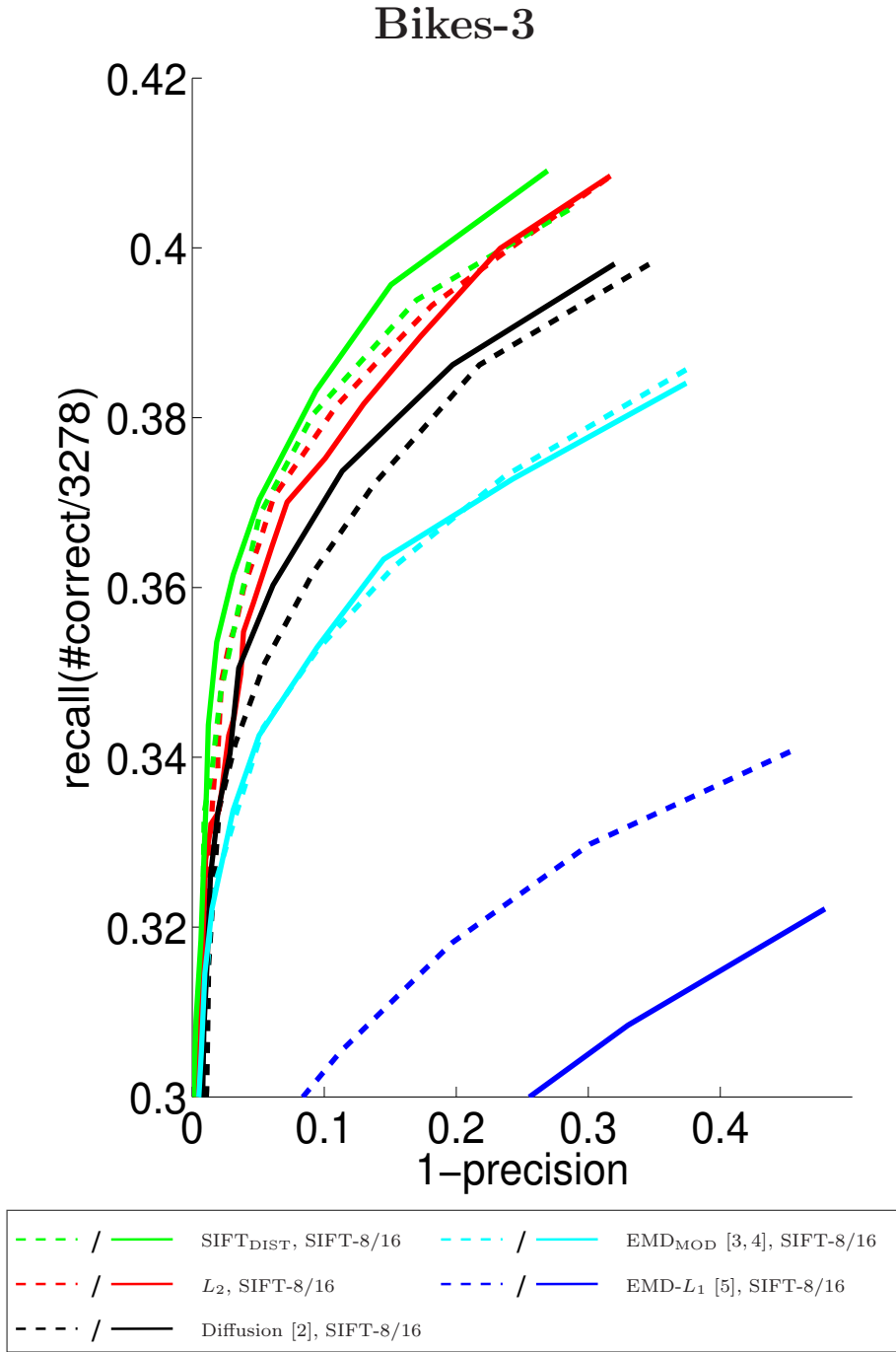


Fig. 22. Results on the Mikolajczyk and Schmid dataset [6]. Should be viewed in color.

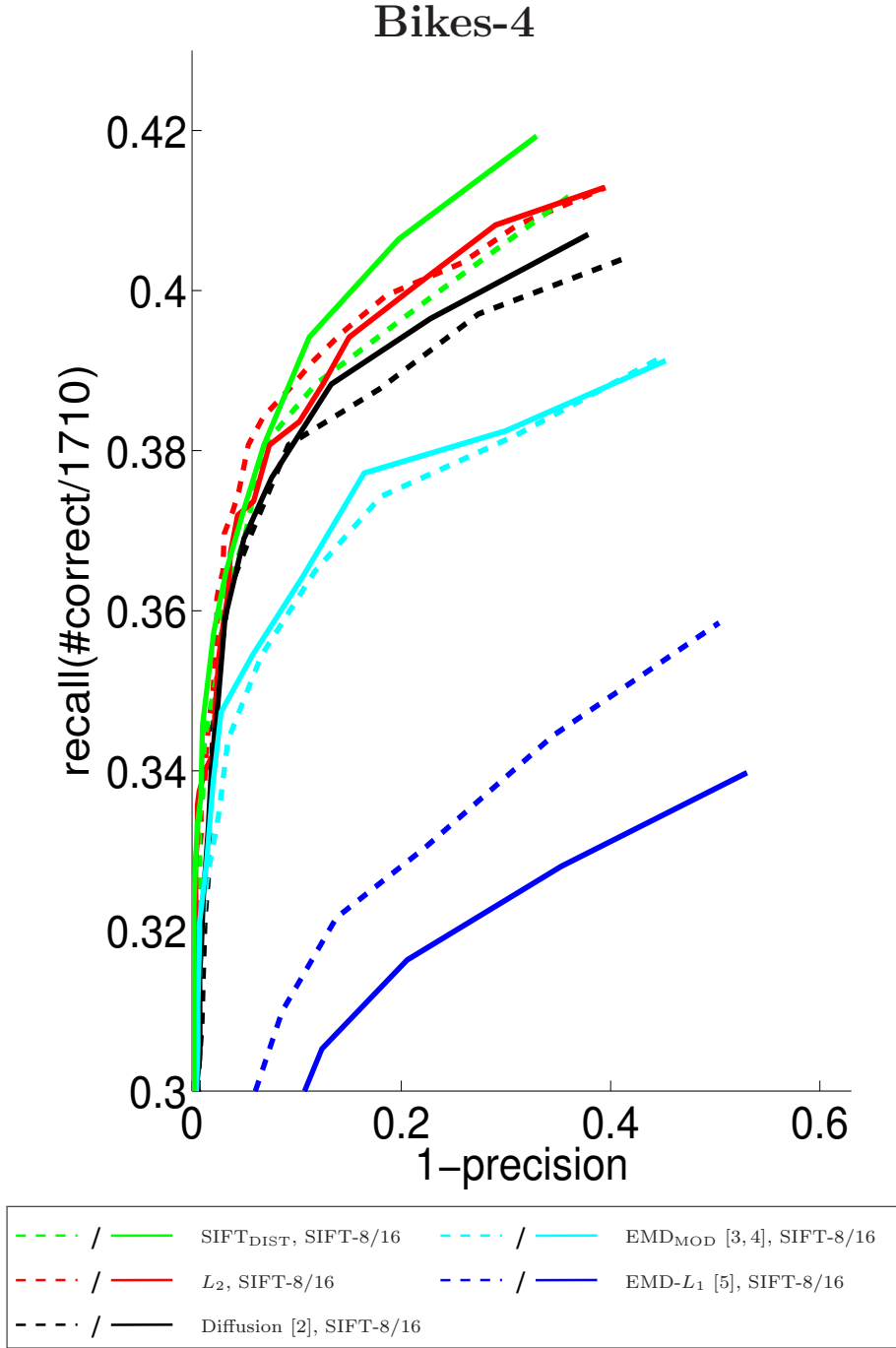


Fig. 23. Results on the Mikolajczyk and Schmid dataset [6]. Should be viewed in color.

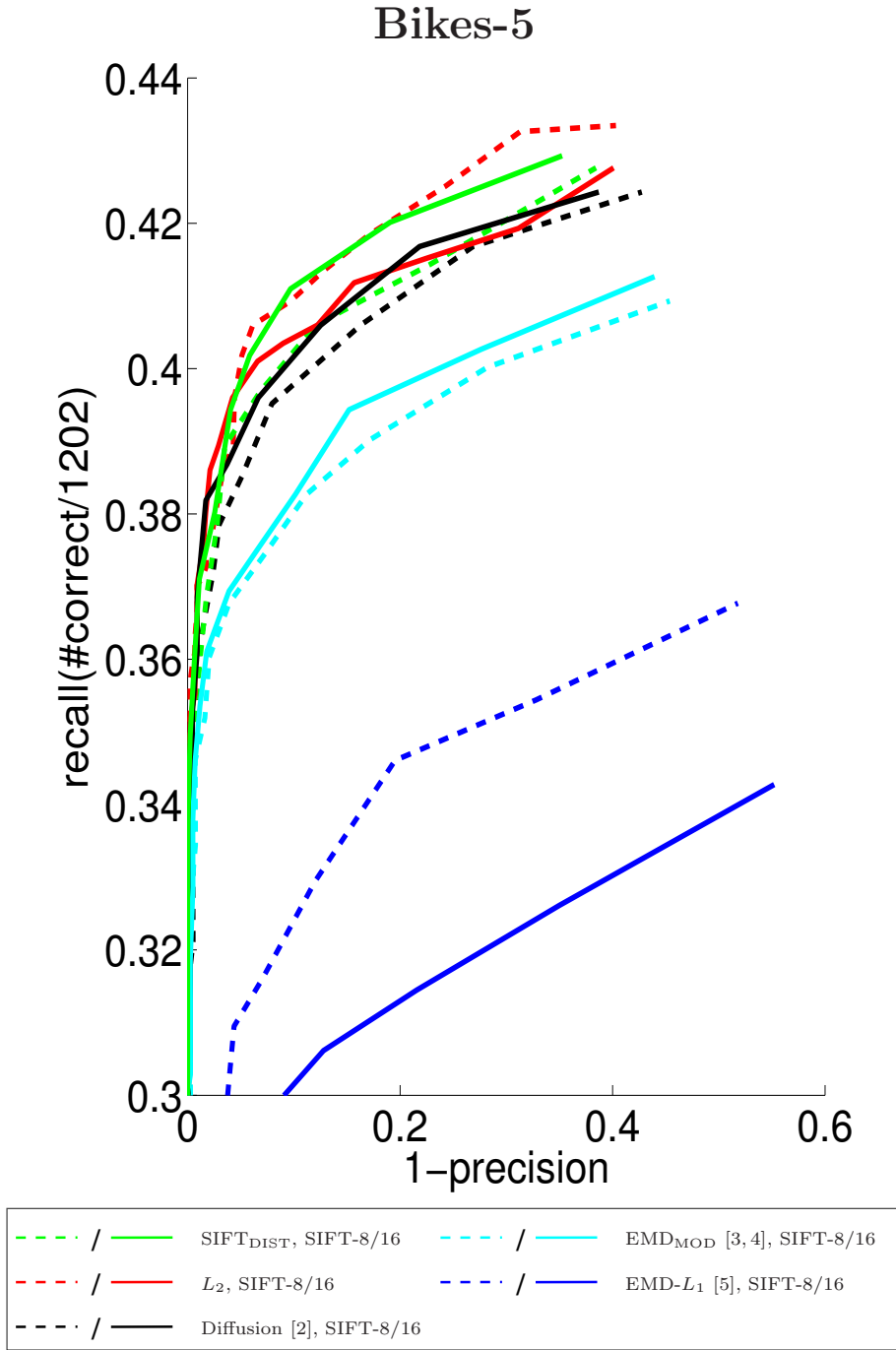


Fig. 24. Results on the Mikolajczyk and Schmid dataset [6]. Should be viewed in color.

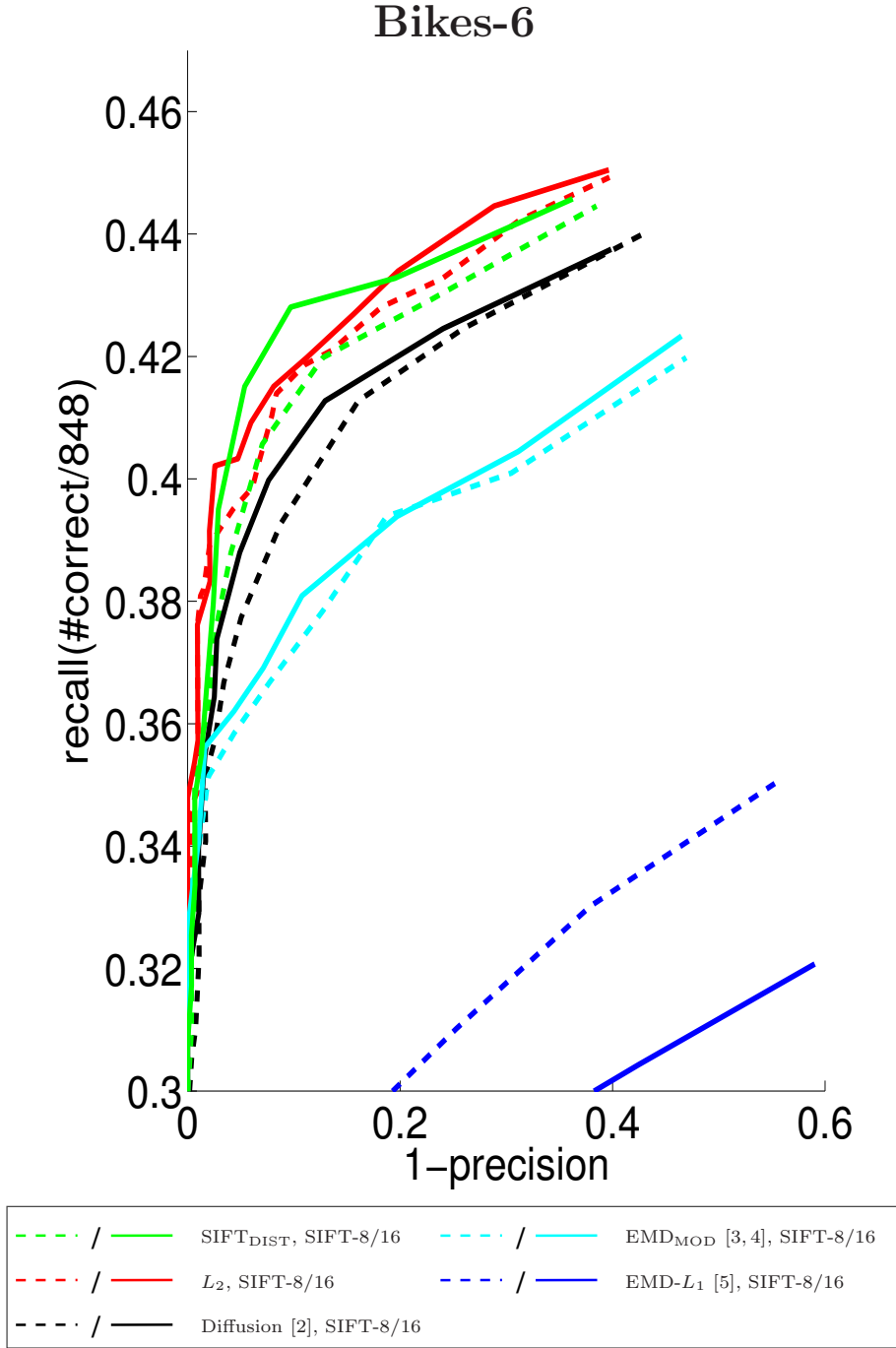


Fig. 25. Results on the Mikolajczyk and Schmid dataset [6]. Should be viewed in color.

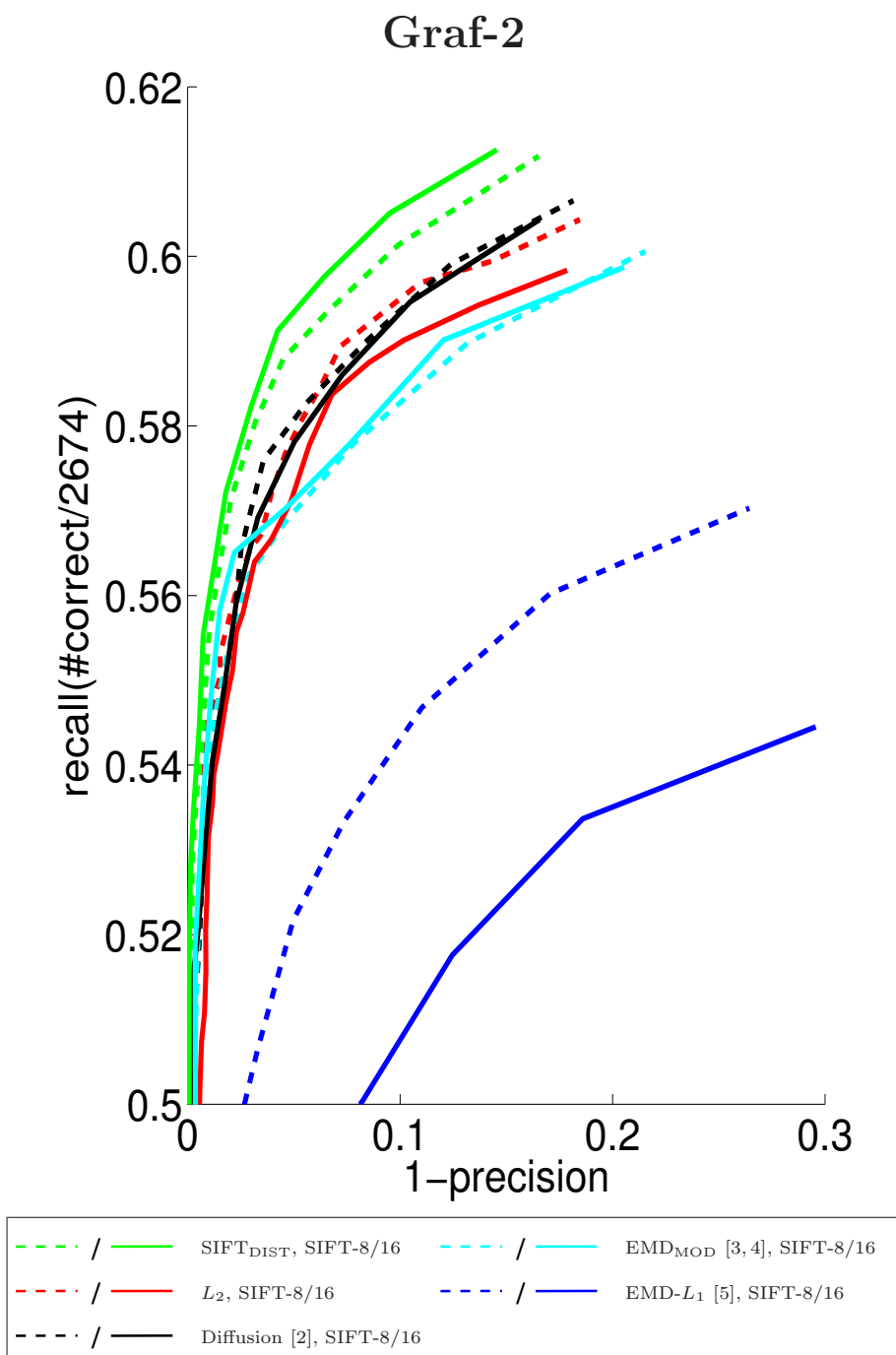


Fig. 26. Results on the Mikolajczyk and Schmid dataset [6]. Should be viewed in color.

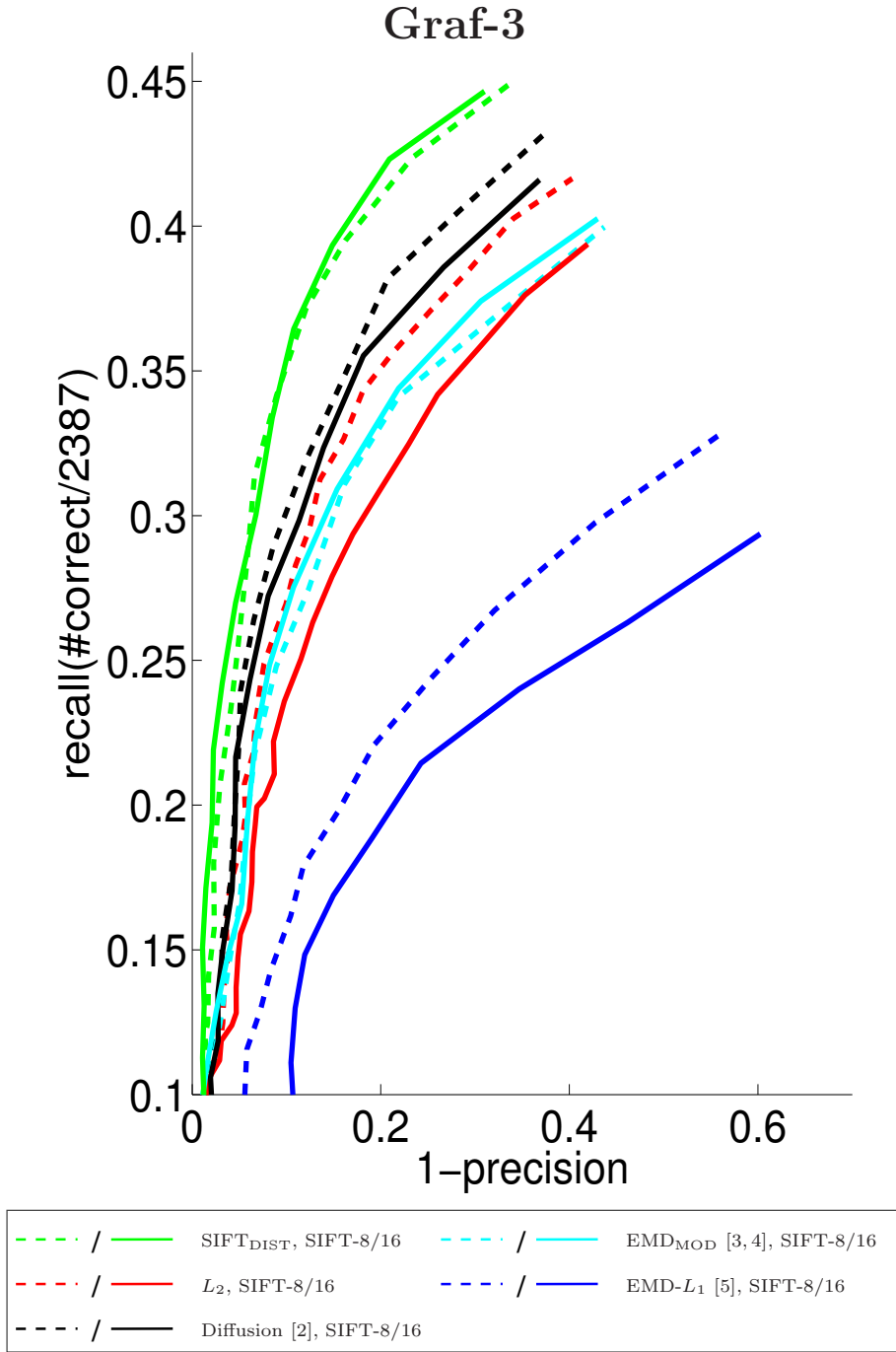


Fig. 27. Results on the Mikolajczyk and Schmid dataset [6]. Should be viewed in color.

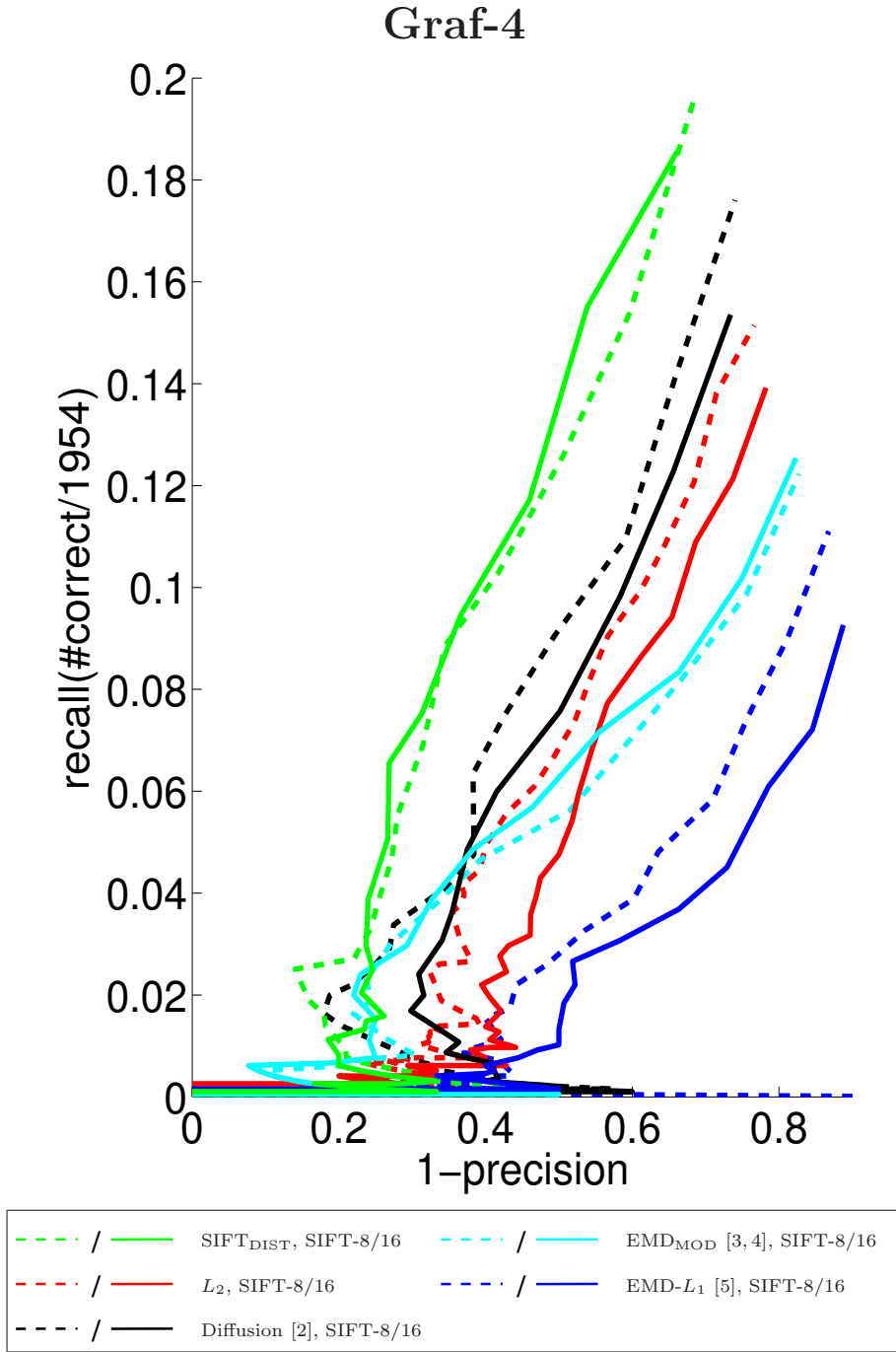


Fig. 28. Results on the Mikolajczyk and Schmid dataset [6]. Should be viewed in color.

Graf-5

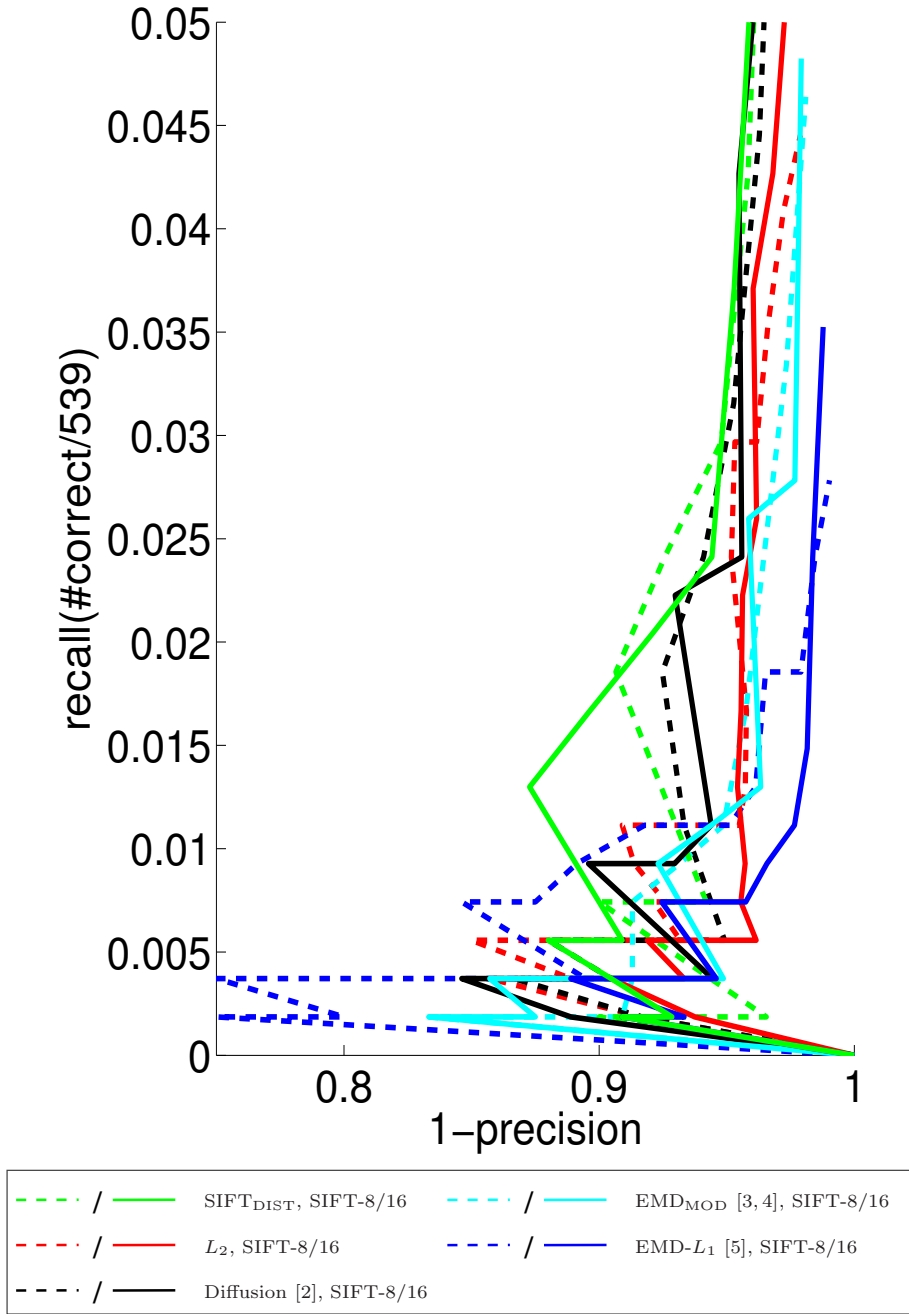


Fig. 29. Results on the Mikolajczyk and Schmid dataset [6]. Should be viewed in color.

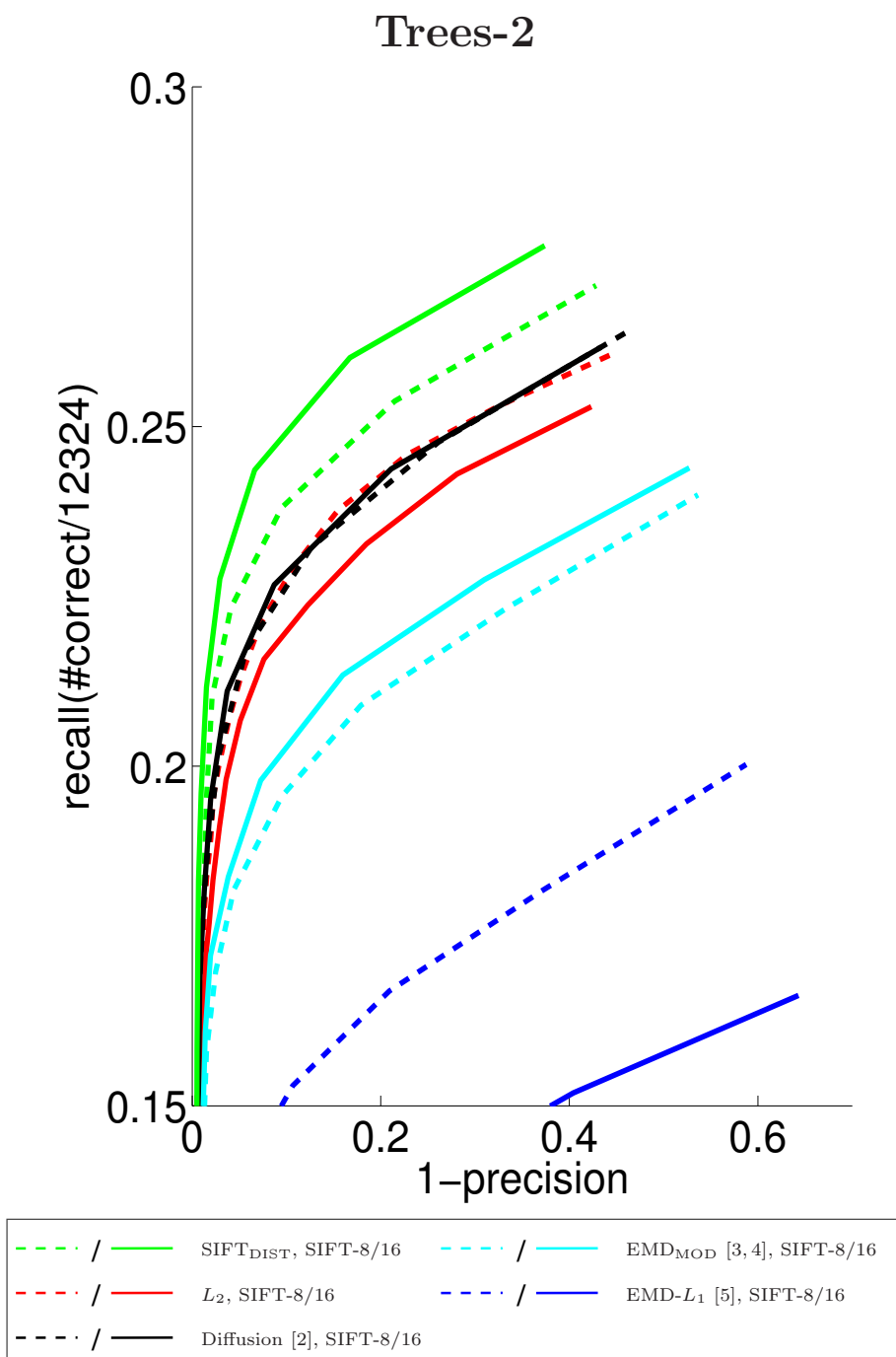


Fig. 30. Results on the Mikolajczyk and Schmid dataset [6]. Should be viewed in color.

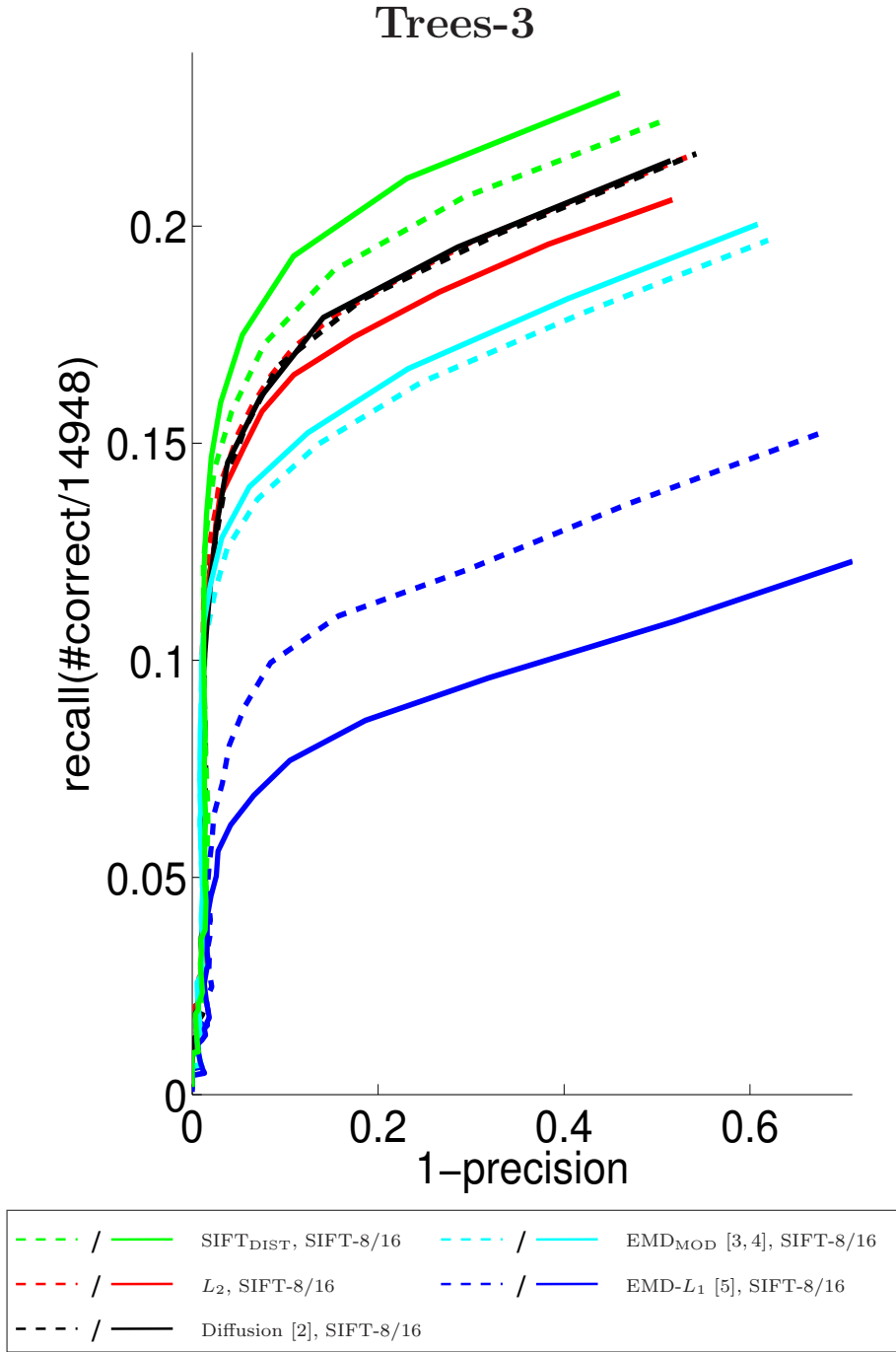


Fig. 31. Results on the Mikolajczyk and Schmid dataset [6]. Should be viewed in color.

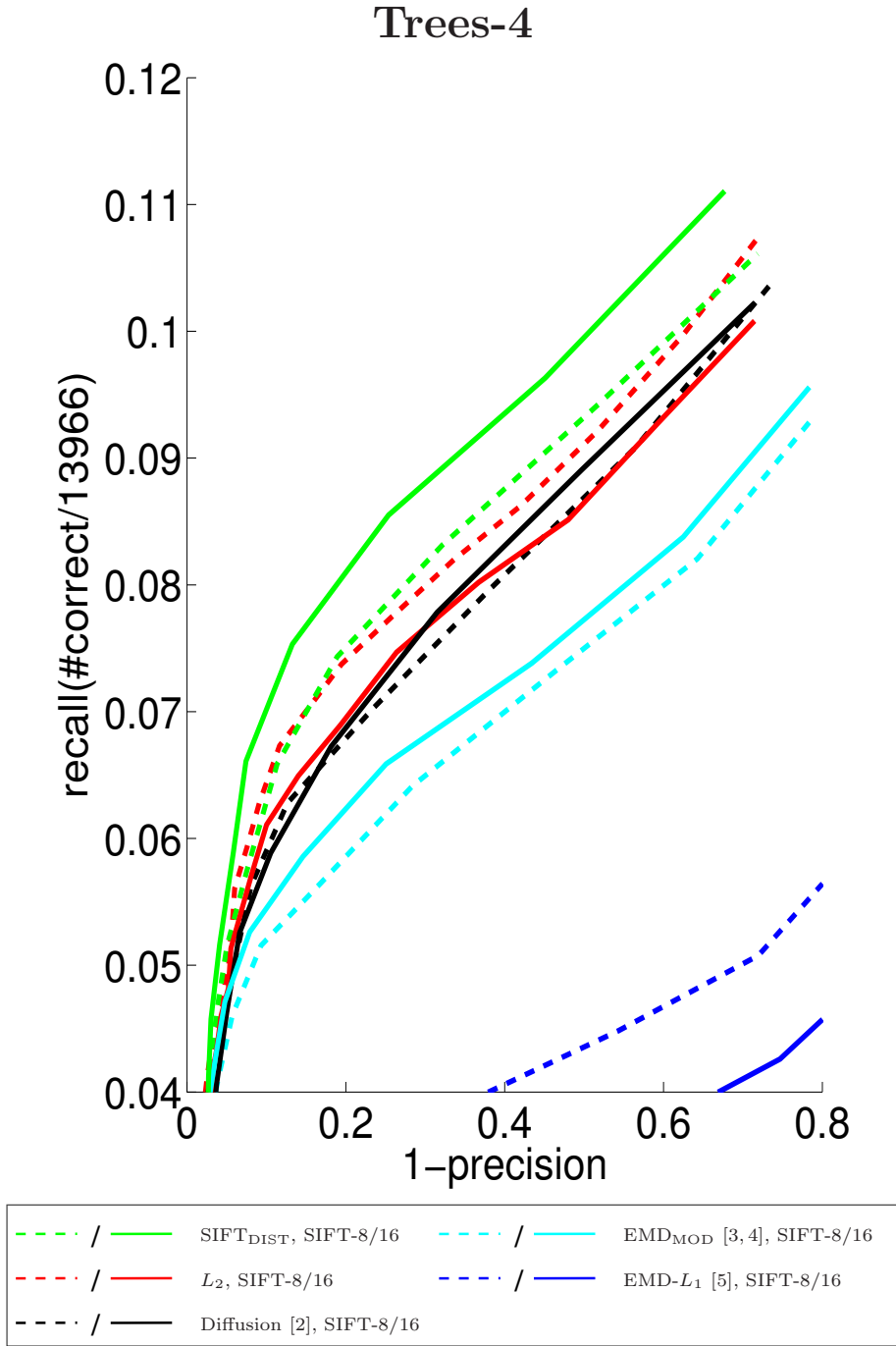


Fig. 32. Results on the Mikolajczyk and Schmid dataset [6]. Should be viewed in color.

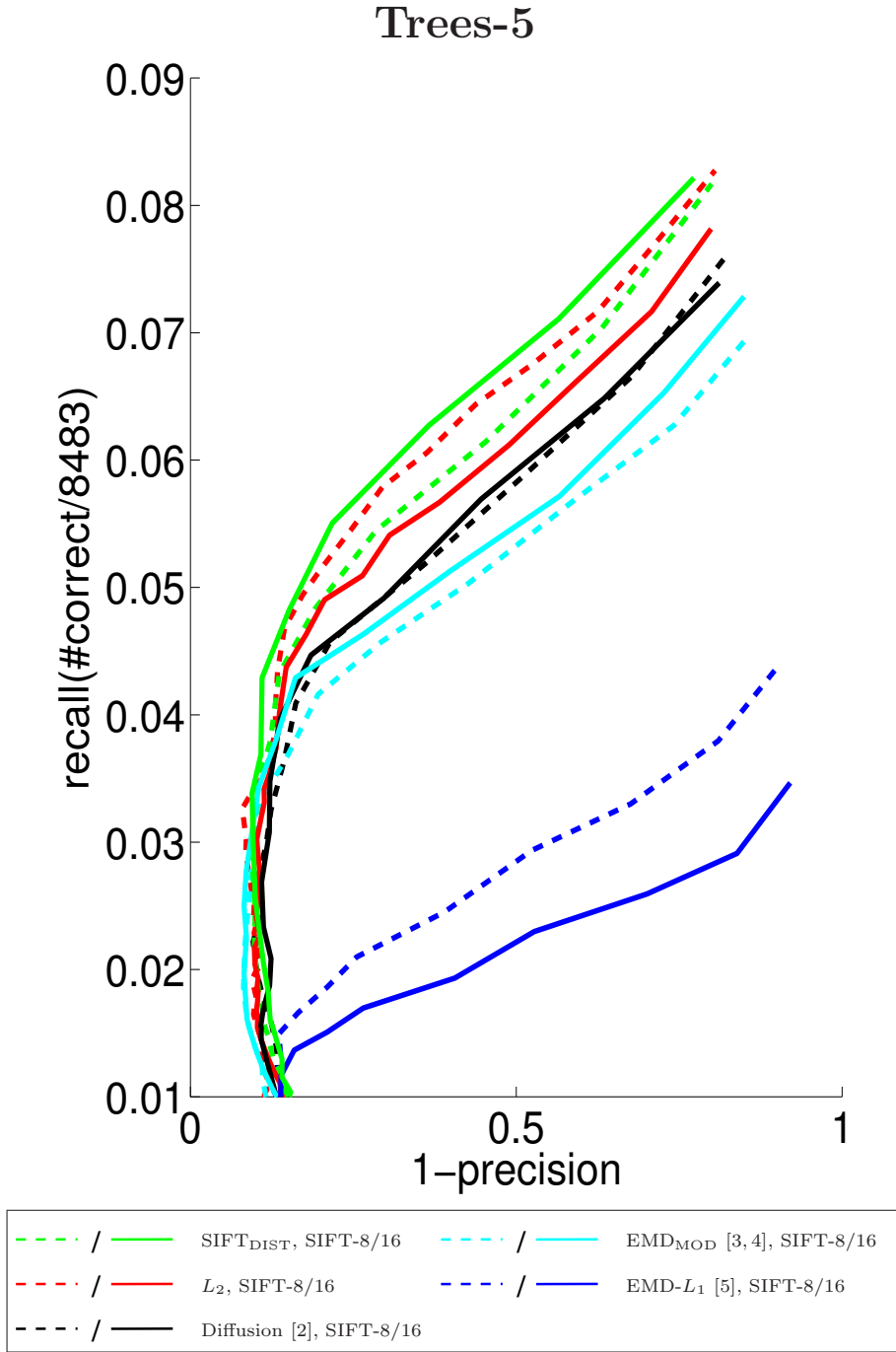


Fig. 33. Results on the Mikolajczyk and Schmid dataset [6]. Should be viewed in color.

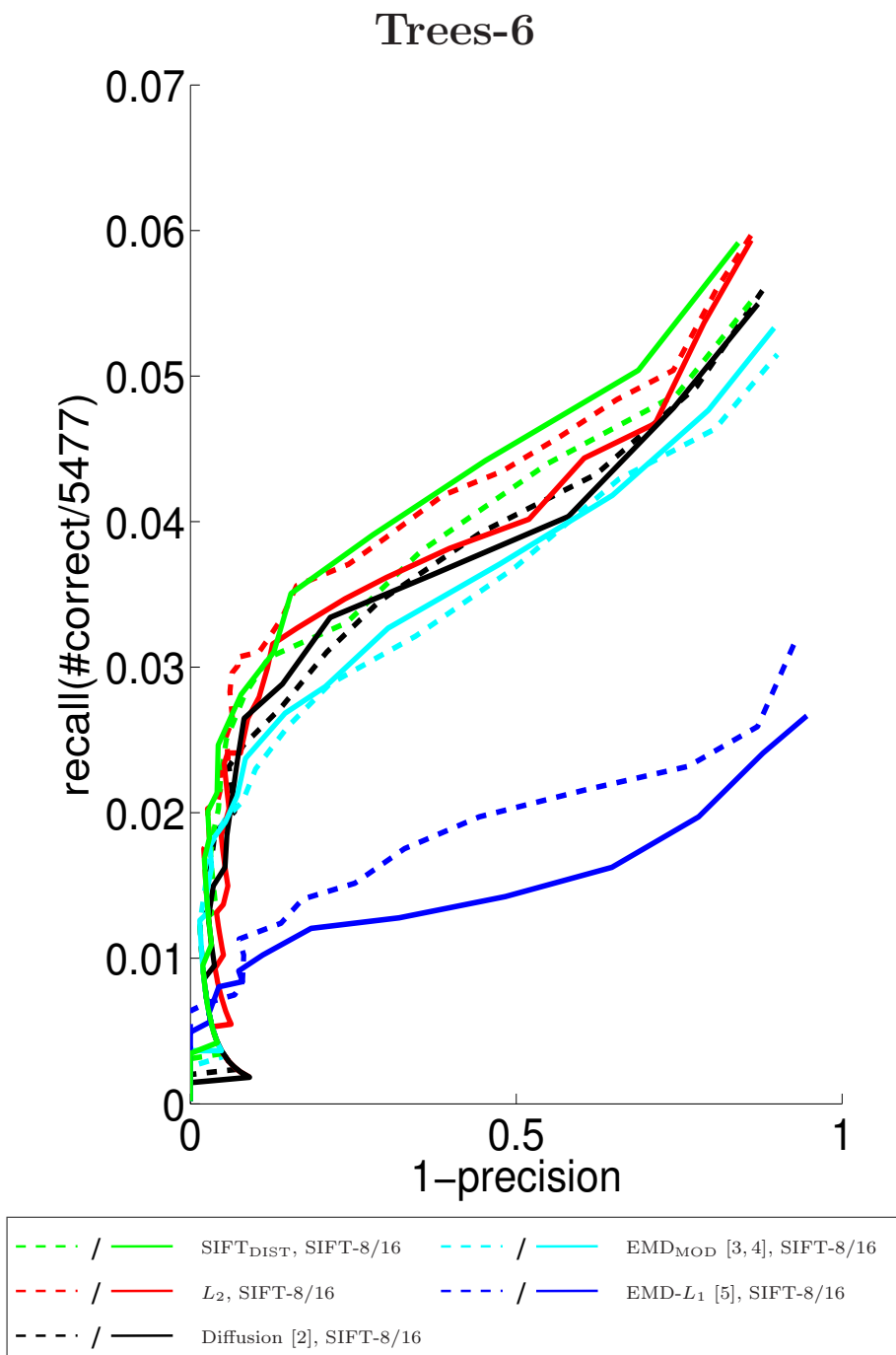


Fig. 34. Results on the Mikolajczyk and Schmid dataset [6]. Should be viewed in color.

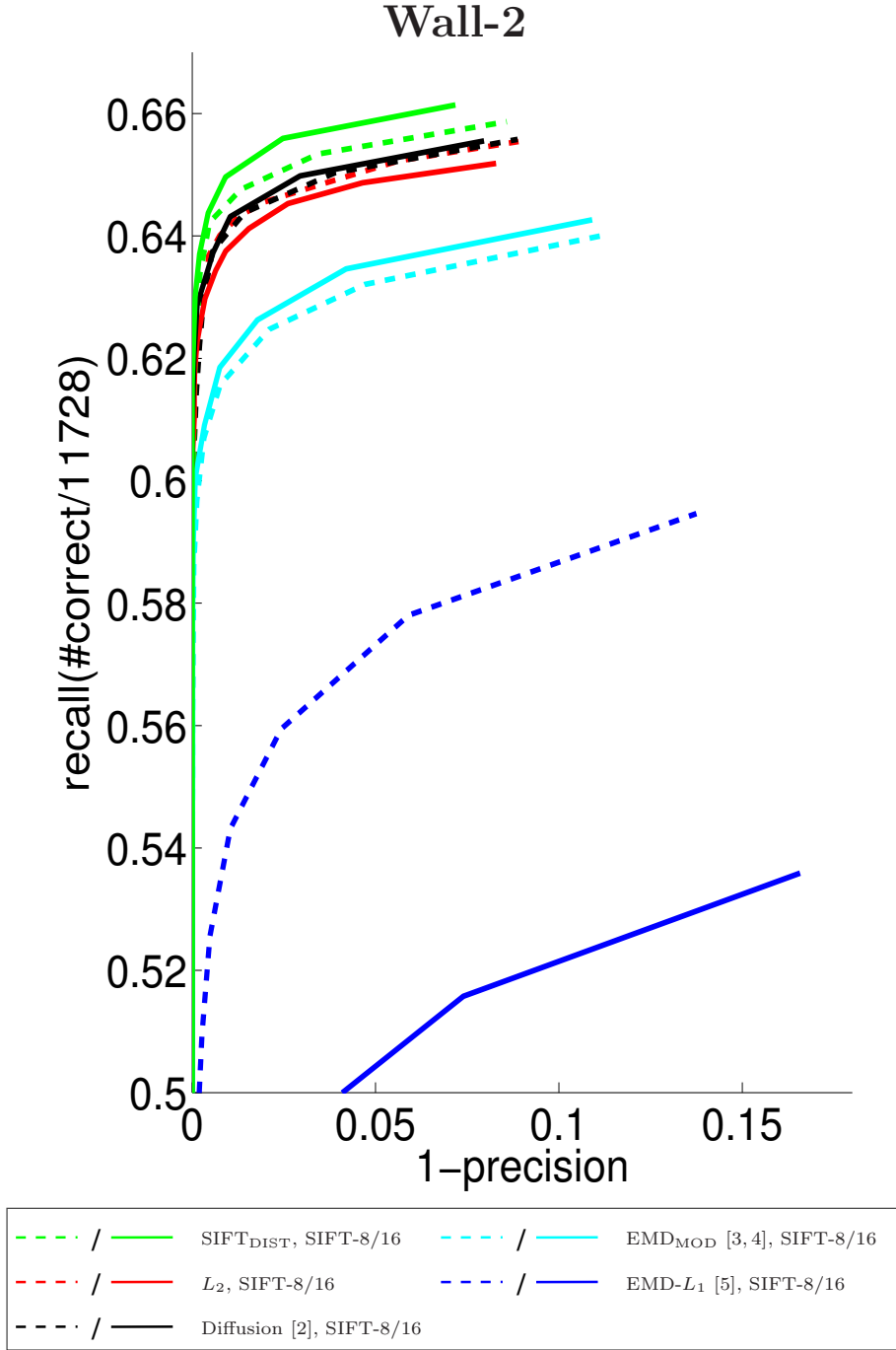


Fig. 35. Results on the Mikolajczyk and Schmid dataset [6]. Should be viewed in color.

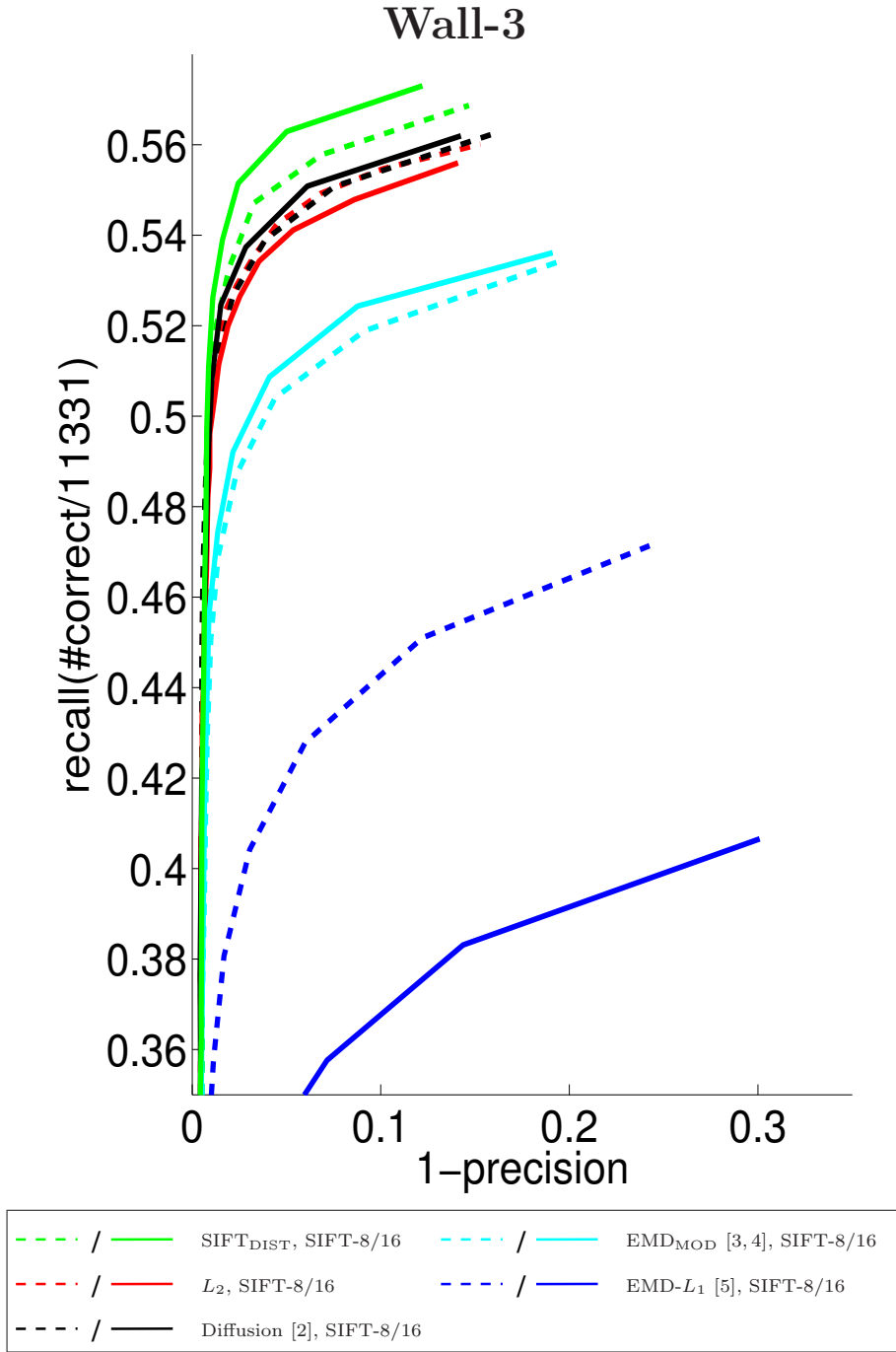


Fig. 36. Results on the Mikolajczyk and Schmid dataset [6]. Should be viewed in color.

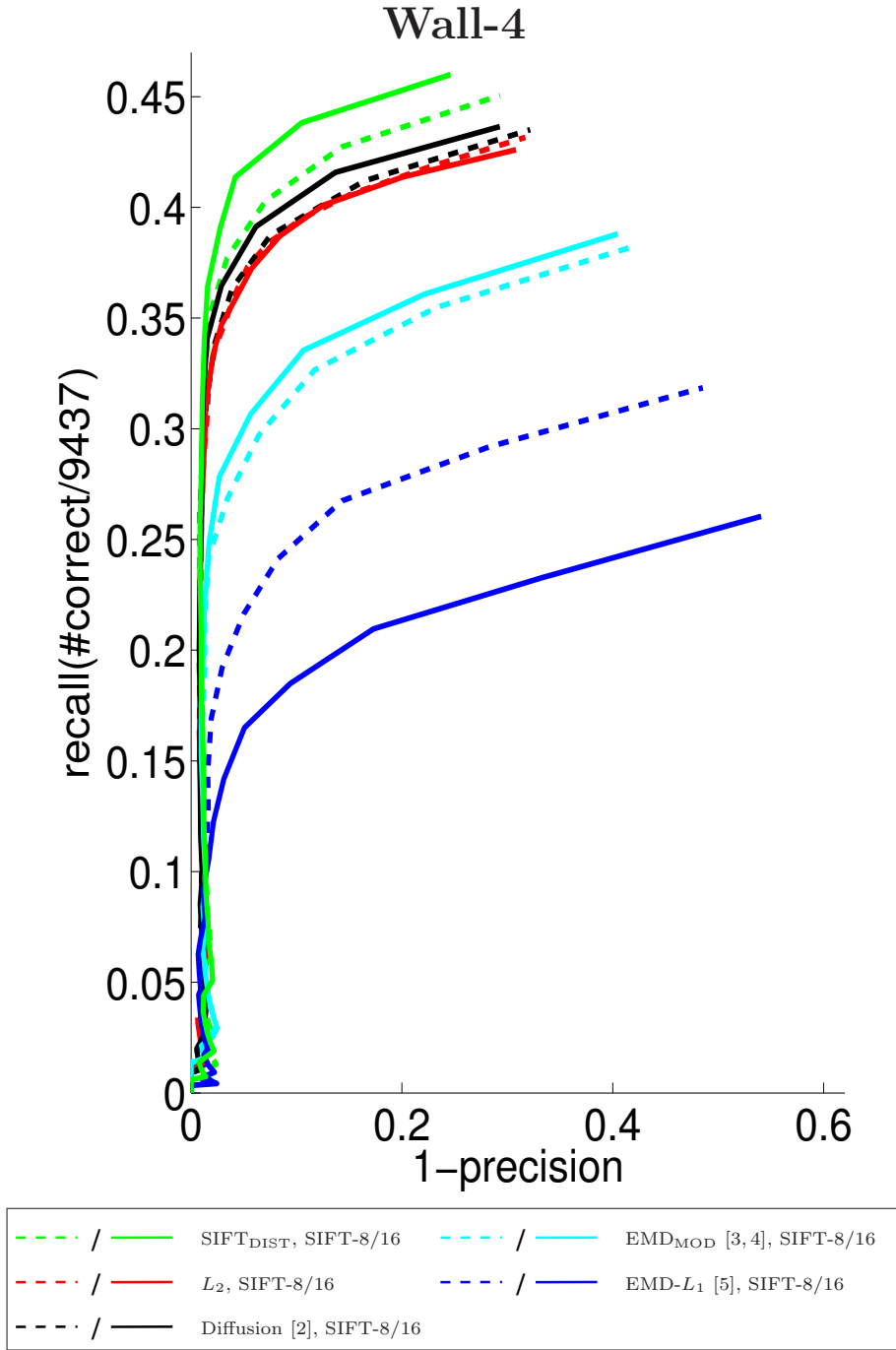


Fig. 37. Results on the Mikolajczyk and Schmid dataset [6]. Should be viewed in color.

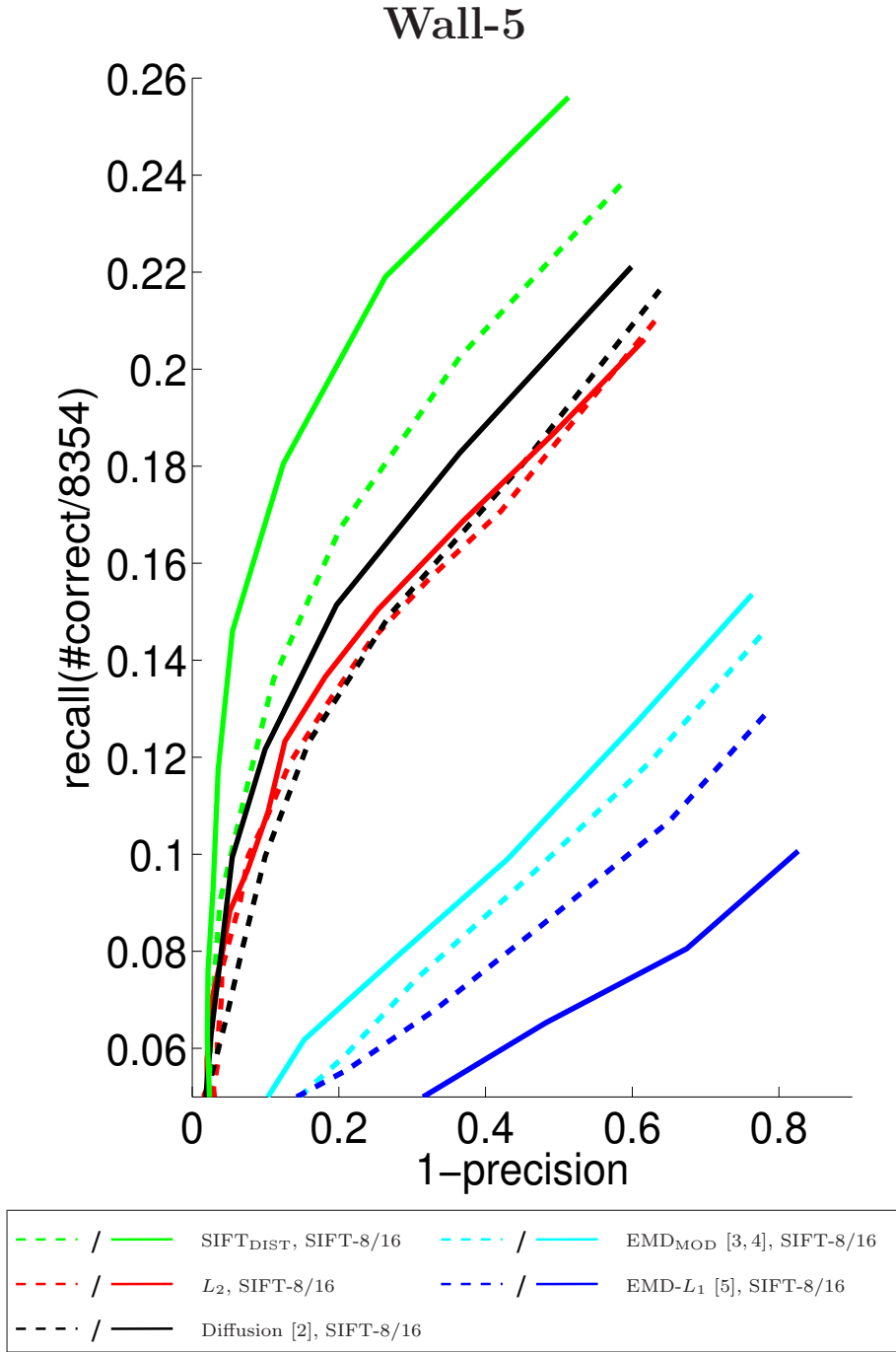


Fig. 38. Results on the Mikolajczyk and Schmid dataset [6]. Should be viewed in color.

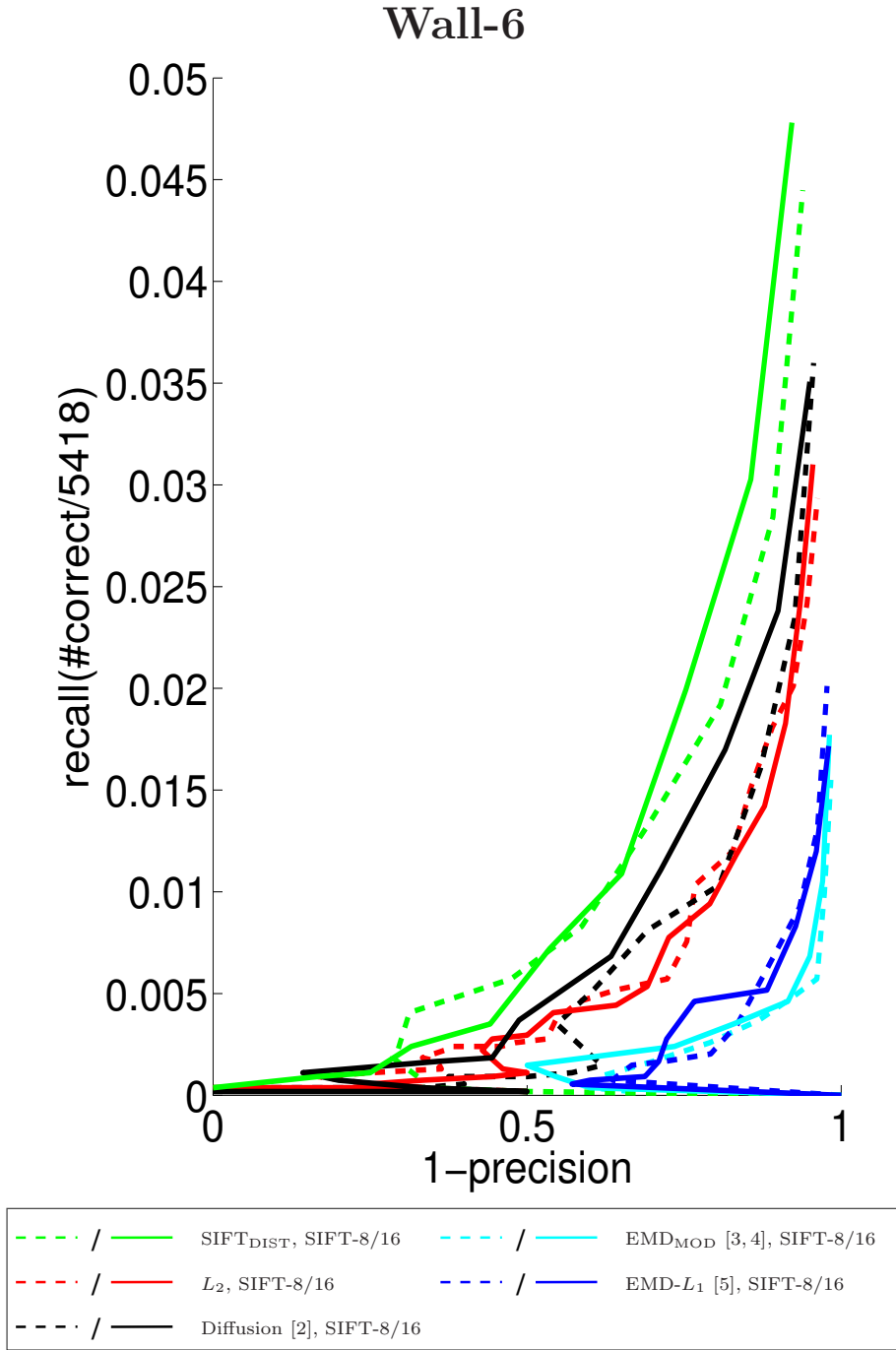


Fig. 39. Results on the Mikolajczyk and Schmid dataset [6]. Should be viewed in color.

References

1. Pele, O., Werman, M.: A linear time histogram metric for improved sift matching. In: ECCV. (2008)
2. Ling, H., Okada, K.: Diffusion distance for histogram comparison. In: CVPR. Volume 1. (2006) 246–253
3. Werman, M., Peleg, S., Melter, R., Kong, T.: Bipartite graph matching for points on a line or a circle. *Journal of Algorithms* **7**(2) (1986) 277–284
4. <http://www.cs.huji.ac.il/~ofirpele/publications/ECCV2008.pdf>
5. Ling, H., Okada, K.: An Efficient Earth Mover’s Distance Algorithm for Robust Histogram Comparison. *IEEE Trans. Pattern Analysis and Machine Intelligence* **29**(5) (2007) 840–853
6. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. *IEEE Trans. Pattern Analysis and Machine Intelligence* **27**(10) (2005) 1615–1630

Chapter 2

Fast and Robust Earth Mover's Distances - Appendices

Fast and Robust Earth Mover's Distances Additional Results - Image Retrieval Examples

Ofir Pele
The Hebrew University of Jerusalem
ofirpele@cs.huji.ac.il

Michael Werman
The Hebrew University of Jerusalem
werman@cs.huji.ac.il

This document contains additional image retrieval results for the paper - “Fast and Robust Earth Mover’s Distances” [1]. The results are for image retrieval as described in the paper. Each figure contain a query image on the left and nearest neighbors images which are ordered from left to right by their distance from the query image. Top row is for the best distance - $\widehat{EMD}$ with thresholded ground distances. Bottom row is for the second best distance - L_1 like distance. By allowing small deformations in the $\widehat{EMD}$ we obtain results that are visually similar to the query image. It is noteworthy that in some cases the returned images are not from the same class, but are still visually similar. For example, Fig. 2.

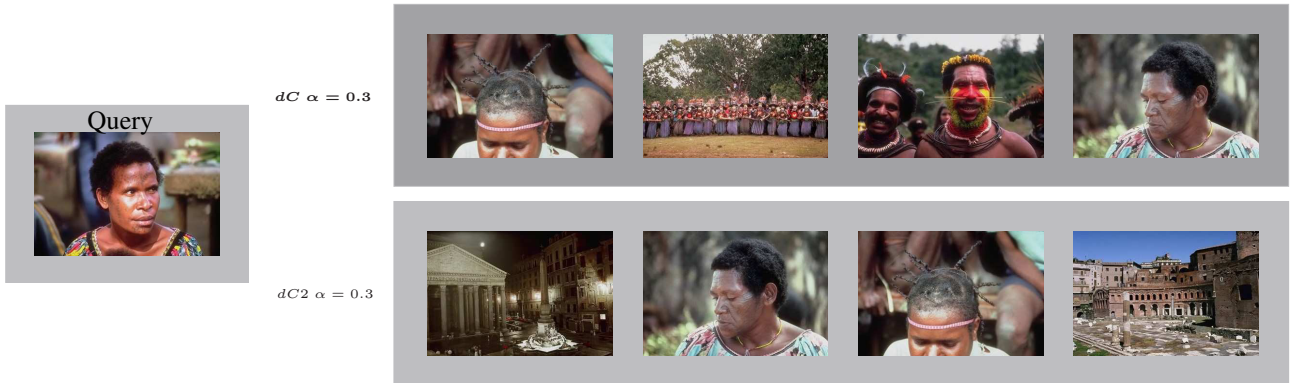


Figure 1.

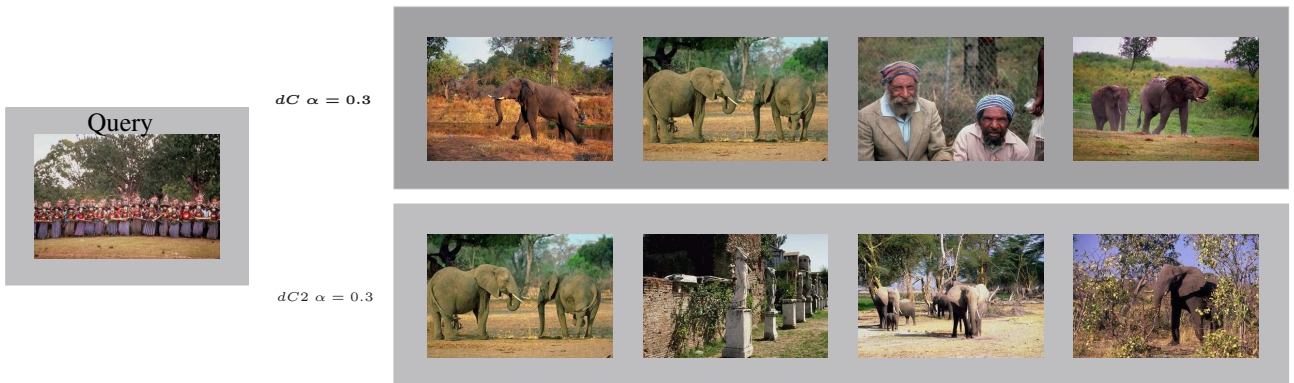
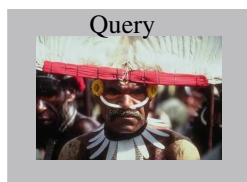


Figure 2.



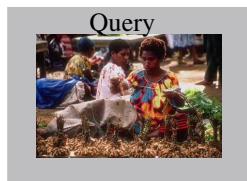
$dC \alpha = 0.3$



$dC2 \alpha = 0.3$



Figure 3.



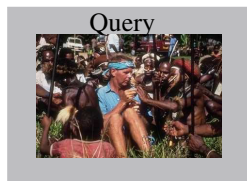
$dC \alpha = 0.3$



$dC2 \alpha = 0.3$



Figure 4.



$dC \alpha = 0.3$



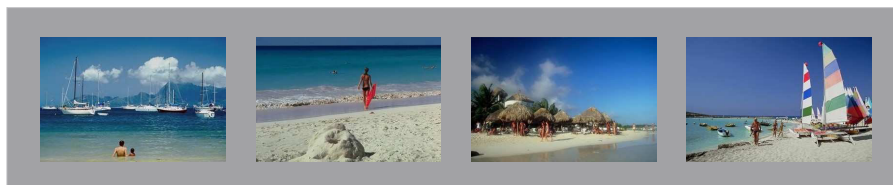
$dC2 \alpha = 0.3$



Figure 5.



$dC \alpha = 0.3$



$dC2 \alpha = 0.3$

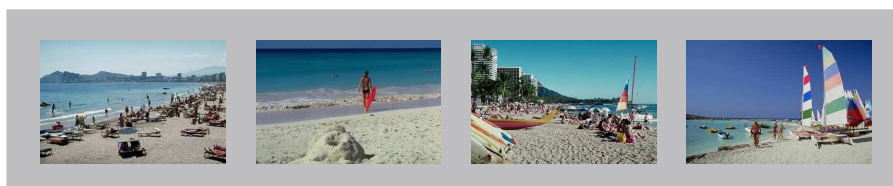
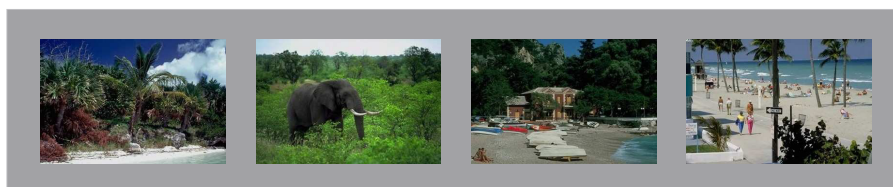


Figure 6.



$dC \alpha = 0.3$



$dC2 \alpha = 0.3$

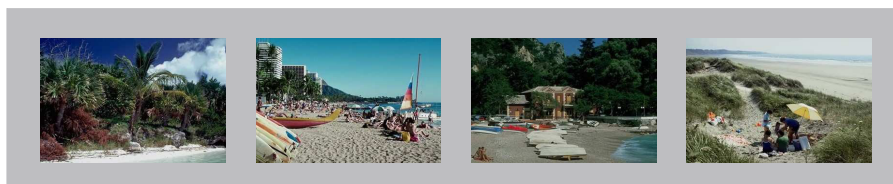
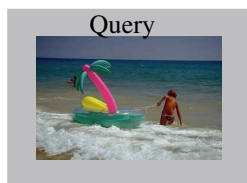


Figure 7.



$dC \alpha = 0.3$



$dC2 \alpha = 0.3$

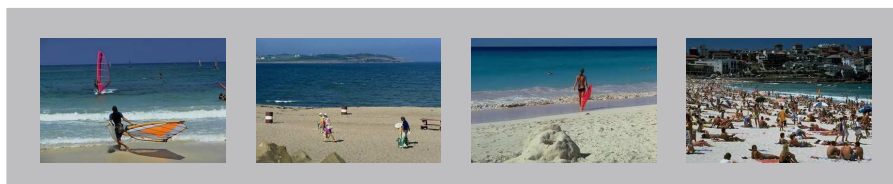
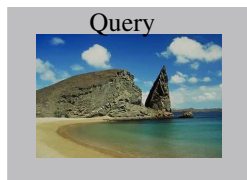


Figure 8.



$dC \alpha = 0.3$



$dC2 \alpha = 0.3$



Figure 9.



$dC \alpha = 0.3$



$dC2 \alpha = 0.3$



Figure 10.



$dC \alpha = 0.3$



$dC2 \alpha = 0.3$



Figure 11.



$dC \alpha = 0.3$



$dC2 \alpha = 0.3$



Figure 12.



$dC \alpha = 0.3$



$dC2 \alpha = 0.3$

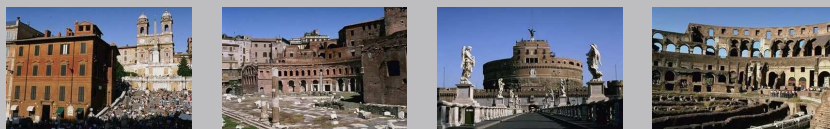


Figure 13.



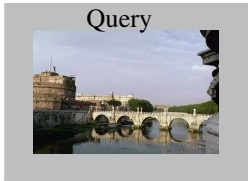
$dC \alpha = 0.3$



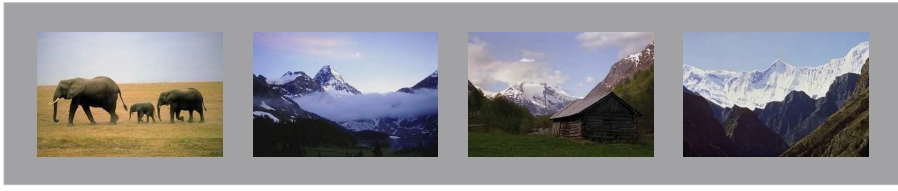
$dC2 \alpha = 0.3$



Figure 14.



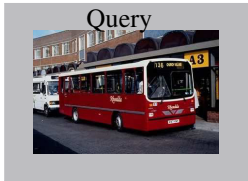
$dC \alpha = 0.3$



$dC2 \alpha = 0.3$



Figure 15.



$dC \alpha = 0.3$



$dC2 \alpha = 0.3$



Figure 16.



$dC \alpha = 0.3$



$dC2 \alpha = 0.3$



Figure 17.



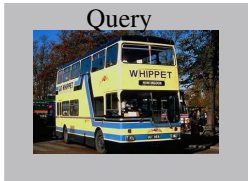
$dC \alpha = 0.3$



$dC2 \alpha = 0.3$



Figure 18.



$dC \alpha = 0.3$



$dC2 \alpha = 0.3$



Figure 19.



$dC \alpha = 0.3$



$dC2 \alpha = 0.3$



Figure 20.

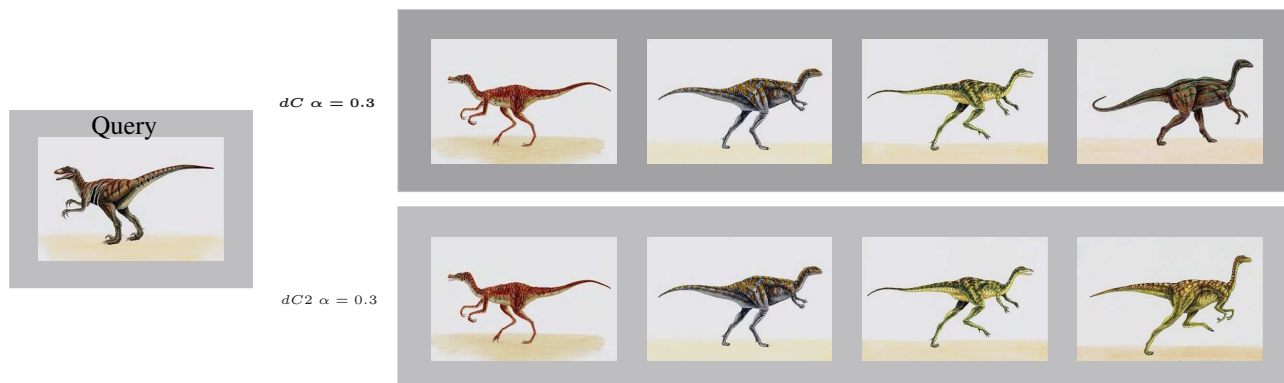


Figure 21.

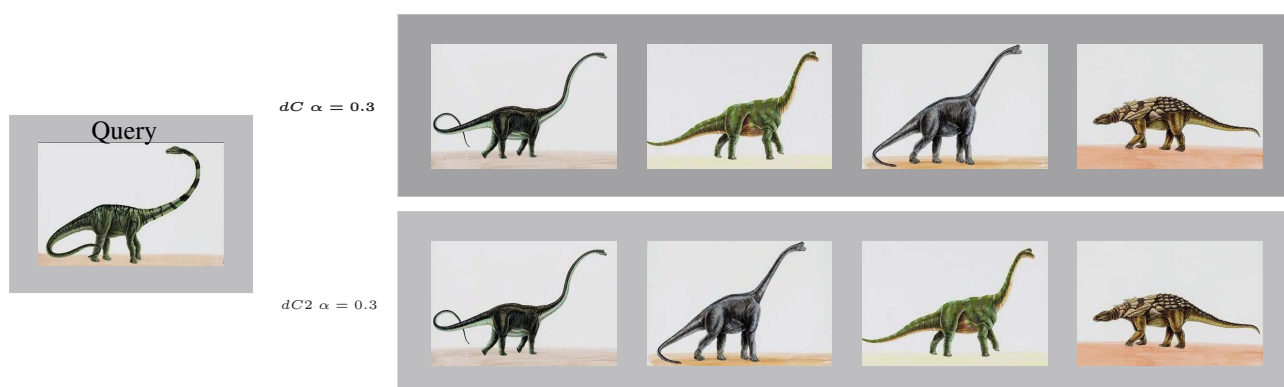


Figure 22.

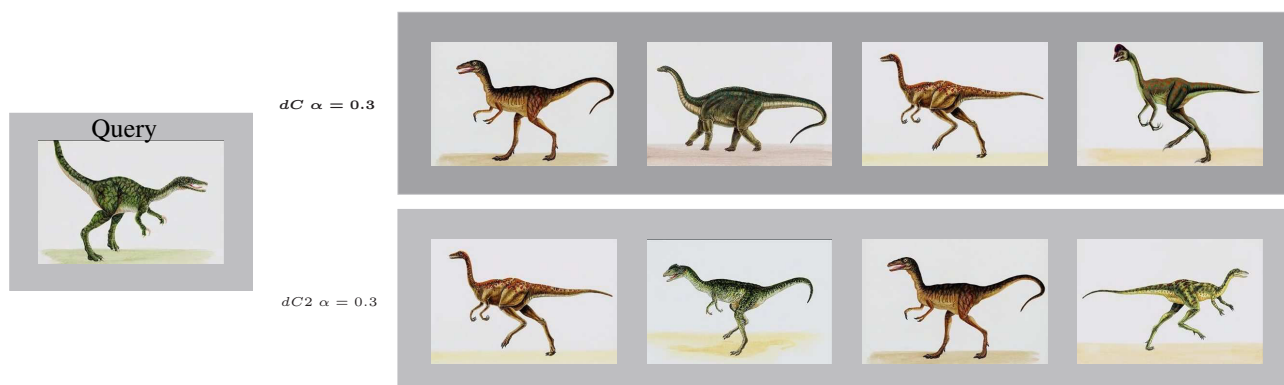


Figure 23.

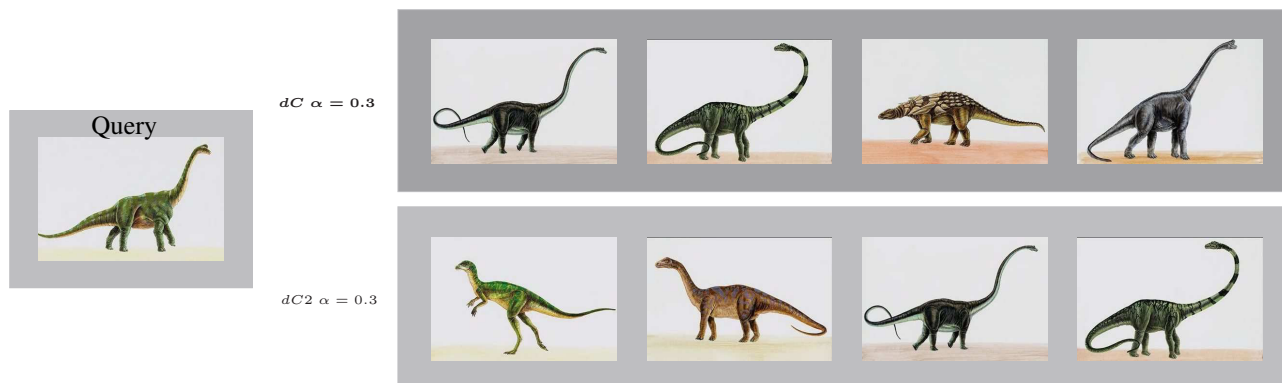


Figure 24.

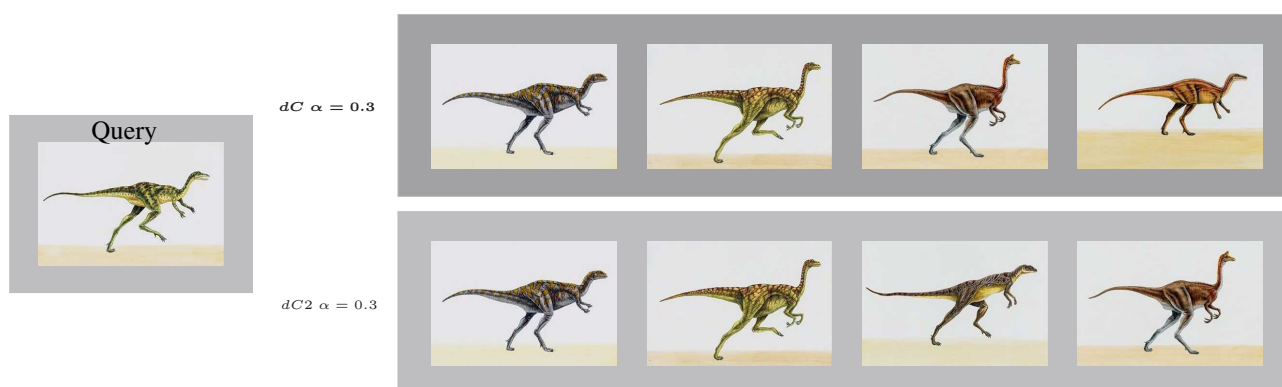


Figure 25.

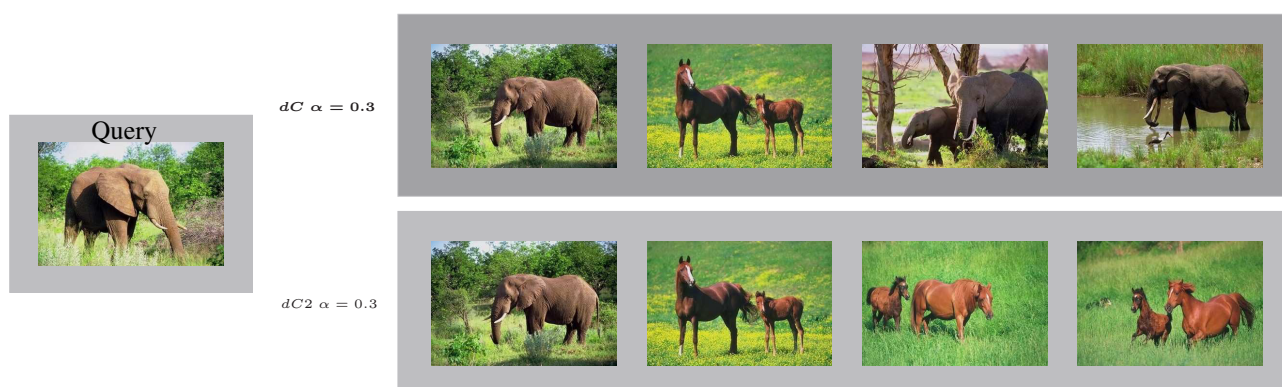
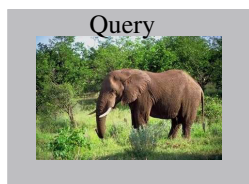


Figure 26.



$dC \alpha = 0.3$



$dC2 \alpha = 0.3$

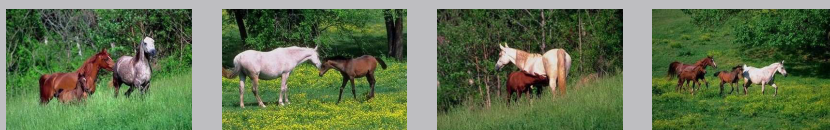
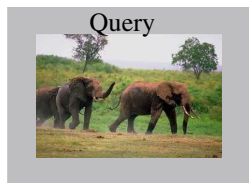


Figure 27.



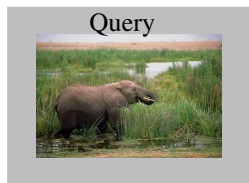
$dC \alpha = 0.3$



$dC2 \alpha = 0.3$



Figure 28.



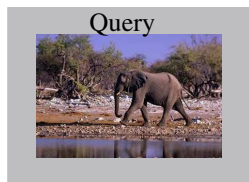
$dC \alpha = 0.3$



$dC2 \alpha = 0.3$



Figure 29.



$dC \alpha = 0.3$



$dC2 \alpha = 0.3$

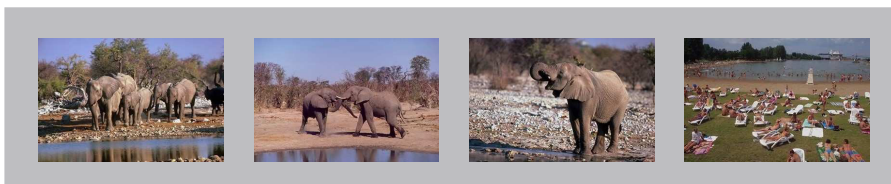
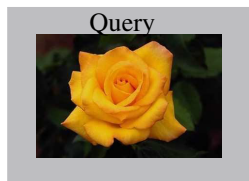


Figure 30.



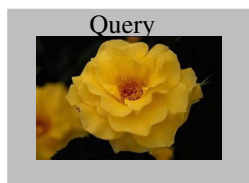
$dC \alpha = 0.3$



$dC2 \alpha = 0.3$



Figure 31.



$dC \alpha = 0.3$



$dC2 \alpha = 0.3$



Figure 32.



Figure 33.

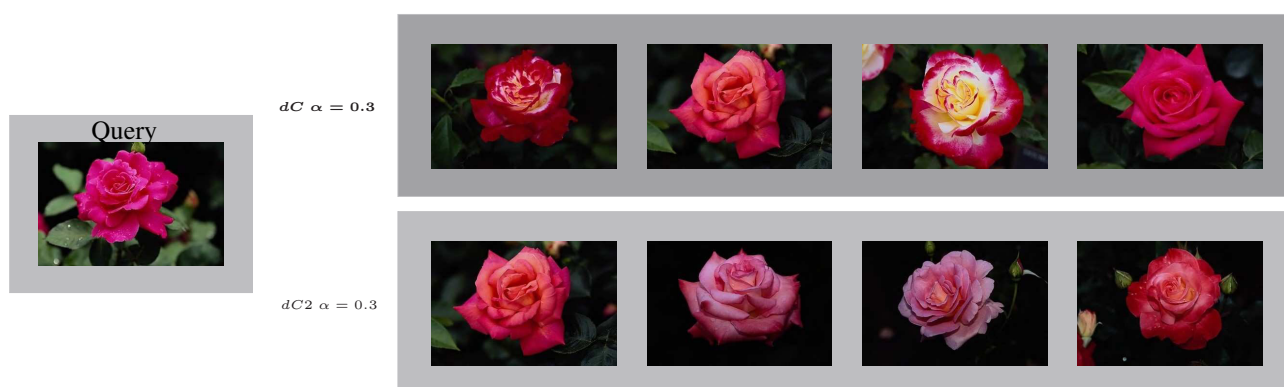
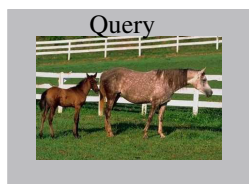


Figure 34.



Figure 35.



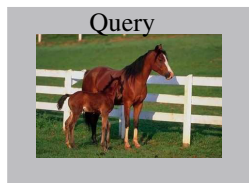
$dC \alpha = 0.3$



$dC2 \alpha = 0.3$



Figure 36.



$dC \alpha = 0.3$



$dC2 \alpha = 0.3$



Figure 37.



$dC \alpha = 0.3$



$dC2 \alpha = 0.3$

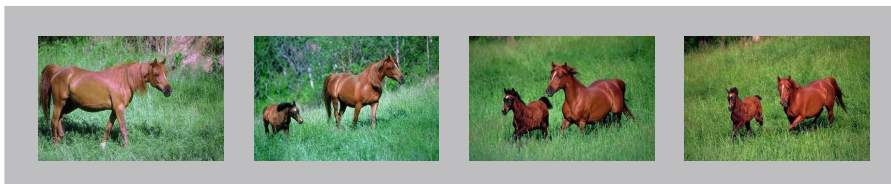
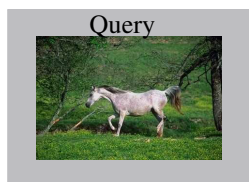


Figure 38.



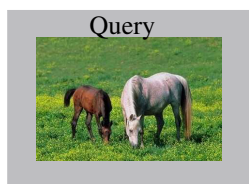
$dC \alpha = 0.3$



$dC2 \alpha = 0.3$



Figure 39.



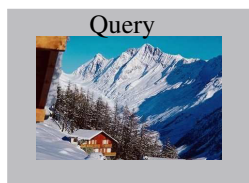
$dC \alpha = 0.3$



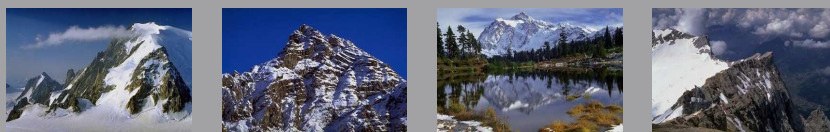
$dC2 \alpha = 0.3$



Figure 40.



$dC \alpha = 0.3$



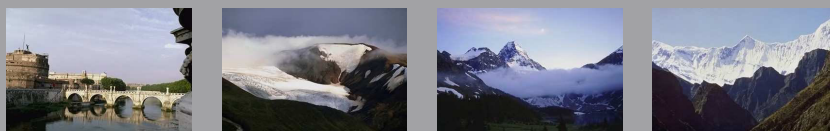
$dC2 \alpha = 0.3$



Figure 41.



$dC \alpha = 0.3$



$dC2 \alpha = 0.3$

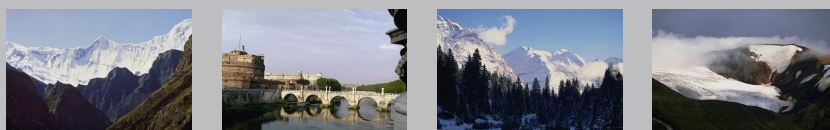


Figure 42.



$dC \alpha = 0.3$



$dC2 \alpha = 0.3$

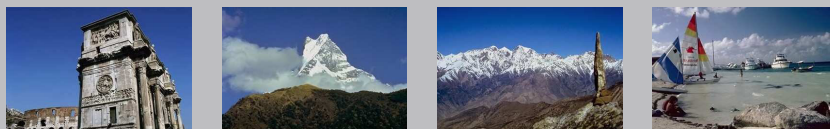
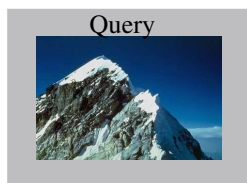


Figure 43.



$dC \alpha = 0.3$



$dC2 \alpha = 0.3$

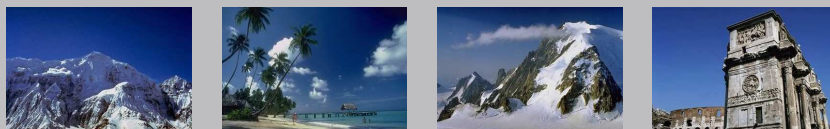
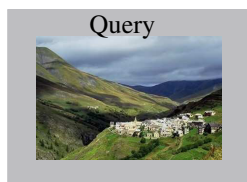


Figure 44.



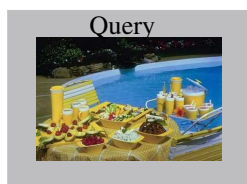
$dC \alpha = 0.3$



$dC2 \alpha = 0.3$



Figure 45.



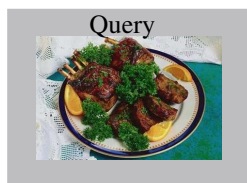
$dC \alpha = 0.3$



$dC2 \alpha = 0.3$



Figure 46.



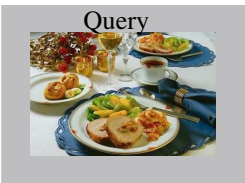
$dC \alpha = 0.3$



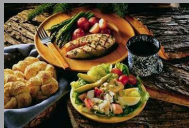
$dC2 \alpha = 0.3$



Figure 47.



$dC \alpha = 0.3$



$dC2 \alpha = 0.3$



Figure 48.



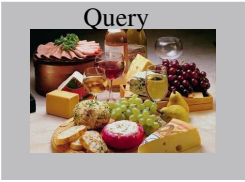
$dC \alpha = 0.3$



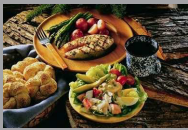
$dC2 \alpha = 0.3$



Figure 49.



$dC \alpha = 0.3$



$dC2 \alpha = 0.3$



Figure 50.

References

- [1] O. Pele and M. Werman. Fast and robust earth mover's distances. In *ICCV*, 2009. 1

Fast and Robust Earth Mover's Distances Additional Results - Comparing SIFT/CSIFT with All Distances

Ofir Pele

The Hebrew University of Jerusalem

ofirpele@cs.huji.ac.il

Michael Werman

The Hebrew University of Jerusalem

werman@cs.huji.ac.il

This document contains additional results for image retrieval that were omitted from the paper - “Fast and Robust Earth Mover’s Distances” [3] due to lack of space. The results are for image retrieval as described in the paper. The results are for all pairs of descriptors and distance measures.

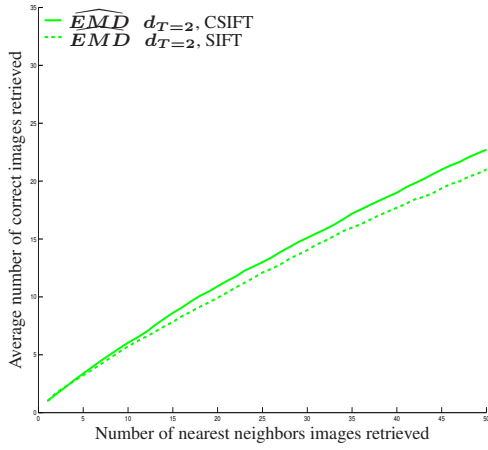


Figure 1.

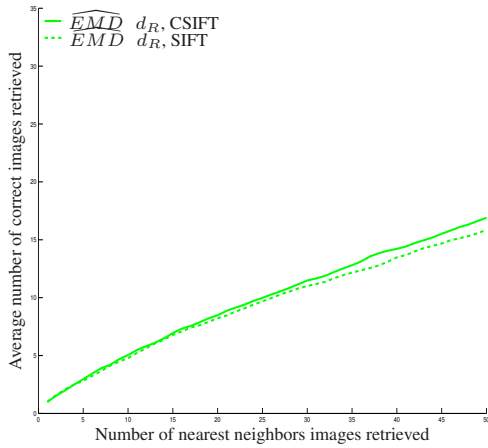


Figure 2.

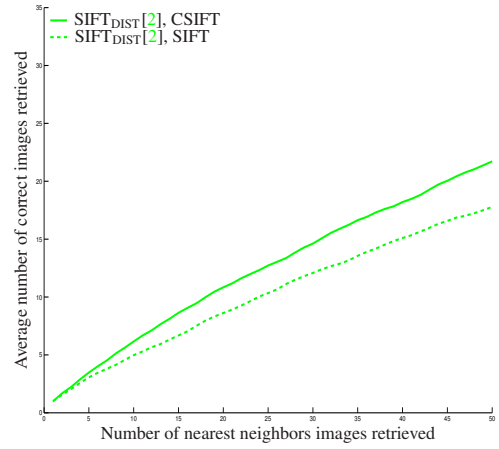


Figure 3.

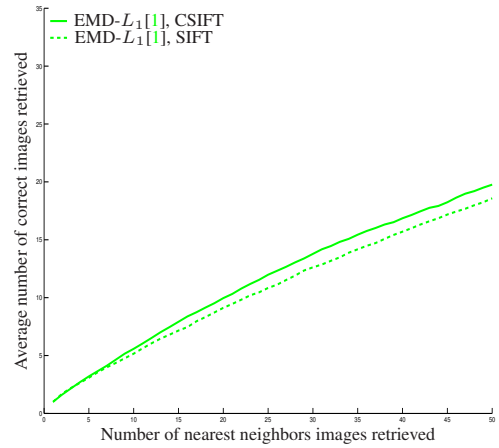


Figure 4.

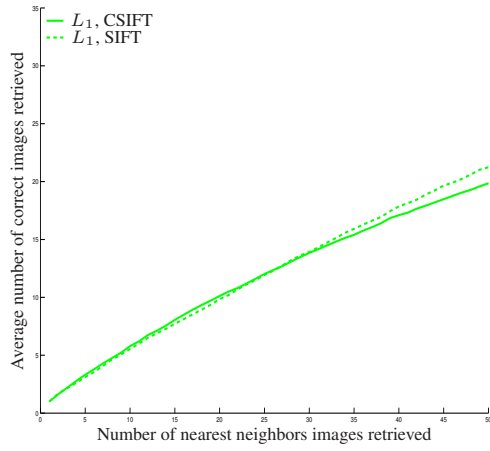


Figure 5.

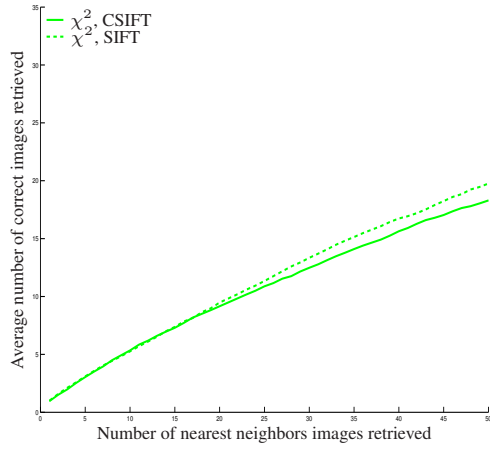


Figure 6.

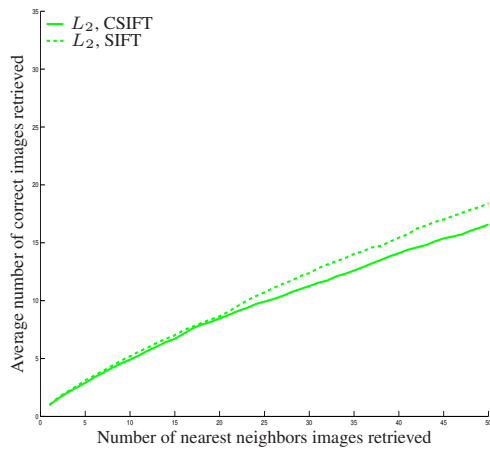


Figure 7.

References

- [1] H. Ling and K. Okada. An Efficient Earth Mover's Distance Algorithm for Robust Histogram Comparison. *PAMI*, 2007. [1](#)
- [2] O. Pele and M. Werman. A linear time histogram metric for improved sift matching. In *ECCV*, 2008. [1](#)
- [3] O. Pele and M. Werman. Fast and robust earth mover's distances. In *ICCV*, 2009. [1](#)

Chapter 3

The Quadratic-Chi Histogram Distance Family - Appendices

The Quadratic-Chi Histogram Distance Family - Appendices

Ofir Pele and Michael Werman

School of Computer Science
The Hebrew University of Jerusalem
{ofirpele,werman}@cs.huji.ac.il

1 Introduction

This document contains the appendices for the paper “The Quadratic-Chi Histogram Distance Family” [1], proofs and additional results. In section 2 we prove that all Quadratic-Chi histogram distances are continuous. In section 3 we prove that EMD, $\widehat{\text{EMD}}$ and all Quadratic-Chi histogram distances are *Similarity-Matrix-Quantization-Invariant*. In section 4 we present additional shape classification results. In section 5 we compare experimentally color image distances with and without spatial pruning. In section 6 we compare experimentally distances which resembles QC distances, but are either not *Similarity-Matrix-Quantization-Invariant* or not *Sparseness-Invariant*. In section 7 we compare the results for all the pairs of descriptors (SIFT/CSIFT) and distance measures used in the image retrieval experiments.

2 Quadratic-Chi Histogram Distance Continuity Proof

In this section we prove that a Quadratic-Chi (QC) histogram distance is continuous. We first recall its definition.

Let P and Q be two non-negative bounded histograms. That is, $P, Q \in [0, U]^N$, where the bound U can be any finite number. Let A be a non-negative symmetric bounded bin-similarity matrix such that each diagonal element is bigger or equal to every other element in its row (this demand is weaker than being a strongly dominant matrix) and to 1. That is, $A \in [0, U]^N \times [0, U]^N$ and $\forall i, j, A_{ii} \geq A_{ij}$ and $\forall i, A_{ii} \geq 1$. Let $0 \leq m < 1$ be the normalization factor. A Quadratic-Chi (QC) histogram distance is defined as:

$$\text{QC}_m^A(P, Q) = \sqrt{\sum_{ij} \left(\frac{(P_i - Q_i)}{(\sum_c (P_c + Q_c) A_{ci})^m} \right) \left(\frac{(P_j - Q_j)}{(\sum_c (P_c + Q_c) A_{cj})^m} \right) A_{ij}} \quad (1)$$

Theorem 1. *Each addend denominator: $(\sum_c (P_c + Q_c) A_{ci})^m (\sum_c (P_c + Q_c) A_{cj})^m$, is zero only if the addend numerator: $(P_i - Q_i)(P_j - Q_j) A_{ij}$, is zero.*

Proof.

$$\left(\sum_c (P_c + Q_c) A_{ci}\right)^m \left(\sum_c (P_c + Q_c) A_{cj}\right)^m = 0 \quad \Rightarrow \quad (2)$$

$$((P_i + Q_i) A_{ii})^m ((P_j + Q_j) A_{jj})^m = 0 \quad \Rightarrow \quad (3)$$

$$(P_i = 0 \wedge Q_i = 0) \vee (P_j = 0 \wedge Q_j = 0) \quad \Rightarrow \quad (4)$$

$$((P_i - Q_i) = 0) \vee ((P_j - Q_j) = 0) \quad \Rightarrow \quad (5)$$

$$(P_i - Q_i)(P_j - Q_j) A_{ij} = 0 \quad (6)$$

Eqs. 3, 4 and 6 are true as everything is non-negative and $A_{ii} \geq 1$.

We defined $\frac{0}{0} = 0$. So, in order to prove continuity, we need to prove that when QC addend's denominator tends to zero, the whole addend tends to zero (this is the only point where discontinuity can occur).

Theorem 2.

$$\begin{aligned} & \left(\sum_c (P_c + Q_c) A_{ci}\right)^m \left(\sum_c (P_c + Q_c) A_{cj}\right)^m \rightarrow 0 \Rightarrow \\ & \left(\frac{(P_i - Q_i)}{\left(\sum_c (P_c + Q_c) A_{ci}\right)^m}\right) \left(\frac{(P_j - Q_j)}{\left(\sum_c (P_c + Q_c) A_{cj}\right)^m}\right) A_{ij} \rightarrow 0 \end{aligned} \quad (7)$$

Proof.

$$\left(\sum_c (P_c + Q_c) A_{ci}\right)^m \left(\sum_c (P_c + Q_c) A_{cj}\right)^m \rightarrow 0 \quad \Rightarrow \quad (8)$$

$$((P_i + Q_i) A_{ii})^m ((P_j + Q_j) A_{jj})^m \rightarrow 0 \quad \Rightarrow \quad (9)$$

$$((P_i + Q_i)^m \rightarrow 0) \vee ((P_j + Q_j)^m \rightarrow 0) \quad \Rightarrow \quad (10)$$

$$((P_i \rightarrow 0) \wedge (Q_i \rightarrow 0)) \vee ((P_j \rightarrow 0) \wedge (Q_j \rightarrow 0)) \quad \Rightarrow \quad (11)$$

$$\left(\frac{(P_i - Q_i)}{((P_i + Q_i) A_{ii})^m}\right) \left(\frac{(P_j - Q_j)}{((P_j + Q_j) A_{jj})^m}\right) A_{ij} \rightarrow 0 \quad \Rightarrow \quad (12)$$

$$\left(\frac{(P_i - Q_i)}{\left(\sum_c (P_c + Q_c) A_{ci}\right)^m}\right) \left(\frac{(P_j - Q_j)}{\left(\sum_c (P_c + Q_c) A_{cj}\right)^m}\right) A_{ij} \rightarrow 0 \quad \Rightarrow \quad (13)$$

Eq. 9 is true as everything is non-negative and $m \geq 0$. Eq. 10 is true as if the whole product tends to zero, at least one of its multiplicands should tend to zero and $A_{ii} \geq 1$ and $A_{jj} \geq 1$. Eq. 11 is true as everything is non-negative and $m \geq 0$. Eq. 12 is true as $|(P_i - Q_i)| \leq |(P_i + Q_i)| A_{ii}$ and $|(P_j - Q_j)| \leq |(P_j + Q_j)| A_{jj}$ and $0 \leq m < 1$ (note that the strictly smaller than one is required here) and everything is bounded. Eq. 13 finishes the proof and is true as:

$$\left|\left(\frac{(P_i - Q_i)}{\left(\sum_c (P_c + Q_c) A_{ci}\right)^m}\right)\right| < \left|\left(\frac{(P_i - Q_i)}{((P_i + Q_i) A_{ii})^m}\right)\right| \wedge \quad (14)$$

$$\left|\left(\frac{(P_j - Q_j)}{\left(\sum_c (P_c + Q_c) A_{cj}\right)^m}\right)\right| < \left|\left(\frac{(P_j - Q_j)}{((P_j + Q_j) A_{jj})^m}\right)\right| \quad (15)$$

3 EMD, $\widehat{\text{EMD}}$ and Quadratic-Chi Histogram Distances Similarity-Matrix-Quantization-Invariance Proof

In this section we prove that EMD, $\widehat{\text{EMD}}$ and all Quadratic-Chi histogram distances are *Similarity-Matrix-Quantization-Invariant*. We recall *Similarity-Matrix-Quantization-Invariance* definition:

Definition 1. Let $\mathcal{D}$ be a cross-bin histogram distance between two histograms P and Q and let A be the bin-similarity/distance matrix. We assume P , Q and A are non-negative and that A is symmetric. Let $A_{k,:}$ be the k th row of A . Let $V = [V_1, \dots, V_N]$ be a non-negative vector and $0 \leq \alpha \leq 1$. We define $V^{\alpha,k,b} = [\dots, \alpha V_k, \dots, V_b + (1 - \alpha)V_k, \dots]$. That is, $V^{\alpha,k,b}$ is a transformation of V where $(1 - \alpha)V_k$ mass has moved from bin k to bin b . We define $\mathcal{D}$ to be *Similarity-Matrix-Quantization-Invariant* if:

$$A_{k,:} = A_{b,:} \Rightarrow \forall 0 \leq \alpha \leq 1, 0 \leq \beta \leq 1 \quad \mathcal{D}^A(P, Q) = \mathcal{D}^A(P^{\alpha,k,b}, Q^{\beta,k,b}) \quad (16)$$

If $\mathcal{D}$ is symmetric then Eq. 16 is equivalent to:

$$A_{k,:} = A_{b,:} \Rightarrow \forall 0 \leq \alpha \leq 1 \quad \mathcal{D}^A(P, Q) = \mathcal{D}^A(P^{\alpha,k,b}, Q) \quad (17)$$

EMD [2] and $\widehat{\text{EMD}}$ [3] distances are *Similarity-Matrix-Quantization-Invariant*. Since both are min-cost-flow problems [4], taking mass from one source/sink and moving it to another one, where both sources/sinks are connected to exactly the same sinks/sources, with exactly the same costs, will not change the min-cost solution.

We begin the proof that a Quadratic-Chi (QC) histogram distance (Eq. 1) is *Similarity-Matrix-Quantization-Invariant* with a lemma:

$$\left(\sum_c (P_c^{\alpha,k,b} + Q_c) A_{ci} \right)^m = \left(\sum_c (P_c + Q_c) A_{ci} \right)^m \quad (18)$$

Proof.

$$\sum_c (P_c^{\alpha,k,b} + Q_c) A_{ci} = \quad (19)$$

$$\left(\sum_{c \neq k,b} (P_c^{\alpha,k,b} + Q_c) A_{ci} \right) + ((P_k^{\alpha,k,b} + Q_k) A_{ki}) + ((P_b^{\alpha,k,b} + Q_b) A_{bi}) = \quad (20)$$

$$\left(\sum_{c \neq k,b} (P_c + Q_c) A_{ci} \right) + ((\alpha P_k + Q_k) A_{ki}) + ((P_b + (1 - \alpha)P_k + Q_b) A_{bi}) = \quad (21)$$

$$\left(\sum_{c \neq k,b} (P_c + Q_c) A_{ci} \right) + ((\alpha P_k + Q_k) A_{ki}) + ((P_b + (1 - \alpha)P_k + Q_b) A_{ki}) = \quad (22)$$

$$\left(\sum_{c \neq k,b} (P_c + Q_c) A_{ci} \right) + ((P_k + Q_k + P_b + Q_b) A_{ki}) = \quad (23)$$

$$\left(\sum_{c \neq k,b} (P_c + Q_c) A_{ci} \right) + ((P_k + Q_k) A_{ki}) + ((P_b + Q_b) A_{bi}) = \quad (24)$$

$$\sum_c (P_c + Q_c) A_{ci} \quad (25)$$

$$\Downarrow \quad (26)$$

$$\left(\sum_c (P_c^{\alpha,k,b} + Q_c) A_{ci} \right)^m = \left(\sum_c (P_c + Q_c) A_{ci} \right)^m \quad (27)$$

In the proof we use the definitions and in Eqs. 22,24 we use $A_{ki} = A_{bi}$.

We now prove that Quadratic-Chi histogram distance is *Similarity-Matrix-Quantization-Invariant* using Eq. 17. That is we prove that:

$$A_{k,:} = A_{b,:} \Rightarrow \forall 0 \leq \alpha \leq 1 \quad QC_m^A(P, Q) = QC_m^A(P^{\alpha,k,b}, Q) \quad (28)$$

We will prove that $(QC_m^A(P, Q))^2 = (QC_m^A(P^{\alpha,k,b}, Q))^2$ which directly implies Eq. 28.

We start by separating $(QC_m^A(P^{\alpha,k,b}, Q))^2$ addends into three disjoint and complementary types. The first type is when both i and j are different from k and b . For these addends the equality is direct from the definition. The second type is when both i and j are either k or b . For this type the proof is:

$$\sum_{i=(k \vee b), j=(k \vee b)} \frac{(P_i^{\alpha,k,b} - Q_i)(P_j^{\alpha,k,b} - Q_j)A_{ij}}{(\sum_c (P_c^{\alpha,k,b} + Q_c)A_{ci})^m (\sum_c (P_c^{\alpha,k,b} + Q_c)A_{cj})^m} = \quad (29)$$

$$\sum_{i=(k \vee b), j=(k \vee b)} \frac{(P_i^{\alpha,k,b} - Q_i)(P_j^{\alpha,k,b} - Q_j)A_{ij}}{(\sum_c (P_c + Q_c)A_{ci})^m (\sum_c (P_c + Q_c)A_{cj})^m} = \quad (30)$$

$$\begin{aligned} & \frac{(P_k^{\alpha,k,b} - Q_k)(P_k^{\alpha,k,b} - Q_k)A_{kk}}{(\sum_c (P_c + Q_c)A_{ck})^{2m}} + \frac{(P_k^{\alpha,k,b} - Q_k)(P_b^{\alpha,k,b} - Q_b)A_{kb}}{(\sum_c (P_c + Q_c)A_{ck})^m (\sum_c (P_c + Q_c)A_{cb})^m} + \\ & \frac{(P_b^{\alpha,k,b} - Q_b)(P_k^{\alpha,k,b} - Q_k)A_{bk}}{(\sum_c (P_c + Q_c)A_{cb})^m (\sum_c (P_c + Q_c)A_{ck})^m} + \frac{(P_b^{\alpha,k,b} - Q_b)(P_b^{\alpha,k,b} - Q_b)A_{bb}}{(\sum_c (P_c + Q_c)A_{cb})^{2m}} = \end{aligned} \quad (31)$$

$$\frac{\left((\alpha P_k - Q_k)^2 + 2(\alpha P_k - Q_k)((P_b + (1 - \alpha)P_k) - Q_b) + ((P_b + (1 - \alpha)P_k) - Q_b)^2 \right) A_{kk}}{(\sum_c (P_c + Q_c)A_{ck})^{2m}} = \quad (32)$$

$$\frac{\left((\alpha P_k - Q_k) + ((P_b + (1 - \alpha)P_k) - Q_b) \right)^2 A_{kk}}{(\sum_c (P_c + Q_c)A_{ck})^{2m}} = \quad (33)$$

$$\frac{\left((P_k - Q_k) + (P_b - Q_b) \right)^2 A_{kk}}{(\sum_c (P_c + Q_c)A_{ck})^{2m}} = \quad (34)$$

$$\frac{\left((P_k - Q_k)^2 + 2(P_k - Q_k)(P_b - Q_b) + (P_b - Q_b)^2 \right) A_{kk}}{(\sum_c (P_c + Q_c)A_{ck})^{2m}} = \quad (35)$$

$$\sum_{i=(k \vee b), j=(k \vee b)} \frac{(P_i - Q_i)(P_j - Q_j)A_{ij}}{(\sum_c (P_c + Q_c)A_{ci})^m (\sum_c (P_c + Q_c)A_{cj})^m} \quad (36)$$

Eq. 30 is based on the lemma in Eq. 18. Eqs. 32,36 are true as $A_{ck} = A_{cb}$ and $A_{kk} = A_{kb} = A_{bb} = A_{bk}$.

The third type of addends is when only i equals b or k or only j equals b or k . These sub-cases are symmetric, so it is enough to prove equality for only one of them:

$$\sum_{i=(k \vee b), j \neq (k \vee b)} \frac{(P_i^{\alpha, k, b} - Q_i)(P_j^{\alpha, k, b} - Q_j)A_{ij}}{(\sum_c (P_c^{\alpha, k, b} + Q_c)A_{ci})^m (\sum_c (P_c^{\alpha, k, b} + Q_c)A_{cj})^m} = \quad (37)$$

$$\sum_{i=(k \vee b), j \neq (k \vee b)} \frac{(P_i^{\alpha, k, b} - Q_i)(P_j - Q_j)A_{ij}}{(\sum_c (P_c + Q_c)A_{ci})^m (\sum_c (P_c + Q_c)A_{cj})^m} = \quad (38)$$

$$\sum_{j \neq (k \vee b)} \left(\frac{(P_k^{\alpha, k, b} - Q_k)(P_j - Q_j)A_{kj}}{(\sum_c (P_c + Q_c)A_{ck})^m (\sum_c (P_c + Q_c)A_{cj})^m} + \frac{(P_b^{\alpha, k, b} - Q_b)(P_j - Q_j)A_{bj}}{(\sum_c (P_c + Q_c)A_{cb})^m (\sum_c (P_c + Q_c)A_{cj})^m} \right) = \quad (39)$$

$$\sum_{j \neq (k \vee b)} \left(\frac{(\alpha P_k - Q_k + (P_b + (1 - \alpha)P_k) - Q_b)(P_j - Q_j)A_{kj}}{(\sum_c (P_c + Q_c)A_{ck})^m (\sum_c (P_c + Q_c)A_{cj})^m} \right) = \quad (40)$$

$$\sum_{j \neq (k \vee b)} \left(\frac{((P_k - Q_k) + (P_b - Q_b))(P_j - Q_j)A_{kj}}{(\sum_c (P_c + Q_c)A_{ck})^m (\sum_c (P_c + Q_c)A_{cj})^m} \right) = \quad (41)$$

$$\sum_{i=(k \vee b), j \neq (k \vee b)} \frac{(P_i - Q_i)(P_j - Q_j)A_{ij}}{(\sum_c (P_c + Q_c)A_{ci})^m (\sum_c (P_c + Q_c)A_{cj})^m} \quad (42)$$

Eq. 38 is based on the lemma in Eq. 18. Eqs. 40,42 are true as $A_{kj} = A_{bj}$ and $A_{ck} = A_{cb}$.

Since we covered the three disjoint and complementary cases we have proved that a Quadratic-Chi histogram distance is *Similarity-Matrix-Quantization-Invariant*.

4 Additional Shape Classification Results

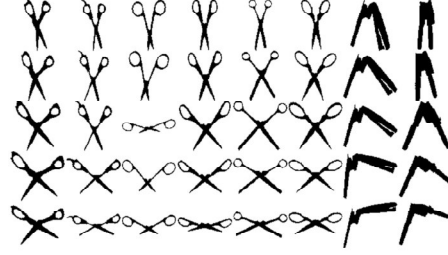
This section presents shape classification results using the same framework as Ling et al. [5,6,7]. We first present results for the articulated shape data set [5,7]. Then we present results for the Kimia-216 data set [8].

4.1 Articulated Data Set

The articulated shape data set [5,7] has 40 images from 8 different objects. Each object has 5 images articulated to different degrees, see Fig. 1(a). The dataset is very challenging because of the similarity between different objects (especially the scissors).

We used Belongie et al.'s Shape Context (SC) [9] and Ling and Jacobs' Inner Distance Shape Context (IDSC) [5].

To evaluate results, for each image, the 4 most similar matches are chosen from other images in the dataset. The retrieval result is summarized as the number of 1st,



(a)

Results using SC[9]						Results using IDSC[5]					
	Top 1	Top 2	Top 3	Top 4	AUC%		Top 1	Top 2	Top 3	Top 4	AUC%
$QCN^{1-\frac{dscT=2}{2}}$	20	6	10	5	0.307	$QCN^{1-\frac{dscT=2}{2}}$	39	38	38	34	0.950
QCN^I	18	7	5	8	0.278	QCN^I	40	37	36	33	0.940
$QCS^{1-\frac{dscT=2}{2}}$	23	11	11	4	0.378	$QCS^{1-\frac{dscT=2}{2}}$	39	35	38	28	0.912
QCS^I	19	9	9	6	0.318	QCS^I	40	34	37	27	0.907
χ^2	25	13	13	7	0.430	χ^2	40	36	36	21	0.902
$QF^{1-\frac{dscT=2}{2}}$	25	16	10	7	0.438	$QF^{1-\frac{dscT=2}{2}}$	40	34	39	19	0.897
L_2	25	15	8	7	0.420	L_2	39	35	35	18	0.873
$\widehat{EMD}_1^{dscT=2}$	23	13	11	7	0.400	$\widehat{EMD}_1^{dscT=2}$	39	36	35	27	0.902
L_1	20	10	9	5	0.333	L_1	39	35	35	25	0.890
$SIFT_{DIST}[10]$	20	9	6	9	0.320	$SIFT_{DIST}[10]$	38	37	27	22	0.848
$EMD-L_1[6]$	15	10	6	10	0.280	$EMD-L_1[6]$	39	35	38	30	0.917
$Diffusion[11]$	19	10	7	6	0.315	$Diffusion[11]$	39	35	34	23	0.880
$Bhattacharyya[12]$	25	14	9	9	0.422	$Bhattacharyya[12]$	40	37	32	23	0.895
$KL[13]$	12	8	4	9	0.223	$KL[13]$	40	38	36	29	0.938
$JS[14]$	25	16	9	7	0.432	$JS[14]$	40	35	37	21	0.900

(b)

(c)

Fig. 1. (a) Articulated shape database. This dataset contains 40 images from 8 objects. Each column contains five images of the same object with different articulation.

(b) Shape classification results using Belongie et al.’s Shape Context (SC)[9]. All distances performance is poor.

(c) Shape classification results using Ling and Jacobs’ Inner Distance Shape Context (IDSC)[5]. $QCN^{1-\frac{dscT=2}{2}}$ outperformed all other distances. QCN^I performance was excellent, which shows the importance of the normalization factor. All cross-bin distances outperformed their bin-by-bin versions. Again, χ^2 and QF improved upon L_2 . QCN and QCS which are mathematically sound combinations of χ^2 and QF outperformed both.

2nd, 3rd and 4th most similar matches that come from the correct object. Tables (b) and (c) in Fig. 1 show the retrieval results for SC and IDSC respectively. Using SC descriptors, the performance was poor on this dataset for all kinds of distances. Using IDSC descriptors, the $QCN^{1-\frac{dsc_{T=2}}{2}}$ histogram distance outperformed all the other distances. QCN^f performance was excellent, which shows the importance of the normalization factor. All cross-bin distances outperformed their bin-by-bin versions. χ^2 and QF improved upon L_2 . QCN and QCS which are mathematically sound combinations of χ^2 and QF outperformed both.

4.2 Kimia-216 Data Set

The Kimia-216 shape data set [8] has 216 images grouped into 18 classes with 12 images in each class, see Fig. 2(a).

We tested for shape classification with Belongie et al. Shape Context (SC) [9]. We do not present results for Ling and Jacob’s IDSC [5], as the code that computes IDSC [7] crashes on this data set (segmentation fault).

To evaluate results, for each image, the 11 most similar matches are chosen from other images in the dataset (as there are 12 images in each class). The retrieval result is summarized as the number of 1st, 2nd, ..., 11th and most similar matches that come from the correct object. Table (b) in Fig. 2(b) show the retrieval results. Again, the $QCN^{1-\frac{dsc_{T=2}}{2}}$ histogram distance outperformed all the other distances. Again, χ^2 and QF improved upon L_2 and QCN and QCS outperformed both.

5 Comparison of Color Image Distances With and Without Spatial Pruning

In this section we compare experimentally the color image distances that have spatial pruning (those used in [1]) to distances without such a pruning (those used in [3]). Results are presented in Fig. 3. Pruning slightly improved accuracy for QF and slightly reduced accuracy for QCN, QCS and $\widehat{EMD}$. However, pruning considerably reduced running time (see Table 1 in page 15).

6 Comparison to *Non-Similarity-Matrix-Quantization-Invariant* and *Non-Sparseness-Invariant* Quadratic-Chi-like Histogram Distances

In this section we compare experimentally distances which resembles QC distances, but are either not *Similarity-Matrix-Quantization-Invariant* or not *Sparseness-Invariant*. The comparison shows that these properties considerably boost performance, especially for sparse color histograms.

As in the *QC* definition, let P and Q be two non-negative bounded histograms. That is, $P, Q \in [0, U]^N$. Let A be a non-negative symmetric bounded bin-similarity matrix such that each diagonal element is bigger or equal to every other element in its



(a)

Results using SC[9]

	Top 1	Top 2	Top 3	Top 4	Top 5	Top 6	Top 7	Top 8	Top 9	Top 10	Top 11	AUC%
$QCN^{1-\frac{dsc_T=2}{2}}$	215	216	215	216	214	214	207	207	200	192	173	0.981
QCN^I	215	208	212	201	198	198	190	177	164	150	127	0.920
$QCS^{1-\frac{dsc_T=2}{2}}$	216	215	215	212	212	205	204	205	193	184	155	0.969
QCS^I	215	215	214	214	215	208	209	205	197	190	167	0.976
χ^2	216	215	212	211	210	203	199	199	192	177	156	0.960
$QF^{1-\frac{dsc_T=2}{2}}$	214	211	207	206	197	190	183	183	167	162	146	0.920
L_2	213	210	203	204	195	189	183	178	167	149	139	0.910
$\widehat{EMD}_1^{dsc_T=2}$	215	215	215	215	214	207	207	202	201	187	162	0.974
L_1	215	215	215	214	213	209	211	207	199	192	170	0.978
SIFT _{DIST} [10]	215	215	215	215	215	211	210	207	201	189	162	0.979
EMD- L_1 [6]	215	215	212	213	212	208	204	203	197	184	161	0.969
Diffusion[11]	215	215	214	214	214	214	208	204	200	196	174	0.979
Bhattacharyya[12]	216	215	213	212	212	206	203	206	193	177	159	0.967
KL[13]	212	203	200	199	195	186	187	179	166	151	130	0.899
JS[14]	216	215	212	210	208	205	200	197	189	182	152	0.959

(b)

Fig. 2. (a) Kimia-216 shape database. This dataset has 216 images from 18 classes. Each row contains 12 images of the same class.

(b) Shape classification results using the Shape Context (SC)[9]. We do not present results for Ling and Jacob's IDSC [5], as the code that computes IDSC [7] crashed (segmentation fault). Again, the $QCN^{1-\frac{dsc_T=2}{2}}$ histogram distance outperformed all other distances. Again, χ^2 and QF improved upon L_2 . QCN and QCS which are mathematically sound combinations of χ^2 and QF outperformed both.

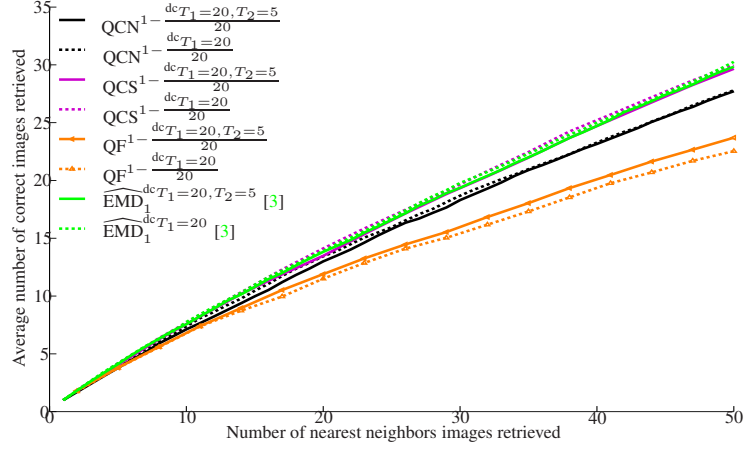


Fig. 3. Results for image retrieval comparing distances which are pruning spatially far away pixels (painted with solid lines) to those that are not (painted with broken lines). Pruning slightly improved accuracy for QF and slightly reduced accuracy for QCN, QCS and $\widehat{\text{EMD}}$. However, pruning considerably reduced running time (see Table 1 in page 15).

row (this demand is weaker than being a strongly dominant matrix) and to 1. That is, $A \in [0, U]^N \times [0, U]^N$ and $\forall i, j, A_{ii} \geq A_{ij}$ and $\forall i, A_{ii} \geq 1$. Let $0 \leq m < 1$ be the normalization factor. A QC-like histogram distance which is not *Sparseness-Invariant* is defined as:

$$\text{NSI-QC}_m^A(P, Q) = \sqrt{\sum_{ij} \left(\frac{(\sum_c P_c A_{ci} - \sum_c Q_c A_{ci})}{(\sum_c (P_c + Q_c) A_{ci})^m} \right) \left(\frac{(\sum_c P_c A_{cj} - \sum_c Q_c A_{cj})}{(\sum_c (P_c + Q_c) A_{cj})^m} \right) A_{ij}} \quad (43)$$

A QC-like histogram distance which is not *Similarity-Matrix-Quantization-Invariant* is defined as:

$$\text{NQI-QC}_m^A(P, Q) = \sqrt{\sum_{ij} \left(\frac{(P_i - Q_i)}{(P_i + Q_i)^m} \right) \left(\frac{(P_j - Q_j)}{(P_j + Q_j)^m} \right) A_{ij}} \quad (44)$$

If I is the identity matrix then:

$$\text{QC}_m^I(P, Q) = \text{NSI-QC}_m^I(P, Q) = \text{NQI-QC}_m^I(P, Q) \quad (45)$$

For any bin-similarity A we get the following equality:

$$\text{NQI-QC}_m^I(AP, AQ) = \text{NSI-QC}_m^I(P, Q) \quad (46)$$

We present results for image retrieval comparing QCN, NSI-QCN(NSI-QC_{0.9}), NQI-QCN(NQI-QC_{0.9}), QCS, NSI-QCS(NSI-QC_{0.5}) and NQI-QCS(NQI-QC_{0.5}) in Fig. 4(a) for SIFT-like descriptors (CSIFT as the CSIFT descriptor, compared to SIFT was better for all distance measures, see Fig. 7) and in Fig. 4(b) for small $L \times a \times b$ images with the pruning distance used in the paper [1] and in 4(c) for small $L \times a \times b$ images with a non-pruning distance as was used in [3]. The results show that *Similarity-Matrix-Quantization-Invariance* and *Sparseness-Invariance* considerably boost performance. The gain in performance is large using the $L \times a \times b$ images, which are sparse 5-dimensional histograms (x, y, L^*, a^*, b^*). That is, the query histogram was always: $[1, \dots, 1, 0, \dots, 0]$ and each image being compared to the query was represented by the histogram: $[0, \dots, 0, 1, \dots, 1]$. That is, the dimension of the space is $32 \times 48 \times 256^3$ where only 32×48 entries are non-zero in each histograms. The bin-similarity matrix is computed for each pair of images. For these histograms:

$$\text{NQI-QCN}^A(P, Q) = \text{NQI-QCS}^A(P, Q) = \text{QF}^A(P, Q) \quad (47)$$

The time complexity of all the distances tested in this section is linear. In practice (see Table 1), QCS distances are faster to compute than QCN distances. In addition, NQI-QC distances are the fastest to compute, QC distances second, and NSI-QC distances have the longest running time. However, the differences between the running times are minor as all distances have linear time complexity.

7 Comparing SIFT/CSIFT with All Distances for Image Retrieval

In this section we present results for image retrieval as described in the paper. The results are for all pairs of descriptors (SIFT/CSIFT) and distance measures and given in Figs. 5, 6, 7.

References

1. Pele, O., Werman, M.: The quadratic-chi histogram distance family. In: ECCV. (2010) 1, 7, 10
2. Rubner, Y., Tomasi, C., Guibas, L.J.: The earth mover's distance as a metric for image retrieval. IJCV (2000) 3
3. Pele, O., Werman, M.: Fast and robust earth mover's distances. In: ICCV. (2009) 3, 7, 9, 10, 13
4. Ahuja, R., Magnanti, T., Orlin, J.: Network flows: theory, algorithms, and applications. (1993) 3
5. Ling, H., Jacobs, D.: Shape classification using the inner-distance. PAMI (2007) 5, 6, 7, 8
6. Ling, H., Okada, K.: An Efficient Earth Mover's Distance Algorithm for Robust Histogram Comparison. PAMI (2007) 5, 6, 8, 13
7. Ling, H.: Articulated shape benchmark and idsc code. <http://www.ist.temple.edu/~hbling/code/inner-dist-articu-distribution.zip> (2010) 5, 7, 8
8. Sebastian, T., Kimia, B.: Curves vs. skeletons in object recognition. SP (2005) 5, 7
9. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. PAMI (2002) 5, 6, 7, 8

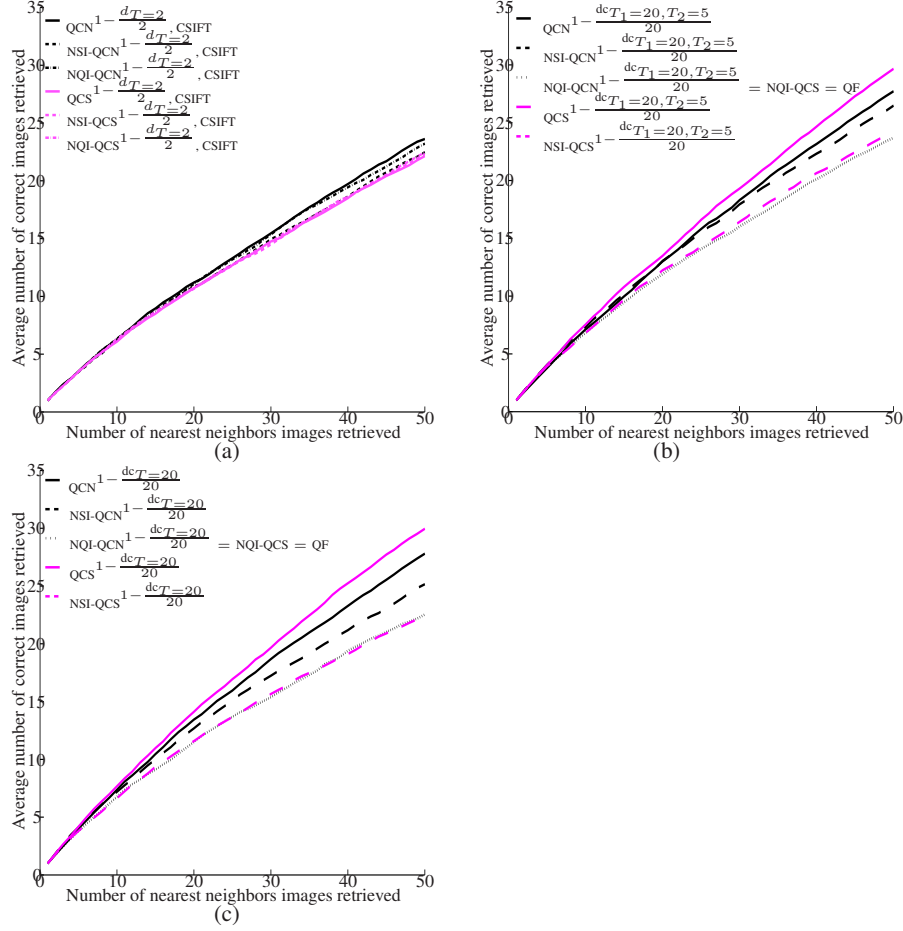
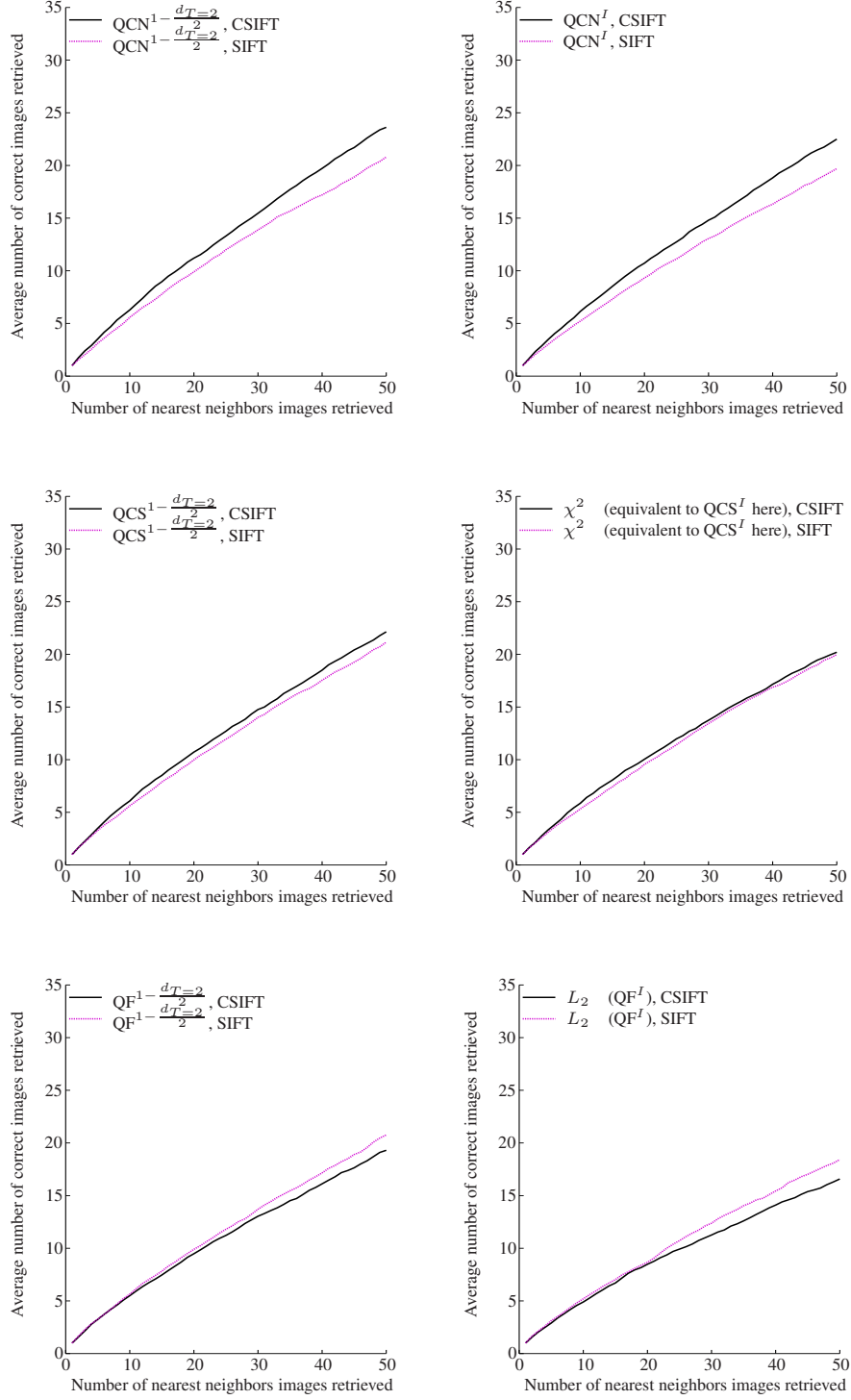
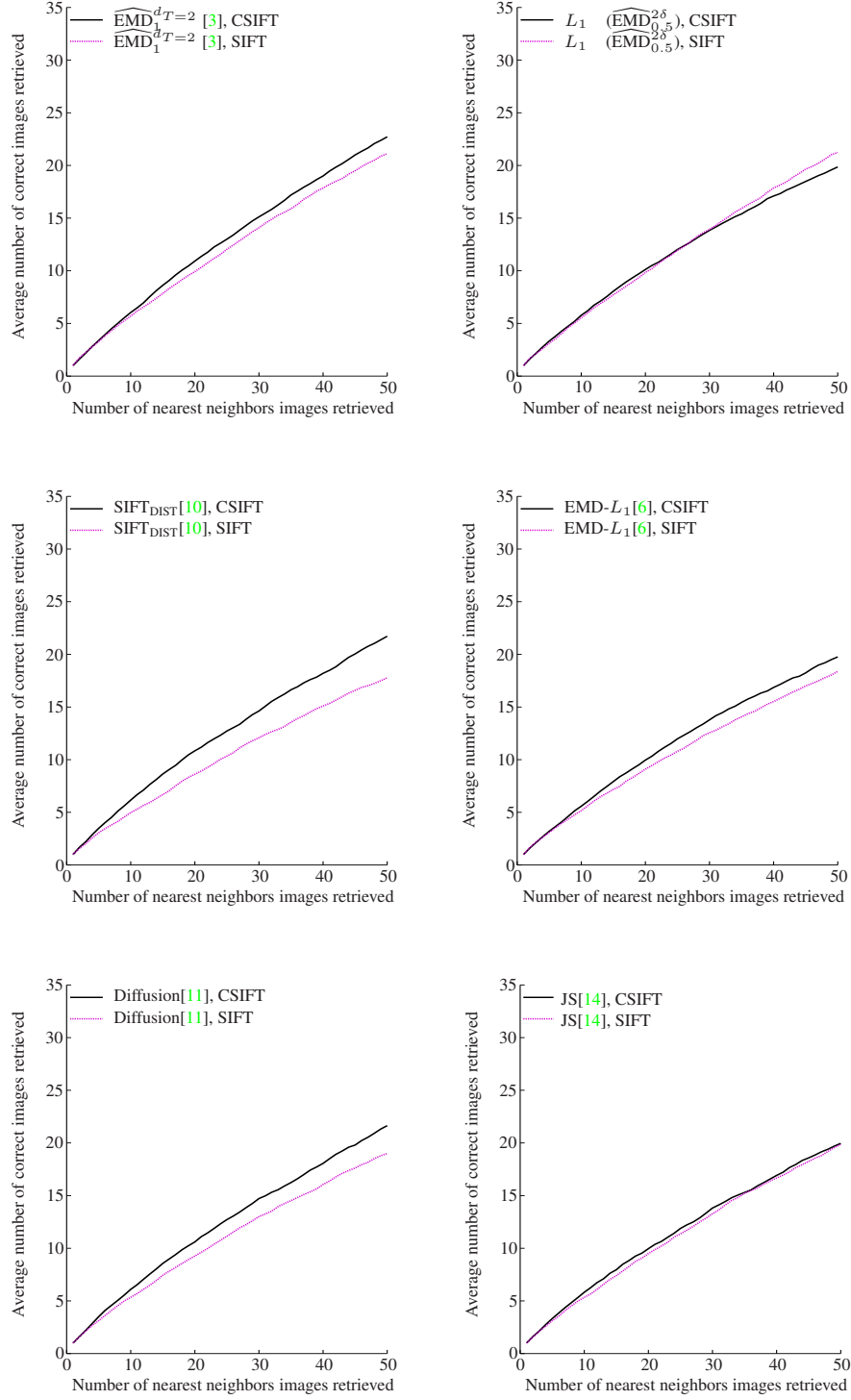


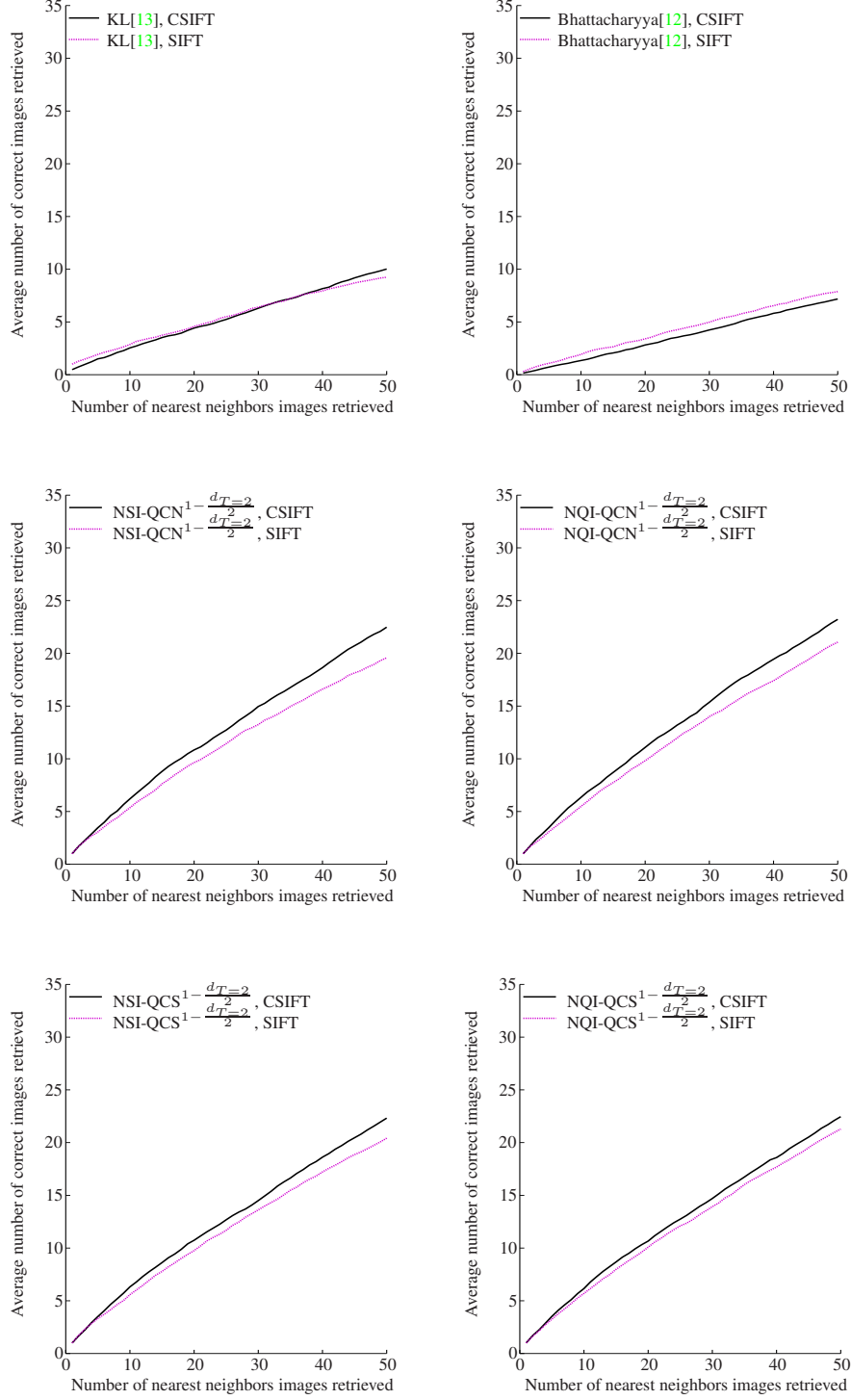
Fig. 4. Results for image retrieval comparing QC which is *Sparseness-Invariant* and *Similarity-Matrix-Quantization-Invariant* to NSI-QC and to NQI-QC which are not *Sparseness-Invariant* and not *Similarity-Matrix-Quantization-Invariant* respectively.

(a) **SIFT-like descriptors.** For each distance measure, we present the descriptor (SIFT/CSIFT) with which it performed best. The results for all the pairs of descriptors and distance measures can be found in Figs. 5, 6 and 7. It can be seen that QCN which is both *Sparseness-Invariant* and *Similarity-Matrix-Quantization-Invariant* has the best performance. However, the differences in performance are small.

(b) **L*a*b* images with pruning and (c) without pruning** Here, the *Sparseness-Invariant* and *Similarity-Matrix-Quantization-Invariant* properties are very important. This is probably because L*a*b* images are represented as very sparse histograms in a very high dimensional space (the dimension of the space is $32 \times 48 \times 256^3$ where only 32×48 entries are non-zero in each histograms). Comparing to (b) we can also note that performance using this representation is better than those obtained with the SIFT-like descriptor.

**Fig. 5.** Comparison of SIFT with CSIFT for all distances, part 1.

**Fig. 6.** Comparison of SIFT with CSIFT for all distances, part 2

**Fig. 7.** Comparison of SIFT with CSIFT for all distances, part 3

Descriptor	QCN ^{A2}	NSI-QCN ^{A2}	NQI-QCN ^{A2}	QCS ^{A2}	NSI-QCS ^{A2}	NQI-QCS ^{A2}
(SIFT)	0.15	0.18	0.14	0.07	0.1	0.06

Descriptor	QCN ^{A20,5}	NSI-QCN ^{A20,5}	QF ^{A20,5}	QCS ^{A20,5}	NSI-QCS ^{A20,5}
(L*a*b*)	20 (370)	21 (371)	11 (361)	19 (369)	18 (368)

Descriptor	QCN ^{A20}	NSI-QCN ^{A20}	QF ^{A20}	QCS ^{A20}	NSI-QCS ^{A20}
(L*a*b*)	20 (3020)	21 (3021)	11 (3011)	19 (3019)	18 (3018)

Table 1. (SIFT) 384-dimensional SIFT-like descriptors matching time (in *milliseconds*). The distances from left to right are the same as the distances in Fig. 4 (a) from up to down.

(L*a*b*) 32×48 L*a*b* images matching time (in *milliseconds*) using a distance which prunes spatially far away pixels (middle row) and a distance which does not (bottom row). On the middle row, the distances from left to right are the same as the distances in Fig. 4 (b). On the bottom row, the distances from left to right are the same as the distances in Fig. 4 (c). In parentheses is the time it takes to compute the distance and the bin-similarity matrix as it cannot be computed offline.

To summaries results, QCS distances are faster to compute than QCN distances. Also, NQI-QC distances are the fastest to compute, QC distances second, and NSI-QC distances have the longest running time. However, the differences between the running times are minor as all distances have linear time complexity. Finally, pruning spatially far away pixels considerably reduced running time, almost without reducing accuracy (see Fig. 3).

10. Pele, O., Werman, M.: A linear time histogram metric for improved sift matching. In: ECCV. (2008) **6, 8, 13**
11. Ling, H., Okada, K.: Diffusion distance for histogram comparison. In: CVPR. (2006) **6, 8, 13**
12. Bhattacharyya, A.: On a measure of divergence between two statistical populations defined by their probability distributions. BCMS (1943) **6, 8, 14**
13. Kullback, S., Leibler, R.: On information and sufficiency. AMS (1951) **6, 8, 14**
14. Lin, J.: Divergence measures based on the Shannon entropy. IT (1991) **6, 8, 13**

Machine Learning for Image Classification and Clustering Using a Universal Distance Measure

Uzi Chester and Joel Ratsaby*

Electrical and Electronics Engineering Department,
Ariel University of Samaria, ARIEL 40700

ratsaby@ariel.ac.il

<http://www.ariel.ac.il/sites/ratsaby/>

Abstract. We present a new method for image feature-extraction which is based on representing an image by a finite-dimensional vector of distances that measure how different the image is from a set of image prototypes. We use the recently introduced Universal Image Distance (UID) [1] to compare the similarity between an image and a prototype image. The advantage in using the UID is the fact that no domain knowledge nor any image analysis need to be done. Each image is represented by a finite dimensional feature vector whose components are the UID values between the image and a finite set of image prototypes from each of the feature categories. The method is automatic since once the user selects the prototype images, the feature vectors are automatically calculated without the need to do any image analysis. The prototype images can be of different size, in particular, different than the image size. Based on a collection of such cases any supervised or unsupervised learning algorithm can be used to train and produce an image classifier or image cluster analysis. In this paper we present the image feature-extraction method and use it on several supervised and unsupervised learning experiments for satellite image data. The feature-extraction method is scalable and is easily implementable on multi-core computing resources.

1 Introduction

Image classification research aims at finding representations of images that can be automatically used to categorize images into a finite set of classes. Typically, algorithms that classify images require some form of pre-processing of an image prior to classification. This process may involve extracting relevant features and segmenting images into sub-components based on some prior knowledge about their context [2,3].

In [1] we introduced a new distance function, called Universal Image Distance (UID), for measuring the distance between two images. The UID first transforms each of the two images into a string of characters from a finite alphabet and then uses the string distance of [4] to give the distance value between the images. According to [4] the distance between two strings x and y is a normalized

* Corresponding author.

difference between the complexity of the concatenation xy of the strings and the minimal complexity of each of x and y . By complexity of a string x we mean the Lempel-Ziv complexity [5].

In the current paper we use the UID to create a finite-dimensional representation of an image. The i^{th} component of this vector is a feature that measures how different the image is from the i^{th} feature category. One of the advantages of the UID is that it can compare the distance between two images of different sizes and thus the prototypes which are representative of the different feature categories may be relatively small. For instance, the prototypes of an *urban* category can be small images of size 45×70 pixels of various parts of cities.

In this paper we introduce a process to convert the image into a labeled case (feature vector). Doing this systematically for a set of images each labeled by its class yields a data set which can be used for training any supervised and unsupervised learning algorithms. After describing our method in details we report on the accuracy results of several classification-learning algorithms on such data. As an example, we apply our method to satellite image classification and clustering.

We note that our process for converting an image into a finite dimensional feature vector is very straightforward and does not involve any domain knowledge about the images. In contrast to other image classification algorithms that extract features based on sophisticated mathematical analysis, such as, analyzing the texture, the special properties of an image, doing edge-detection, or any of the many other methods employed in the immense research-literature on image processing, our approach is very basic and universal. It is based on the complexity of the 'raw' string-representation of an image. Our method extracts features automatically just by computing distances from a set of prototypes. It is therefore scalable and can be implemented using parallel processing techniques, such as on system-on-chip and FPGA hardware implementation [6,7,8].

Our method extracts image features that are unbiased in the sense that they do not employ any heuristics in contrast to other common image-processing techniques [2]. The features that we extract are based on information implicit in the image and obtained via a complexity-based UID distance which is an information-theoretic measure. In our method, the feature vector representation of an image is based on the distance of the image from some fixed set of representative class-prototypes that are initially and only once picked by a human user running the learning algorithm.

Let us now summarize the organization of the paper: in section 2 we review the definitions of LZ-complexity and a few string distances. In section 3 we define the UID distance. In section 4 we describe the algorithm for selecting class prototypes. In section 5 we describe the algorithm that generates a feature-vector representation of an image. In section 6 we discuss the classification learning method and in section we conclude by reporting on the classification accuracy results.

2 LZ-Complexity and String Distances

The UID distance function [1] is based on the LZ-complexity of a string. The definition of this complexity follows [4, 5]: let S, Q and R be strings of characters that are defined over the alphabet $\mathcal{A}$. Denote by $l(S)$ the length of S , and $S(i)$ denotes the i^{th} element of S . We denote by $S(i, j)$ the substring of S which consists of characters of S between position i and j (inclusive). An extension $R = SQ$ of S is reproducible from S (denoted as $S \rightarrow R$) if there exists an integer $p \leq l(S)$ such that $Q(k) = R(p+k-1)$ for $k = 1, \dots, l(Q)$. For example, $aacgt \rightarrow aacgtcgtcg$ with $p = 3$ and $aacgt \rightarrow aacgtac$ with $p = 2$. R is obtained from S (the seed) by first copying all of S and then copying in a sequential manner $l(Q)$ elements starting at the p^{th} location of S in order to obtain the Q part of R .

A string S is *producible* from its prefix $S(1, j)$ (denoted $S(1, j) \Rightarrow R$), if $S(1, j) \rightarrow S(1, l(S) - 1)$. For example, $aacgt \rightarrow aacgtac$ and $aacgt \rightarrow aacgtacc$ both with pointers $p = 2$. The production adds an extra 'different' character at the end of the copying process which is not permitted in a reproduction.

Any string S can be built using a *production process* where at its i^{th} step we have the production $S(1, h_{i-1}) \Rightarrow S(1, h_i)$ where h_i is the location of a character at the i^{th} step. (Note that $S(1, 0) \Rightarrow S(1, 1)$).

An m -step production process of S results in parsing of S in which $H(S) = S(1, h_1) \cdot S(h_1 + 1, h_2) \cdots S(h_{m-1} + 1, h_m)$ is called the *history* of S and $H_i(S) = S(h_{i-1} + 1, h_i)$ is called the i^{th} component of $H(S)$. For example for $S = aacgtacc$ we have $H(S) = a \cdot ac \cdot g \cdot t \cdot acc$ as the history of S .

If $S(1, h_i)$ is not reproducible from $S(1, h_{i-1})$ then the component $H_i(S)$ is called *exhaustive* meaning that the copying process cannot be continued and the component should be halted with a single character *innovation*. Every string S has a unique exhaustive history [5].

Let us denote by $c_H(S)$ the number of components in a history of S . The LZ complexity of S is $c(S) = \min \{c_H(S)\}$ where the minimum is over all histories of S . It can be shown that $c(S) = c_E(S)$ where $c_E(S)$ is the number of components in the exhaustive history of S .

A distance for strings based on the LZ-complexity was introduced in [4] and is defined as follows: given two strings X and Y , denote by XY their concatenation then define

$$d(X, Y) := \max \{c(XY) - c(X), c(YX) - c(Y)\}.$$

In [1] we have found that the following normalized distance

$$d(X, Y) := \frac{c(XY) - \min \{c(X), c(Y)\}}{\max \{c(X), c(Y)\}}. \quad (1)$$

performs well in classification and clustering of images.

We note in passing that (1) resembles the normalized compression distance of [9] except that here we do not use a compressor but instead resort to the LZ-complexity c of a string. Note that d is not a metric since it does not satisfy

the triangle inequality and a distance of 0 implies that the two strings are close but not necessarily identical. We refer to $\mathbf{d}$ as a universal distance because it is not dependent on some specific representation of a string nor on heuristics common to other string distances such as edit-distances [10]. D only depends on the LZ-complexity of each of the two strings and their concatenation and this is a pure information-quantity which depends on the string's context but not its representation.

3 Universal Image Distance

Based on $\mathbf{d}$ we now define a distance between images. The idea is to convert each of two images I and J into strings $X^{(I)}$ and $X^{(J)}$ of characters from a finite alphabet of symbols. Once in string format, we use $\mathbf{d}(X^{(I)}, X^{(J)})$ as the distance between I and J . The details of this process are described in Algorithm 1 below.

Algorithm 1. UID distance measure

1. **Input:** two color images I, J in jpeg format (RGB representation)
 2. Transform the RGB matrices into gray-scale by forming a weighted sum of the R, G, and B components according to the following formula: $grayscaleValue := 0.2989R + 0.5870G + 0.1140B$, (used in Matlab®). Each pixel is now a single numeric value in the range of 0 to 255. We refer to this set of values as the alphabet and denote it by $\mathcal{A}$.
 3. Scan each of the grayscale images from top left to bottom right and form a string of symbols from $\mathcal{A}$. Denote the two strings by $X^{(I)}$ and $X^{(J)}$.
 4. Compute the LZ-complexities: $c(X^{(I)})$, $c(X^{(J)})$ and the complexity of their concatenation $c(X^{(I)}X^{(J)})$
 5. **Output:** $UID(I, J) := \mathbf{d}(X^{(I)}, X^{(J)})$.
-

Remark 1. The transformation into gray-scale is a matter of representational convenience. To deal with color images without this transformation one can create a 3D alphabet whereby each 'letter' in this alphabet corresponds to an RGB triple with each component in the range 0 to 255. This way the image color information remains in the string representation.

4 Prototype Selection

In this section we describe the algorithm for selecting image prototypes from each of the feature categories. This process runs only once before the stage of converting the images into finite dimensional vectors, that is, it does not run

once per image but once for all images. For an image I we denote by $P \subset I$ a sub-image P of I where P can be any rectangular-image obtained by placing a window over the image I where the window is totally enclosed by I . The size of the window depends on how much information a single prototype will convey about the associated feature-category.

In the following algorithm we use clustering as a simple means of validation that the prototypes selected maintain the inherent differences between the feature-categories (the clustering algorithm is not given the feature-category information but only the inter-prototype distance information).

Algorithm 2. Prototypes selection

1. **Input:** M image feature categories, and a corpus $\mathcal{C}_N$ of N unlabeled colored images $\{I_j\}_{j=1}^N$.
 2. **for** ($i := 1$ **to** M) **do**
 - (a) Based on *any* of the images I_j in $\mathcal{C}_N$, let the user **select** L_i prototype images $\{P_k^{(i)}\}_{k=1}^{L_i}$ and set them as feature category i . Each prototype is contained by some image, $P_k^{(i)} \subset I_j$, and the size of $P_k^{(i)}$ can vary, in particular it can be much smaller than the size of the images I_j , $1 \leq j \leq N$.
 - (b) **end for**;
 3. **Enumerate** all the prototypes into a single *unlabeled* set $\{P_k\}_{k=1}^L$, where $L = \sum_{i=1}^M L_i$ and calculate the distance matrix $H = \left[UID(X^{(P_k)}, X^{(P_l)})\right]_{k=1, l=1}^L$ where the (k, l) component of H is the UID distance between the unlabeled prototypes P_k and P_l .
 4. **Run** hierarchical clustering on H and obtain the associated dendrogram (note: H does not contain any 'labeled' information about feature-categories, as it is based on the unlabeled set).
 5. **If** there are M clusters with the i^{th} cluster consisting of the prototypes $\{P_k^{(i)}\}_{k=1}^{L_i}$ **then** terminate and **go to** step 7.
 6. **Else go to** step 2.
 7. **Output:** the set of labeled prototypes $\mathcal{P}_L := \left\{ \left\{ P_k^{(i)} \right\}_{k=1}^{L_i} \right\}_{i=1}^M$ where L is the number of prototypes.
-

From the theory of learning pattern recognition, it is known that the dimensionality M of a feature-vector is usually taken to be small compared to the data size N . A large L will obtain better feature representation accuracy of the image, but it will increase the time for running Algorithm 3 (described below).

Algorithm 2 convergence is based on the user's ability to select good prototype images. We note that from our experiments this is easily achieved primarily because the UID permits to select prototypes $P_k^{(i)}$ which are considerably *smaller* than the size of the full images I_j . For instance, in our experiments we used 45×70

pixels prototype size for all feature categories. This fact makes it easy for a user to quickly choose typical representative prototypes from every feature-category. This way it is easy to find informative prototypes, that is, prototypes that are distant when they are from different feature-categories and close when they are from the same feature category. Thus Algorithm 2 typically converges rapidly.

As an example, Figure 1 displays 12 prototypes selected by a user from a corpus of satellite images. The user labeled prototypes 1, . . . , 3 as representative of the feature category *urban*, prototypes 4, . . . , 6 as representatives of class *sea*, prototypes 7, . . . , 9 as representative of feature *roads* and prototypes 10, . . . , 12 as representative of feature *arid*. The user easily found these representative prototypes as it is easy to fit in a single picture of size 45×70 pixels a typical image. The dendrogram produced in step 4 of Algorithm 2 for these set of 12 prototypes is displayed in Figure 2. It is seen that the following four clusters were found $\{10, 12, 11\}$, $\{1, 2, 3\}$, $\{7, 8, 9\}$, $\{4, 6, 5\}$ which indicates that the prototypes selected in Algorithm 2 are good.

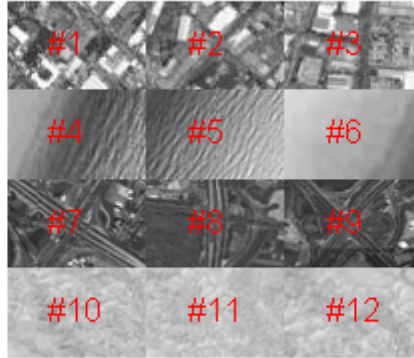


Fig. 1. Labeled prototypes of feature-categories *urban*, *sea*, *roads*, and *arid* (each feature has three prototypes, starting from top left and moving right in sequence)

5 Image Feature-Representation

In the previous section we described Algorithm 2 by which the prototypes are manually selected. This algorithm is now used to create a feature-vector representation of an image. It is described as Algorithm 3 below (in 1 we used a similar algorithm UIC to soft-classify an image whilst here we use it to only produce a feature vector representation of an image which later serves as a single labeled case for training any supervised learning algorithm or a single unlabeled case for training an unsupervised algorithm).

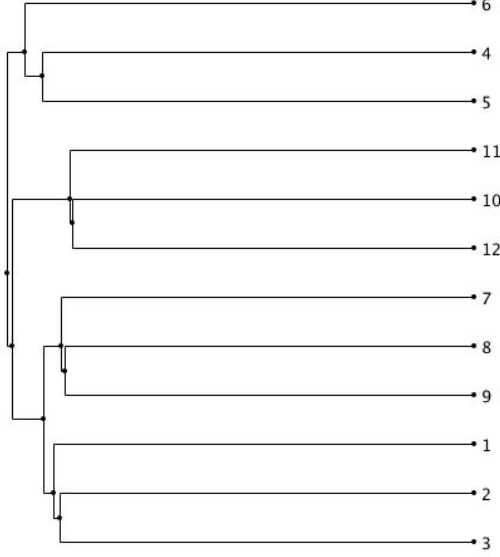


Fig. 2. Dendrogram of prototypes of Figure 1

Algorithm 3. Feature-vector generation

1. **Input:** an image I to be represented on the following feature categories $1 \leq i \leq M$, and given a set $\mathcal{P}_L := \left\{ \left\{ P_k^{(i)} \right\}_{k=1}^{L_i} \right\}_{i=1}^M$ of labeled prototype images (obtained from Algorithm 2).
 2. **Initialize** the count variables $c_i := 0, 1 \leq i \leq M$
 3. Let W be a rectangle of size equal to the maximum prototype size. (See remark below)
 4. Scan a window W across I from top-left to bottom-right in a non-overlapping way, and let the sequence of obtained sub-images of I be denoted as $\{I_j\}_{j=1}^m$.
 5. **for** ($j := 1$ to m) **do**
 - (a) **for** ($i := 1$ to M) **do**
 - i. $temp := 0$
 - ii. **for** ($k := 1$ to L_i) **do**
 - A. $temp := temp + \left(UID(I_j, P_k^{(i)}) \right)^2$
 - B. **end for**;
 - iii. $r_i := \sqrt{temp}$
 - iv. **end for**;
 - (b) Let $i^*(j) := \operatorname{argmin}_{1 \leq i \leq M} r_i$, this is the decided feature category for sub-image I_j .
 - (c) **Increment** the count, $c_{i^*(j)} := c_{i^*(j)} + 1$
 - (d) **end for**;
 6. **Normalize** the counts, $v_i := \frac{c_i}{\sum_{l=1}^M c_l}, 1 \leq i \leq M$
 7. **Output:** the normalized vector $v(I) = [v_1, \dots, v_M]$ as the feature-vector representation for image I
-

Remark 2. In Step 3, we choose the size of W to be the maximal size of a prototype but this is not crucial since D can measure the distance between two images of different sizes. From our experiments, the size of W needs to be large enough such that the amount of image information in W is not smaller than that captured in any of the prototypes.

6 Supervised and Unsupervised Learning on Images

Given a corpus $\mathcal{C}$ of images and a set $\mathcal{P}_L$ of labeled prototypes we use Algorithm 3 to generate the feature-vectors $v(I)$ corresponding to each image I in $\mathcal{C}$. At this point we have a database $\mathcal{D}$ of size equal to $|\mathcal{C}|$ which consists of feature vectors of all the images in $\mathcal{C}$. This database can be used for *unsupervised* learning, for instance, discover interesting clusters of images. It can also be used for *supervised* learning provided that each of the cases can be labeled according to a value of some target class variable which in general may be different from the feature categories. Let us denote by T the class target variable and the database $\mathcal{D}_T$ which consists of the feature vectors of $\mathcal{D}$ with the corresponding target class values. The following algorithm describes the process of learning classification of images.

Algorithm 4. Image classification learning

1. **Input:** (1) a target class variable T taking values in a finite set $\mathcal{T}$ of class categories, (2) a database $\mathcal{D}_T$ which is based on the M -dimensional feature-vectors database $\mathcal{D}$ labeled with values in $\mathcal{T}$ (3) any supervised learning algorithm $\mathcal{A}$
 2. Partition $\mathcal{D}_T$ using n -fold cross validation into Training and Testing sets of cases
 3. Train and test algorithm $\mathcal{A}$ and produce a classifier C which maps the feature space $[0, 1]^M$ into $\mathcal{T}$
 4. Define Image classifier as follows: given any image I the classification is $F(I) := C(v(I))$, where $v(I)$ is the M -dimensional feature vector of I
 5. **Output:** classifier F
-

7 Computational Time

Given an image I let us denote by $\tau(I)$ the total time that it takes Algorithm 3 to generate the case (vector-representation) $v(I)$ of I that can be used as a training case or as an input to the classifier F in order to classify the image I .

As mentioned above, Algorithm 2 involves a one-time manual selection of prototypes and the speed is dictated by the user (not the computer). Algorithms 3 is where the computational time is relevant. Step 5 of Algorithm 3 is the time-critical section which governs the overall computational time of the algorithm. This step iterates over all subimages I_j , $1 \leq j \leq m$, of the input image I , and for each subimage it computes the values r_i , $1 \leq i \leq M$, one for each feature-category. In order to compute r_i it computes the *UID* distance between I_j and

prototype $P_k^{(i)}$, $1 \leq k \leq L_i$. To compute $UID(I, J)$ requires building the exhaustive history of both strings $X^{(I)}$, $X^{(J)}$ and of their concatenation $X^{(I)}X^{(J)}$. So the time to compute $UID(I, J)$ is $O(c(X^{(I)}X^{(J)}))$ where $c(X^{(I)}X^{(J)})$ is the length of the exhaustive history of their concatenation. Denoting by $\tau(I, J)$ the time to compute $UID(I, J)$ then it is clear that $\tau(I, J)$ depends on the images I, J and not just on their sizes. That is, $\tau(I, J)$ depends on the LZ-complexity of the two images and on their similarity—the more similar the two, the less complex the concatenation string $X^{(I)}X^{(J)}$ and the smaller $\tau(I, J)$ is.

The time to compute the decided feature-category for subimage I_j of I is big-O the time that it takes to perform the j^{th} iteration of the outer for-loop of step 5. We refer to this as *subimage-time* and denote it by $\tau_j(I)$. We have

$$\tau_j(I) := O\left(\sum_{i=1}^M \sum_{k=1}^{L_i} \tau(I_j, P_k^{(i)})\right)$$

where M is the number of categories, and L_i is the number of prototypes for category i .

Hence if we run on a single processor (single core) the case-generation time $\tau(I)$ of an image I is

$$\tau(I) = \sum_{j=1}^m \tau_j(I) \quad (2)$$

where m is the number of subimages in a single image I . It is clear from this formula that parallel computations (in particular, stream processing where the same function is applied to different data) can be very advantageous for reducing the case-generation time $\tau(I)$.

For instance, on a processor with n cores, where $n \geq m$, each of the cores can compute in parallel a different subimage. This yields a total time

$$\tau(I) = O(\max_{1 \leq j \leq m} \tau_j(I)).$$

If the number of cores n satisfies $m > n \geq M$ then we can let each core compute a different category-sum. This takes a single *sub-image-category* time

$$\tau_j^{(i)}(I) := O\left(\sum_{k=1}^{L_i} \tau(I_j, P_k^{(i)})\right)$$

and in this case the total time is

$$\tau(I) = \sum_{j=1}^m \max_{1 \leq i \leq M} \tau_j^{(i)}(I). \quad (3)$$

If the number of cores $n \geq m \cdot \sum_{i=1}^M L_i$ then the total time to generate $v(I)$ from I is

$$\tau(I) = O\left(\max_{\substack{1 \leq j \leq m \\ 1 \leq i \leq M \\ 1 \leq k \leq L_i}} \tau(I_j, P_k^{(i)})\right). \quad (4)$$

8 Experimental Setup and Results

We created a corpus $\mathcal{C}$ of 60 images of size 670×1364 pixels from GoogleEarth© of various types of areas. Figure 3 displays a few scaled-down examples of such images. From these images we let a user define four feature-categories: *sea*, *urban*, *arid*, *roads* and choose three relatively-small image-prototype of size 45×70 pixels from each feature-category, that is, we ran Algorithm 2 with $M = 4$ and $L_i = 3$ for all $1 \leq i \leq M$. We then ran Algorithm 3 to generate the feature-vectors for each image in the corpus and obtained a database $\mathcal{D}$.

We then let the user label the images by a target variable *Humidity* with possible values 0 or 1. An image is labeled 0 if the area is of low humidity and labeled 1 if it is of higher humidity. We note that an image of a low humidity region may be in an arid (dry) area or also in the higher-elevation areas which are not necessarily arid. Since elevation information is not available in the feature-categories that the user has chosen then the classification problem is hard since the learning algorithm needs to discover the dependency between humid regions and areas characterized only by the above four feature categories.

With this labeling information at hand we produced the labeled database $\mathcal{D}_{Humidity}$. We used Algorithm 4 to learn an image classifier with target *Humidity*. As the learning algorithm $\mathcal{A}$ we used the following standard supervised algorithms: *J48*, *CART*, which learn decision trees, *NaiveBayes* and *Multi-Layer Perceptrons* (backpropagation) all of which are available in the WEKA© toolkit.

We performed 10-fold cross validation and compared their accuracies to a baseline classifier (denoted as ZeroR) which has a single decision that corresponds to the class value with the highest *prior* empirical probability. As seen in Table 1 (generated by WEKA©) *J48*, *CART*, *NaiveBayes* and *Backpropagation* performed with an accuracy of 86.5%, 81.5%, 89.25%, and 87.25%, respectively, compared to 50% achieved by the baseline *ZeroR* classifier. The comparison concludes that all three learning algorithms are significantly better than the baseline classifier, based on a T-test with a significance level of 0.05.

Next, we performed clustering on the unlabeled database $\mathcal{D}$. Using the k-means algorithm, we obtained 3 significant clusters, shown in Table 2.

One can read the information directly from Table 2 and see that the first cluster captures images of *highly urban* areas that are next to concentration of roads, highways and interchanges. The second cluster contains *less populated* (urban) areas in arid locations (absolutely no sea feature seen) with very low concentration of roads. The third cluster captures the *coastal areas* and here we can see that there can be a mixture of urban (but less populated than images of the first cluster) with roads and extremely low percentage of arid land.

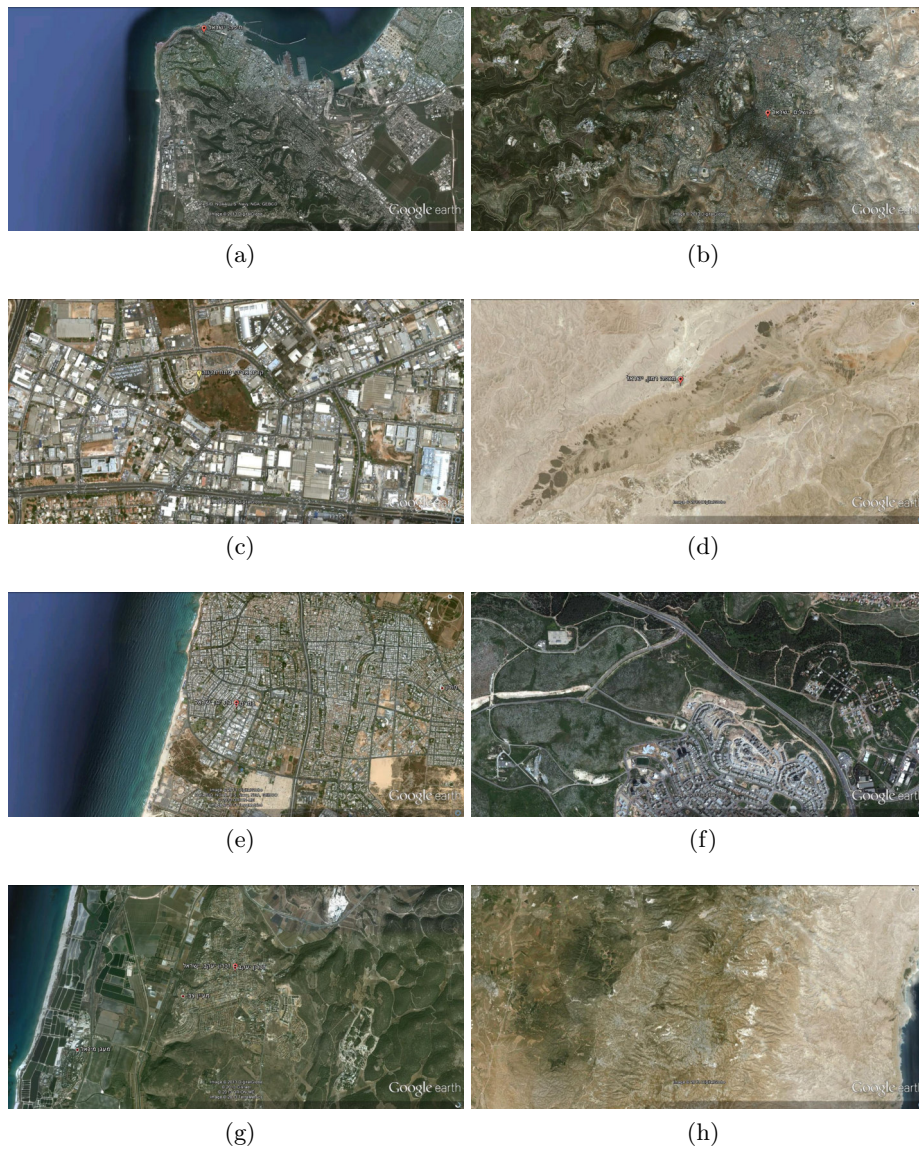


Fig. 3. Examples of images in the corpus

Table 1. Percent correct results for classifying *Humidity*

Dataset $\mathcal{D}_{Humidity}$	(1)	(2)	(3)	(4)	(5)
Classify Image into Humidity:	50.00	86.50	81.50	89.25	87.25

◦, • statistically significant improvement or degradation

- (1) rules.ZeroR " 48055541465867954
(2) trees.J48 '-C 0.25 -M 2' -217733168393644444
(3) trees.SimpleCart '-S 1 -M 2.0 -N 5 -C 1.0' 4154189200352566053
(4) bayes.NaiveBayes " 5995231201785697655
(5) functions.MultilayerPerceptron '-L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a' -5990607817048210779

Table 2. k-means clusters found on unsupervised database $\mathcal{D}$

Feature	Full data	Cluster#1	Cluster#2	Cluster#3
urban	0.3682	0.6219	0.1507	0.2407
sea	0.049	0.0085	0	0.1012
road	0.4074	0.2873	0.0164	0.655
arid	0.1754	0.0824	0.8329	0.003

The fact that such interesting knowledge can be extracted from raw images using our feature-extraction method is very significant since as mentioned above our method is fully automatic and requires no image or mathematical analysis or any sophisticated preprocessing stages that are common in image pattern analysis.

Let us report on some computational time statistics. The hardware we used is a 2.8Ghz AMD Phenom©II X6 1055T Processor with number of cores $n = 6$. The operating system is Ubuntu Linux 2.6.38-11-generic. The corpus consists of images of size 670×1364 pixels, with a sub-image and prototype sizes of 45×70 pixels, the number of subimages per image is $m = 266$. For this test we chose the number of feature-categories $M = 3$ with a total number of prototypes $\sum_{i=1}^M L_i = 11$. We implemented the algorithms in Matlab©in standard code style, that is, with no time-efficiency optimization, except in step 5.a. where the for statement is implemented using a **parfor** statement of Matlab. Note that in this case $m > n > M$ hence the total time to compute one image I is as stated in (3).

We obtained the set of average computational times

$$Timing_Data = \left\{ \bar{\tau}_j^{(i)} : 1 \leq j \leq m, 1 \leq i \leq M \right\}$$

where

$$\bar{\tau}_j^{(i)} := \frac{1}{L_i} \sum_{k=1}^{L_i} \tau(I_j, P_k^{(i)}).$$

Figure 4 shows the histogram for this *Timing_Data*, where the horizontal axis is time in units of seconds. The mean time is 0.851 sec. and the standard deviation

is 0.264 sec. Some of the state-of-the-art Graphics Processor Unit (GPU) accelerators have thousands of execution cores (see for instance, NVIDIA Tesla© K20 which has 2,496 cores) and are offered at current prices of approximately \$2,000. On the NVIDIA Tesla© K10 the number of execution cores is $n = 3072$ and is greater than $m \sum_{i=1}^M L_i = 2,926$ so the total computation time $\tau(I)$ to process a single image I in the corpus will be as in (4), which for this example is approximately 0.851 sec. using Matlab with no optimization. We have not yet done so but we expect that transforming the code from Matlab to C and rewriting it with parallel processing code optimization can yield an average $\tau(I)$ which is lower by several orders of magnitude.

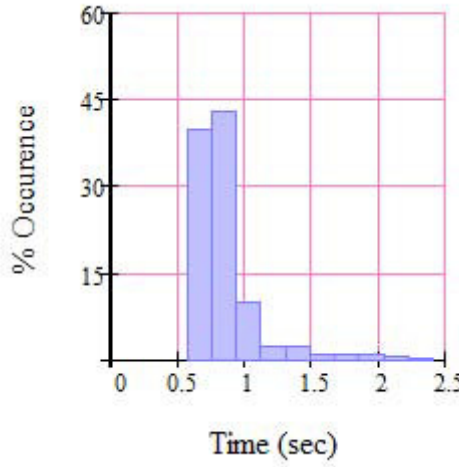


Fig. 4. Histogram of the computational times $\bar{\tau}_j^{(i)}$, $1 \leq j \leq m$, $1 \leq i \leq M$, $m = 266$, $M = 3$. The mean is 0.851.

9 Conclusion

We introduced a method for automatically defining and measuring features of colored images. The method is based on a universal image distance that is measured by computing the complexity of the string-representation of the two images and their concatenation. An image is represented by a feature-vector which consists of the distances from the image to a fixed set of small image prototypes, defined once by a user. There is no need for any sophisticated mathematical-based image analysis or pre-processing since the universal image distance regards the image as a string of symbols which contains all the relevant information of the image. The simplicity of our method makes it very attractive for fast and scalable implementation, for instance on a specific-purpose hardware acceleration chip. We applied our method to supervised and unsupervised machine learning on satellite images. The results show that standard machine learning algorithms perform well based on our feature-vector representation of the images.

Acknowledgements. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K20 GPU used for this research.

References

1. Chester, U., Ratsaby, J.: Universal distance measure for images. In: 2012 IEEE 27th Convention of Electrical Electronics Engineers in Israel (IEEEI), pp. 1–4 (November 2012)
2. Canty, M.J.: Image Analysis, Classification and Change Detection in Remote Sensing: With Algorithms for Envi/Idl. CRC/Taylor & Francis (2007)
3. Lillesand, T.M., Kiefer, R.W., Chipman, J.W.: Remote sensing and image interpretation. John Wiley & Sons (2008)
4. Sayood, K., Otu, H.H.: A new sequence distance measure for phylogenetic tree construction. *Bioinformatics* 19(16), 2122–2130 (2003)
5. Ziv, J., Lempel, A.: On the complexity of finite sequences. *IEEE Transactions on Information Theory* 22(3), 75–81 (1976)
6. Ratsaby, J., Sirota, V.: Fpga-based data compressor based on prediction by partial matching. In: 2012 IEEE 27th Convention of Electrical Electronics Engineers in Israel (IEEEI), pp. 1–5 (November 2012)
7. Ratsaby, J., Zavielov, D.: An fpga-based pattern classifier using data compression. In: Proc. of 26th IEEE Convention of Electrical and Electronics Engineers, Israel, Eilat, November 17–20, pp. 320–324 (2010)
8. Kaspi, G., Ratsaby, J.: Parallel processing algorithm for Bayesian network inference. In: 2012 IEEE 27th Convention of Electrical Electronics Engineers in Israel (IEEEI), pp. 1–5 (November 2012)
9. Cilibiasi, R., Vitanyi, P.: Clustering by compression. *IEEE Transactions on Information Theory* 51(4), 1523–1545 (2005)
10. Deza, M., Deza, E.: Encyclopedia of Distances. Series in Computer Science, vol. 15. Springer (2009)



On Advances in Statistical Modeling of Natural Images

A. SRIVASTAVA*

Department of Statistics, Florida State University, Tallahassee, FL 32306, USA

anuj@stat.fsu.edu

A.B. LEE

Division of Applied Mathematics, Brown University, Providence, RI 02912, USA

E.P. SIMONCELLI

Courant Institute for Mathematical Sciences, New York University, New York, NY 10003, USA

S.-C. ZHU

Department of Computer Science, Ohio State University, Columbus, OH 43210, USA

Abstract. Statistical analysis of images reveals two interesting properties: (i) invariance of image statistics to scaling of images, and (ii) non-Gaussian behavior of image statistics, i.e. high kurtosis, heavy tails, and sharp central cusps. In this paper we review some recent results in statistical modeling of natural images that attempt to explain these patterns. Two categories of results are considered: (i) studies of probability models of images or image decompositions (such as Fourier or wavelet decompositions), and (ii) discoveries of underlying image manifolds while restricting to natural images. Applications of these models in areas such as texture analysis, image classification, compression, and denoising are also considered.

Keywords: natural image statistics, non-Gaussian models, scale invariance, statistical image analysis, image manifold, generalized Laplacian, Bessel K form

1. Introduction

Applications dealing with static or dynamic images (movies) have become increasingly important and popular in recent years. Tools for image analysis, compression, denoising, transmission, and understanding have become widely involved in many scientific and commercial endeavors. For instance, compression of images is important in transmission or storage, and automatic recognition of people from their camera images has become important for security purposes. De-

velopment of tools for imaging applications starts with mathematical representations of images. Many of these applications need to “learn” some image patterns or tendencies before specific data can be analyzed. The area of natural image statistics has resulted from efforts to observe, isolate, and explain patterns exhibited by natural images. Lately, there has been a greater emphasis on explicit probability models for images. Perhaps one reason for that is the growing appreciation for the variability exhibited by the images, and the realization that exact mathematical/physical models may not be practical, and a statistical approach needs to be adopted. Any statistical approach will need probability models that capture “essential” image variability and yet are

*To whom correspondence should be addressed.

computationally tractable. Another motivation for pursuing image statistics has been to understand sensory coding as a strategy for information storage/processing in animal vision. An understanding of image statistics may provide clues to the architecture of animal visual system [79].

There are several paths to understanding image statistics. Even though images are expressed as elements of a large vector space (e.g. the space of rectangular arrays of positive numbers), referred to as the *image space* here, the subset of interesting images is rather small and restricted. So one path is to isolate this subset, called an *image manifold* here, and impose a simplistic probability model, such as a uniform or a Gaussian-like model, on it. Given the image manifolds and probability distributions on them, statistical tools for imaging applications follow. The other path is to derive probability models that are defined on the larger image space but put any significant probability only on the image manifold. We have divided this paper along these two categories.

In search for statistical descriptions of images, mathematical and physical ideas are not abandoned but are intimately involved. For example, harmonic analysis of image intensities is used to decompose images into individual components that better lend to the model building than the original images. Since image spaces are rather high dimensional, and common density estimation techniques apply mostly to small sizes, one usually starts by decomposing images into their components, and then estimating lower order statistics of these components individually. Also, imaging is a physical process and physical considerations often contribute to the search for statistical descriptors in some form. Physical models have been the motivation of many studies that have led to interesting statistical characterizations of images.

This paper is laid out as follows: In Section 2 we start with a historical perspective on spectral analysis of images and some discoveries that followed. An important achievement was the discovery of non-Gaussianity in image statistics. In next two sections we present some recent results categorized into two sets: Section 3 studies probability models for images decomposed into their spectral components, while Section 4 looks at methods for discovering/approximating image manifolds for natural images. A few applications of these methods are outlined in Section 5, and some open issues are discussed in Section 6.

2. Image Decompositions and Their Statistical Properties

For the purpose of statistical modeling, an image is treated as a realization of a spatial stochastic process defined on some domain in $\mathbb{R}^2$. The domain is assumed to be either a (continuous) rectangular region or a finite, uniform grid. A common assumption in image modeling is that the underlying image process is stationary, i.e. image probabilities are invariant to translations in the image plane.

2.1. Classical Image Analysis

In case the image process is modeled as a second order spatial process, the spectral analysis becomes a natural tool. Defining the covariance function, for two arbitrary pixel values, as $C(x)$ where x is the difference between two pixel locations, one can define the power spectrum as $P(w) = \int_{\mathbb{R}^2} C(x) e^{-jwx} dx$, and where w denotes the 2D spatial frequency. Since a Fourier basis is also an eigen basis for circulant matrices, which is the case for $C(x)$ under the stationarity assumption, Fourier analysis also coincides with the popular principal component analysis (discussed later in Section 4). Also, the Fourier representation guarantees uncorrelated coefficients. Images are then represented by their Fourier coefficients, image statistics are studied via coefficient statistics, and one can inherit a stochastic model on images by imposing a random structure on the Fourier coefficients. Specification of means and covariances of the coefficients completely specifies a second order image process. Early studies of spatial power spectra indicated that the power $P(w)$ decays as $\frac{A}{|w|^{2-\eta}}$ where $|w|$ is the magnitude of the spatial frequency. This property, called the power law spectrum for images, was first observed by television engineers in the 50's [25, 49] and discovered for natural images in late 80s by Field [30] and Burton and Moorhead [17]. As summarized in [61], the value of η changes with the image types but is usually a small number.

Although Fourier analysis is central to classical methods, other bases have found popularity for a variety of reasons. For instance, in order to capture the locality of objects in images, decomposition of images using a wavelet basis has become an attractive tool. In particular, it is common to use Gabor wavelets [31] for decomposing the observed images simultaneously

in space and frequency. In addition, Marr [56] suggested using the Laplacian of Gaussian filter to model early vision. If one considers images as realizations on a finite, uniform grid in $\mathbb{R}^2$, the image space becomes finite-dimensional, and one can linearly project images into low-dimensional subspaces that are optimal under different criteria. Some of these linear projections are covered in Section 4. Once again image statistics are studied via the statistics of the projected coefficients.

2.2. Scale Invariance of Image Statistics

A discovery closely related to the power law is the invariance of image statistics when the images are scaled up or down [16, 30]. In other words, the marginal distributions of statistics of natural images remain unchanged if the images are scaled. The power law is a manifestation of the fractal or scale invariant nature of images. By studying the histograms of the pixel contrasts ($\log(I(x)/I_0)$) at many scales, Ruderman and Bialek [71] showed its invariance to scaling. Independently, Zhu and Mumford [98] showed a broader invariance by studying the histograms of wavelet decompositions of images. Ruderman [69, 70] also provided evidence of scale invariance in natural images and proposed a physical model for explaining them. Turiel and Parga [84] investigated the multi-fractal structure of natural images and related it to the scale invariance. Scaling of different types of scenes was studied by Huang [43]. In [85], Turiel et al. showed that hyperspectral images (color images) also demonstrate multiscaling properties and the statistics are invariant to scale. Scaling of order statistics of pixel values in small windows was studied by Geman and Koloydenko [32]. In addition to pixel statistics, scaling of topological statistics obtained from morphological operations on images was demonstrated by Alvarez et al. [1].

It must be emphasized that only the statistics of large ensembles of images are scale invariant; statistics of individual images vary across scales. Theoretical models that seek probabilistic description of image ensembles aim for scale invariance, while application driven models that work with individual images aim to capture individual image variability.

2.3. Non-Gaussianity of Marginal Statistics

Classical methods assume that images are second order processes but the observations do not support this

assumption. Higher order statistics of natural images were found to exhibit interesting patterns and the researchers focused next on these higher order moments. One implication is that image statistics do not follow Gaussian distribution and require higher order statistics. For example, a popular mechanism for decomposing images locally—in space and frequency—using wavelet transforms leads to coefficients that are quite non-Gaussian, i.e. the histograms of wavelet coefficients display heavy tails, sharp cusps at the median and large correlations across different scales. To our knowledge, Field [30] was the earliest to highlight the highly kurtotic shapes of wavelet filter responses. Mallat [55] pointed out that coefficients of multiscale, orthonormal wavelet decompositions of images could be described by generalized Laplacian density (given later in Section 3.2). This non-Gaussian behavior of images has also been studied and modeled by Ruderman [69], Simoncelli and Adelson [76], Moulin and Liu [60], and Wainwright and Simoncelli [88]. Recent work of Thomson [82] studies the statistics of natural images using phase-only second spectrum, a fourth order statistic, and demonstrates both the power law and the scale invariance for this statistic. In fact, projection onto any localized zero-mean linear kernel seems to produce kurtotic responses [95]. Huang [43] showed that images, when filtered by 8×8 random mean-0 filters, have distributions with high kurtosis, a sharp cusp at zero and long exponential tails. This suggests a role for linear decompositions that maximize the kurtosis or some another measure of non-Gaussianity. Such efforts [6, 62, 86] have resulted in bases that are spatially oriented with (spatial) frequency bandwidths being roughly one octave, similar to many multiscale decompositions. Similar results were obtained by minimizing the independence of coefficients, under linear decompositions, leading to independent component analysis [18, 23, 44]. These observations justify widespread use of orthonormal wavelets in general image analysis applications. Use of Gabor wavelets is also motivated by the fact that the receptive fields of simple cells in the visual cortex of animals have been found to resemble Gabor functions [58]. Note, however, that most wavelet decompositions of images are based on separable application of one-dimension filters, which leads to non-oriented (mixed diagonal) subbands. Alternate representations that provide better orientation decomposition, and thus higher kurtosis response, include [24, 27, 78, 90].

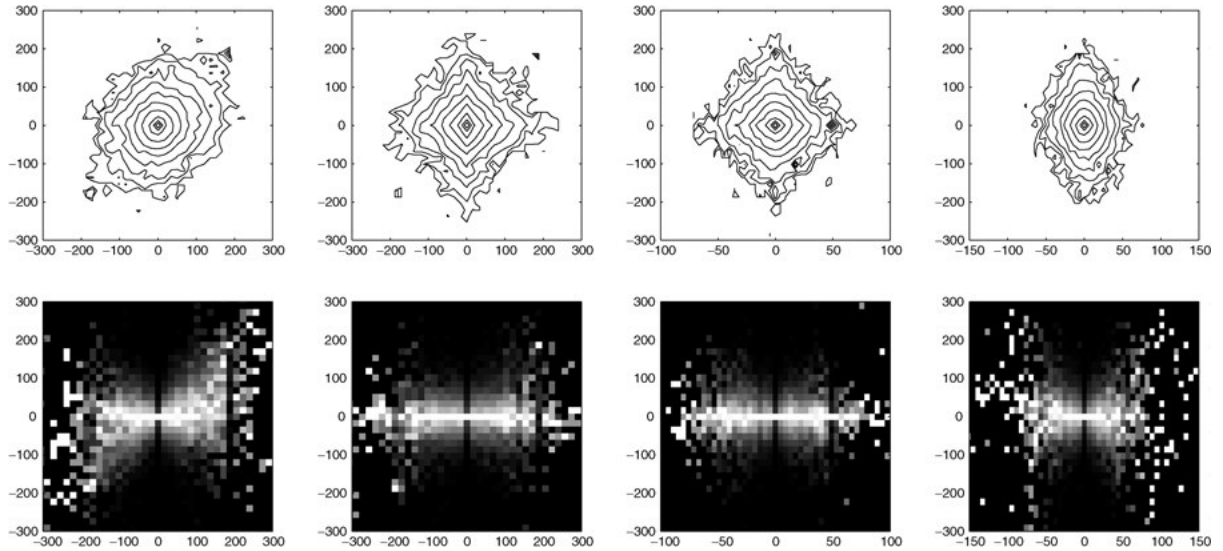


Figure 1. Bivariate histograms of wavelet coefficients associated with different basis functions. Top row shows contour plots, with lines at equal intervals of log probability. Left two cases are for different spatial offsets (same scale and orientation), third is for different orientation (same scale and nearly same position), and the rightmost corresponds to a pair at adjacent scales (same orientation and nearly same position). Bottom row shows some conditional distributions: brightness corresponds to larger frequency.

2.4. Non-Gaussianity of Joint Statistics

In addition to the non-Gaussian behavior of marginal statistics, a number of authors have studied joint statistics of filter responses. In particular, the local structures present in most images lead to dependencies in the responses of local linear operators. A study of joint histograms of wavelet coefficients shows dependency across scales, orientations and positions. Zetzsche and colleagues noted that the joint density of projections onto even- and odd-symmetric Gabor filters have circular symmetry. If the density were Gaussian, this would correspond to independent (uncorrelated) marginals, but the highly kurtotic marginals imply that the responses are strongly dependent [91]. Shapiro developed a heuristic method, for taking advantage of joint dependencies between wavelet coefficients, that revolutionized the field of image compression [73]. Simoncelli and colleagues studied and modeled the dependency between responses to pairs of bandpass filters and found that the amplitudes are strongly correlated, even when the signed responses are not [14, 74]. This is illustrated in Fig. 1, which shows conditional histograms for several pairs of coefficients. Note that unlike second-order correlations, these dependencies can not be eliminated with a linear transform.

Observed statistics of an image patch can be described by joint histograms of coefficients under several

filters. For jointly Gaussian coefficients equiprobable contours in joint histograms would all be ellipsoidal, but in the case of natural images the 2D and 3D contour surfaces display striking polyhedra-like shapes. Huang [43] showed that the peaks and cusps in the contours correspond to occurrences of simple geometries in images with partially constant intensity regions and sharp discontinuities. These results points to the ubiquity of “object-like” structures in natural images and underline the importance of object-based models. Grenander and Srivastava [36] also has attributed non-Gaussianity to the presence of objects in images and used that idea for model building. Lee et al. have shown that images synthesized from occlusion models (Section 3.1) show irregular polyhedra-like shapes in contour surfaces of histograms.

3. Emerging Statistical Models on Image Space

Earliest, and still widely used, probability models for images were based on Markov random field models (MRFs) [93]. An image field is as a collection of random variables, denoting pixel values on a uniformly spaced grid in the image plane. In MRFs, the conditional probability of a pixel value given the remaining image is reduced to a conditional probability given a neighborhood of that pixel. Efficiency results if the neighborhood of a pixel is small, and furthermore,

stationarity holds if the same conditional density is used at all pixels. Ising and Potts model are the simplest examples of this family. Besag [8, 9] expressed the joint density of image pixels as a product of conditional densities, and ignored the normalizer to obtain a pseudo-likelihood formulation. Clifford-Hammersley theorem, see for example [93], states that full conditionals completely specify the joint density function (under a positivity assumption) and enabled the analysis of images using a Gibbs sampler. Geman and Geman [33] utilized the equivalence of MRFs and Gibbs distributions to sample from these distributions. Kersten [47] worked on computing the conditional entropies of the pixel values, given the neighboring pixels in an MRF framework. Zhu and Mumford [98] used a Gibbs model on images and estimated model parameters using a minimax entropy criterion. Let $H(I)$ be a concatenation of the histograms of image coefficients, obtained using several wavelet bases. Then the maximum entropy probability takes the form: $P(I | \lambda) \propto e^{-(\lambda, H(I))}$. The vector λ is estimated by setting the mean histograms to equal the observed histograms, i.e. $\int H(I)P(I | \lambda) dI = H_{obs}$, and reduces to the maximum likelihood estimation of λ under $P(I_{obs} | \lambda)$.

3.1. Models Motivated by Physics

A number of researchers have studied image statistics from a physical viewpoint, trying to capture the phenomena that generates images in the first place. A common theme to all these models is random placements of planar shapes (lines, templates, objects, etc.) in images according a Poisson process. Different models differ in their choice of shapes (e.g. primitive versus advanced), and their interaction (e.g. some favor occlusion while others favor superposition). Here we summarize a few of these models:

3.1.1. Superposition Models. In order to capture scale invariance and non-Gaussianity of images, Mumford and Gidas [61] utilized a family of infinitely divisible distributions. They showed that such distributions arise when images are modeled as superpositions of random placements of objects. To achieve self-similarity, it was proposed that the sizes of objects present in images be distributed according to density function Zr^{-3} over a subset of $\mathbb{R}_+$ (Z is the normalizing constant). Using this model, Chi [19] described a Poisson placement of objects with sizes sampled according to the $1/r^3$ -law. Additionally, he assumed a

surface process that models the texture variation on the 2D surfaces of the objects. Bitouk et al. [11] and Grenander and Srivastava [36] have also assumed that images are made up of 2D appearances of the objects g_i s placed at homogeneous Poisson points z_i s. The objects are chosen randomly from an object space (arbitrary sizes, shapes or texture). A weighted superposition, with weights (or contrasts) given by independent standard normals a_i s, is used to form an image; it allows for expressing the linearly filtered images as a superposition of filtered objects. This formulation leads to an analytical (parametric) probability for individual images with the final form given in Section 3.2.

3.1.2. Occlusion Models. In more sophisticated models, objects are placed randomly as earlier but now the objects in the front occlude those in the back. An example is the dead leaves (also called a random collage) model which assumes that images are collages of approximately independent objects that occlude each other. One generates images from the model by placing an ordered set of elementary 2D shapes g_i s in layers. The locations z_i s with sizes r_i s are sampled from a Poisson process. For gray level images, the random sets or “leaves” $T_i = g_i(z - z_i)$ are furthermore associated with intensities a_i . The leaves are typically placed front-to-back¹ according to

$$I^{(i)}(z) = \begin{cases} a_i & \text{if } z \notin T_j, z \in T_i, \forall j < i \\ I^{(i-1)}(z) & \text{otherwise} \end{cases},$$

where $i = 1, 2, \dots$

Left two panels in Fig. 2 show two samples of a dead leaves model with elliptical and square random shapes, respectively. The final model is defined in the limit $\lim_{i \rightarrow \infty} I^{(i)}(z)$, or equivalently, when the finite image domain $\Omega \subset \mathbb{R}^2$ is completely covered. The dead leaves model dates back to Matheron [57] and Serra [72] in mathematical morphology. They showed that the probability of any compact set $K \subset \Omega$ belonging to the same leaf and not being occluded by other leaves equals the ratio $\frac{E[v(X_0 \ominus K)]}{E[v(X_0 \oplus K)]}$. Here, X_0 is the random shape used in the model, $E[v(\cdot)]$ is the expected Lebesgue measure, and $X_0 \ominus K = \{x : x + K \in X_0\}$ and $X_0 \oplus K = \{x : X_0 \cap (x + K) \neq \emptyset\}$ represent the *erosion* and the *dilation*, respectively, of X_0 by K . Early applications of this model include studies of inhomogeneous materials and the size distribution of grains in powder [45]. More recently, researchers have applied different versions of the dead leaves model to natural

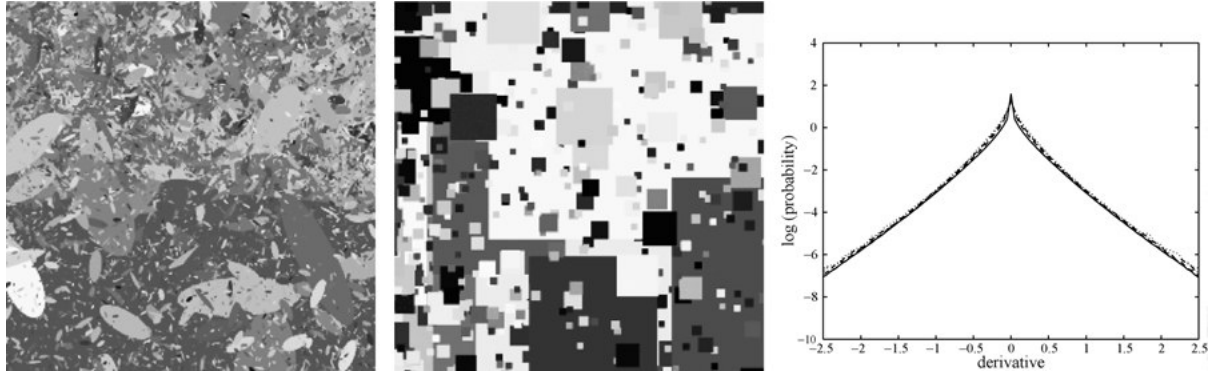


Figure 2. Left two pictures: Two different versions of the dead leaves model with elliptical (left) and square (middle) random shapes, respectively. Right panel: Derivative statistic and scaling of a dead leaves model with disks distributed according to a cubic law of sizes. The curves correspond to the marginal distribution of the differences between adjacent pixels after contrast normalization at four different scales ($N = 1, 2, 4, 8$).

scenes. In [70], Ruderman proposed that a random collage description with statistically independent ‘objects’ can explain scaling of correlations in images. Using an occluding object model and the difference function of hand-segmented images, he infers a power law for the probability of two points belonging to the same object. This result is consistent with [1] that uses level sets and the original dead leaves formalism of Matheron and Serra. Here, Alvarez et al. analyze the morphological structure of natural images—in particular, the effect of occlusions on the area, perimeter and intercept lengths of homogeneous, connected image regions.

Lee and Mumford [50] have used a dead leaves model with statistically independent objects and the measure of a marked Poisson process to show that $1/r^3$ power law in object sizes also leads to approximate full scale invariance when occlusions are taken into account. Provided that synthesized images are smoothed by, for example by block averaging, their statistics match those observed in large data sets of calibrated natural images. Scale invariance of histograms of filter responses, full co-occurrence statistics of two pixels, and joint statistics of Haar wavelet coefficients are studied in this paper. Figure 2 (right panel) shows a histogram of the derivative statistic, under scaling, in a dead leaves model with smoothing.

In addition, there have been some analogies drawn from other sciences to explain self-similarity of natural images. For example, Ruderman et al. [69, 71] utilized the statistics of turbulent flows. In [83], Turiel et al. formally connected the statistical modeling of turbulent flows to that of images. In turbulent flows, if the second moment of the energy dissipation is proportional to r^τ for some τ , where r is the box size, then self-similarity

holds. Using a measure based on edges to compute energy dissipation in images, the authors showed self-similarity of edge statistics.

3.2. Analytical Densities: Univariate Case

An important requisite of statistical models is that they be computationally efficient for real-time, or near real-time, applications, and one obvious way to accomplish that is via parametric densities. We want to represent statistical nature of images by means of parametric densities, using only limited parameters. In this section, we discuss families of parametric densities that seems to capture the variability in low-dimensional representations of images. Rather than explaining scale-invariance of a large collection of images, the goal here is to *capture variability of individual images*. For an image I and a linear filter F , this section deals with characterizing the marginal density of the pixel values in $I * F$.

1. *Generalized Laplacian Model*: Marginal densities of image coefficients are well modeled by a generalized Laplacian density (also called generalized Gaussian or stretched exponential) $f_1(x; c, p) = \frac{e^{-|x/c|^p}}{Z_1(p, c)}$, [55, 60, 76], where the normalizer is $Z_1(p, c) = 2 \frac{c}{p} \Gamma(\frac{1}{p})$. The parameters, $\{p, c\}$, may be estimated for the subbands of specific images using maximum likelihood or the method of moments. Another way of estimating them is via the (linear) regression of $\log(\log(h(x) + h(-x)) - 2 \log(h(0)))$ versus $\log(|x|)$, where $h(x)$ is the histogram value at x and x is the variable for bin centers. Values for the exponent p are typically within the range $[0.5, 0.8]$,

and the width parameter c varies monotonically with the size of the basis functions, producing higher variance for coarser-scale components [76].

Under Grenander's formulation of a superposition model (item 1, Section 3.1), if the random variable $u(z) \equiv \sum_{i=1}^n ((g_i * F)(z - z_i))^2$ is assumed to have a scaled-Gamma density, then the univariate density of the filtered pixel has been shown in [36] to be: for $p > 0$, $c > 0$, $f_2(x; p, c) = \frac{1}{Z_2(p, c)} |x|^{p-0.5} K_{(p-0.5)}(\sqrt{\frac{2}{c}} |x|)$, where K is the modified Bessel function of third kind, and Z_2 is the normalizing constant. (Its characteristic function is of the form $1/(1 + 0.5c\omega^2)^p$ which for $p = 1$ becomes the characteristic function of a Laplacian density.) This parametric family of densities has been called *Bessel K forms* with (p, c) referred to as *Bessel parameters*. Earlier, Wainwright et al. [89] also investigated the application of the Gaussian scale mixture family to image modeling and referred to f_2 as the *symmetrized Gamma* density (without reaching its analytic, parametric form). As described in [36], p and c are easily estimated from the observed data, with $\hat{p} = \frac{3}{SK(I^{(j)})-3}$ and $\hat{c} = \frac{SV(I^{(j)})}{\hat{p}}$, where SK is the sample kurtosis and SV is the sample variance of the filtered image. A distinct advantage of this model is that parameters can be related mathematically to the physical characteristics of the objects present in the image. If a filter F is applied to an image I to extract some specific feature—vertical edges, say—then, the resulting p has been shown in

[80] to depend on two factors: (i) distinctness and (ii) frequency of occurrence of that feature in I . Objects with sharper, distinct edges have low p values, while scenes with many objects have large p values.

2. *Gaussian Scale Mixtures*: Observed statistics of filtered images have pointed towards a rich family of univariate densities that are highly kurtotic and heavy tailed. In statistics literature, there is a wide usage of the variables defined as *normal variance-mean mixture*; X is called a normal variance-mean mixture if the conditional density function of X given u is normal with mean $\mu + u\beta$ and variance $u\Delta$, and u is called the mixing variable. Generalized hyperbolic distributions were introduced by Barndorff-Nielsen [3] as specific normal variance-mean mixtures that result when the mixing variable u is of certain class. For $\mu = \beta = 0$ and $\Delta = 1$, the resulting family is also called Gaussian scale mixture [2] and has seen applications in financial mathematics [12] and speech processing [13]. Furthermore, if u is a scaled Gamma density then Bessel K density results.

Shown in Fig. 3 are some examples of estimating this density function: natural images from van Hateren database are filtered using arbitrary Gabor filters (not shown), and the resulting pixel values are used to form the histogram $h(x)$. Bottom panels of this figure show the plots of $\log(h(x))$, $\log(f_1(x))$, and $\log(f_2(x))$ with parameters estimated from the corresponding images.

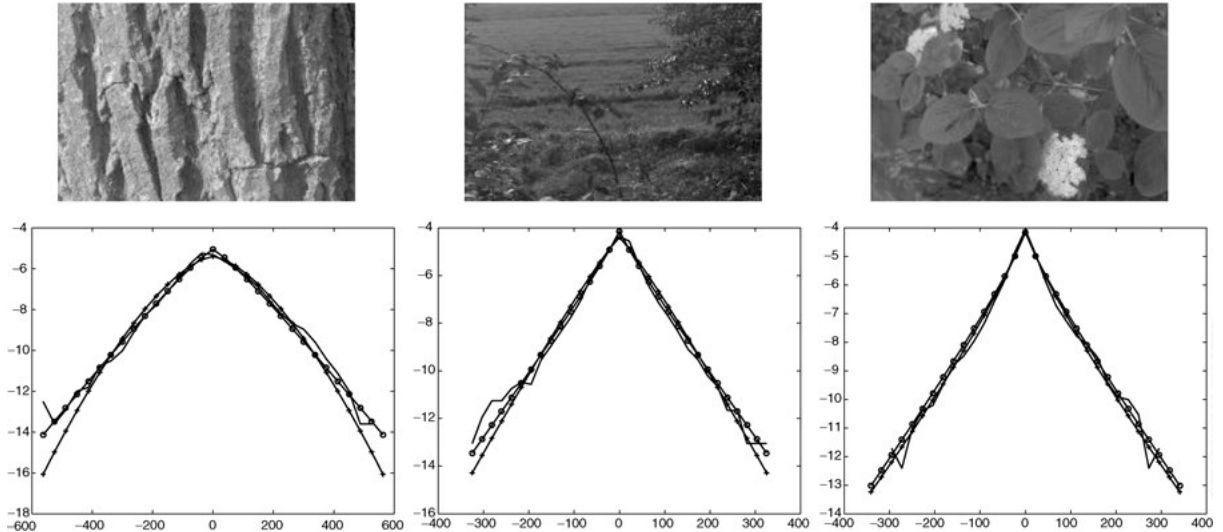


Figure 3. Estimated Bessel K and generalized Laplacian densities compared to the observed histograms for the images in top. Plain lines are histograms, lines with large beads are Bessel K, and lines with small beads are generalized Laplacian. These densities are plotted on a log scale.

There are several ways to judge the performance of any proposed probability model. The simplest idea is to compare the observed frequencies to the frequencies predicted by the model, using any metric on the space of probability distributions. Two commonly used metrics are the Kolmogorov-Smirnov distance: $\max_{x \in \mathbb{R}} |\int_{-\infty}^x (f_1(y) - f_2(y)) dy|$ and the Kullback-Leibler divergence: $\int_{\mathbb{R}} f_1(x) \log(\frac{f_1(x)}{f_2(x)}) dx$. Both the generalized Laplacian and the Bessel K form perform well in capturing univariate image statistics under these metrics. In general, for small values of p (sharper features and less number of objects in image) the generalized Laplacian performs better while for larger p the Bessel K density better matches the observed densities.

3.3. Bivariate Probability Models

So far we have discussed only univariate models but the complexity of observed images points to statistical interactions of high orders. As a first extension, we look at the models for capturing pairwise interactions between image representations, i.e. the bivariate probability densities. For instance, these models may involve a pair of wavelet coefficients at different scales, positions, or orientations.

3.3.1. Variance Dependency and Gaussian Scale Mixtures. As described in Section 2.4, pairs of wavelet coefficients corresponding to basis functions at nearby positions, orientations, or scales are not independent, even when the responses are (second-order) decorrelated. Specifically, the conditional variance of any given coefficient depends quite strongly on the values of surrounding coefficients [74]. As one considers coefficients that are more distant (either in spatial position, orientation, or scale), the dependency becomes weaker. This dependency appears to be ubiquitous, and may be observed across a wide variety of image types.

What sort of probability model can serve to explain the observations of variance dependency? One candidate is the Gaussian scale mixture model described earlier for the univariate case. In this scheme, wavelet coefficients are modeled as the product of a Gaussian random variable and a hidden “multiplier” random variable (same as mixing variable u earlier) that controls the variance. To explain pairwise statistics shown in Fig. 1, one can prescribe a relationship between the hidden multiplier variables of neighboring coefficients. That is, the hidden variables now depend on one another, generating the variance scaling seen in the data. Mod-

eling of this dependency remains an ongoing topic of research. One possibility is to link these variables together in a Markov tree, in which each multiplier is independent of all others when conditioned on its parent and children in the tree [89].

3.3.2. Bivariate Extension of Bessel K Forms. The univariate Bessel K form has been extended by Grenander to specify bivariate densities of the filtered images. The basic idea is to model all linear combinations of the filtered pixels by Bessel K forms and then invoke the Cramer-Wold device, see [10] for a definition, which states that specification of densities of all linear combinations of a number of random variables specifies uniquely their joint density. The use of the Cramer-Wold device here assumes that there exists a 2D distribution whose Radon-like marginals (for half spaces, not on lines) behave according to Bessel K densities. An imposition of Bessel K forms on marginals seems to agree well with the data qualitatively but its performance remains to be quantified over large datasets.

Let $I_1 = I * F^{(1)}$ and $I_2 = I * F^{(2)}$ for two wavelet filters $F^{(1)}$ and $F^{(2)}$, and for $a_1, a_2 \in \mathbb{R}$, let $J(a_1, a_2, z) \equiv a_1 I_1(z) + a_2 I_2(z)$. The Cramer-Wold idea is to compute the characteristic functions of $J(a_1, a_2, z)$, for all pairs (a_1, a_2) , using Bessel parameter estimation, and then take an inverse Fourier transform to obtain an estimate of the joint density $f(I_1(z), I_2(z))$. This estimate is parametrized by eight joint moments: $\mu_{0,2}, \mu_{1,1}, \mu_{2,0}, \mu_{4,0}, \mu_{3,1}, \mu_{2,2}, \mu_{1,3}, \mu_{0,4}$, where $\mu_{i,j} = \int \int I_1^i I_2^j f(I_1, I_2) dI_1 dI_2$. Shown in Fig. 4 is an example of this bivariate density estimation. For the image shown in left, consider the bivariate density of its two filtered versions under two Gabor filters at the same scale but orientations 20 degrees apart. Top two panels show the mesh and the contour plots of the estimated density while the bottom panels show the observed bivariate histogram. The densities are all plotted on a log scale.

4. Discovering Image Manifolds

It has been well highlighted that in the space of rectangular arrays of positive numbers, only a small subset has images of natural scenes. One seeks to isolate and characterize this subset for use in image analysis applications. The main idea is to identify this set as a low-dimensional, differentiable manifold and use its geometry to characterize images. Having defined this manifold, a simplistic probability model can help capture the image variability. We now present a summary

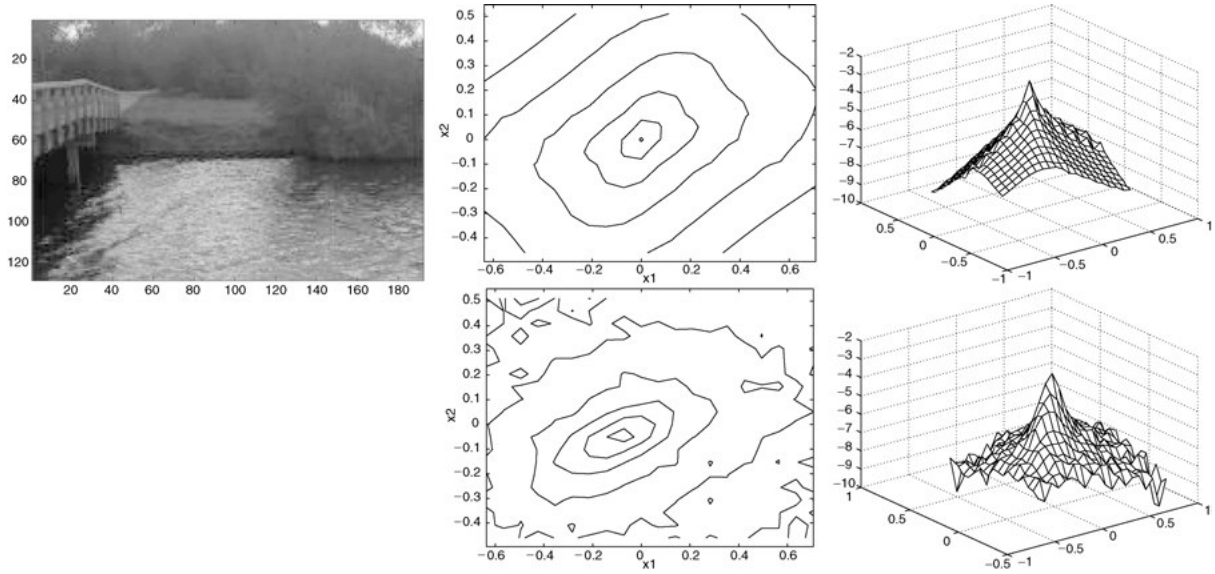


Figure 4. Bivariate extension of Bessel K form: For the image shown in left panel, the estimated (top) and observed (bottom) bivariate densities plotted as meshes and contours, on a log scale.

of some commonly used methods for estimating image manifolds.

4.1. Approximating Image Manifolds by Linear/Locally-Linear Subspaces

Perhaps the easiest technique to approximate the image manifold is to fit a linear subspace through the observations. The fitting criterion will specify the precise subspace that is selected. For example, if one intends to minimize the cumulative residual error (Euclidean distance between the observation points and the subspace), then the optimal subspace is the dominant (or principal) subspace of the data matrix, and is easily computed using eigen decomposition or singular value decomposition. A common application of PCA is in recognition of people from their facial images [48] or study of natural images [39]. Instead, if the goal is to minimize the statistical correlation between the projected components, or to make them as independent as possible, then the independent component basis results [6, 23, 86]. Other criteria lead to similar formulations of the subspace basis such as sparsity [63], Fisher discrimination [5], and non-negative factorization [52]. The use of sparseness is often motivated by the scale invariance of natural images.

Approximating the image manifold by a flat subspace is clearly limiting in general situations. In [96], the authors argue that linear processing of images

leaves substantial dependencies between the components, and a nonlinear technique is required. One extension is to seek a “local linear embedding” approximation of the image manifold by fitting neighboring images by low dimensional subspaces [68, 81]. Definition of a neighborhood is through Euclidean metric but that deserves further study. Another idea is to combine local basis elements, such as wavelet bases, into higher level structures that provide a better representation of the image manifold. For instance, Zhu et al. [97] have combined placements of transformed basis elements to form structures called *textons* in order to better characterize images and their manifolds. Shown in the left panel of Fig. 5 is an example of building a “star” texton to match the given image.

4.2. Deformable Template Characterization of Image Manifold

Grenander’s pattern theoretic framework for image understanding is motivated by physical models and leads to probabilistic, algebraic representations of scenes. Objects appearing in the scenes are represented by 3D models of shape, texture, and reflectivity, and their occurrences in scenes are captured by group transformations on the typical occurrences or templates. A strong feature of this approach is the logical separation between the actual, physical scenes and the resulting images. Variability of scenes is better modeled in a

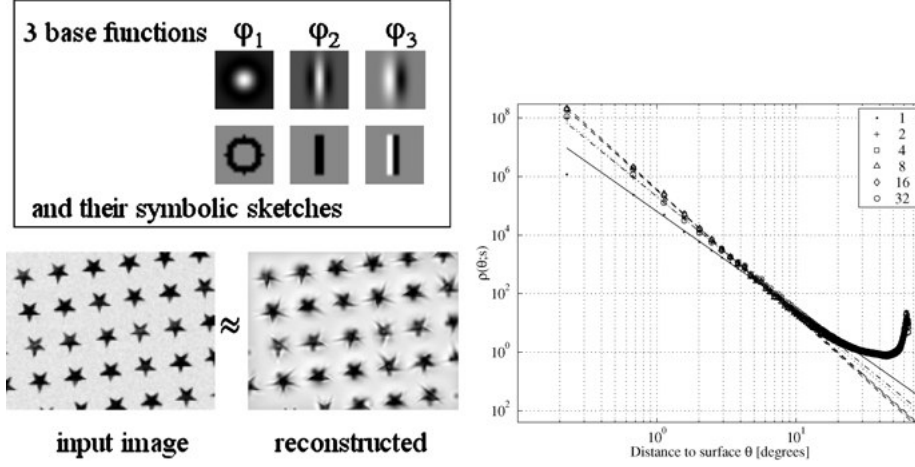


Figure 5. Left panel: To model an image of stars, some wavelet bases are combined to form a star textron. Right Panel: Natural images have an infinite probability density at a manifold in state space that corresponds to blurred edges. The figure shows the empirical probability density as a function of the distance to the manifold of Gaussian scale-space edges in jet space, with curves corresponding to jet representations of natural images at scales $s = 1, 2, 4, 8, 16, 32$.

3D coordinate system as it is guided by the physical principles, rather than the image space where the Euclidean representations do not work well. Since the two, 3D scenes and 2D images, are related by a projection (orthographic or perspective) map, manifolds formed by images can be generated by projecting the manifolds formed using 3D representations. As described in [34, 59], occurrences of physical objects in 3D system are modeled as the orbits of groups acting on the configurations of objects in the scenes. These orbits are projected into the image space to form manifolds on which the images lie. Let C^α be a CAD model of a 3D object labeled by α and let S be the group of transformations (3D rotation, translation, deformation, etc.) that change the object occurrences in a scene. Let, for $s \in S$, sC^α denote the action of s on the template. Then, $\{sC^\alpha : s \in S\}$ is a 3D orbit associated with object α . Further, if T is the imaging projection from a 3D scene to the image plane, then $\{T(sC^\alpha) : s \in S\}$ is the image manifold generated by this object, and dimension of this image manifold equals $\dim(S)$. Geometry of this manifold, such as the tangent spaces or the exponential maps, can also be obtained by projecting their counterparts in the bigger space.

4.3. Empirical Evidence of Manifolds for Image Primitives

In contrast to a physical specification, as advocated by the deformable template theory, one can also study image manifolds directly using the ob-

served images. The image based search [51, 64] is also inspired by Marr’s idea [56] of representing early vision by converting an array of raw intensities into a symbolic representation—a so called “primal sketch”—with primitives such as edges, bars, blobs and terminations as basic elements. An important issue in computer and human vision is: How are Marr’s primitives represented geometrically and statistically in the state space of image data? The geometrical model that arises from Marr’s hypothesis is a set of continuous manifolds of the general form $M(s) = [F^{(1)}(\cdot) * sC^\alpha, F^{(2)}(\cdot) * sC^\alpha, \dots]$, $F^{(j)}(j = 1, \dots, n)$ form a bank of filters, and sC^α is an image of a primitive α parameterized by $s \in S$. Dimension of resulting image manifold $M(s)$ is determined by an intrinsic dimensionality of the primitives; it is usually a small number (2 for contrast-normalized edges, 3 for bars and circular curves etc.). Furthermore, the manifolds generated by planar primitives form a hierarchical structure: The 2D manifold of straight edges, for example, is a subset of both the 3D manifold of bars and the 3D manifold of circular edges.

In [51], Lee et al. found that the state space of 3×3 natural image patches is extremely sparse with the patches densely concentrated around a non-linear manifold of blurred edges. For the top 20% highest contrast patches in natural images, half of the data lie within a neighborhood of the edge manifold that occupies only 9% of the total state space. Estimated probability density, as a function of the distance to the manifold, takes the form $f(\text{dist}) \sim 1/\text{dist}^\beta$ with $\beta = 2.5$; it has an

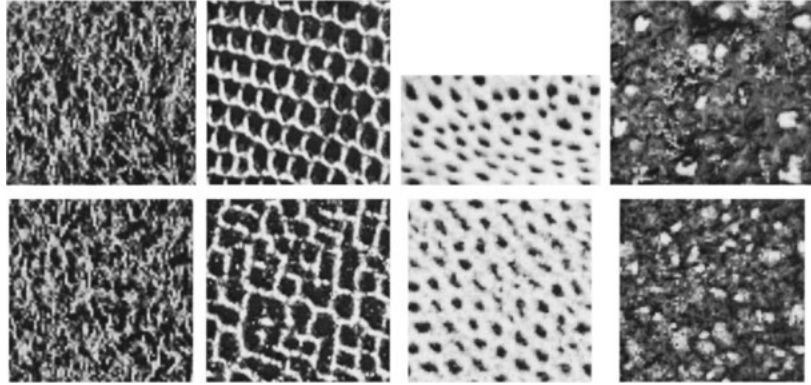


Figure 6. Texture synthesis using Julesz ensemble model (first two columns) and the Gibbs model (last two columns). Top panels: observed real images and bottom panels: synthesized images.

infinite density at the image manifold where $\text{dist} = 0$. The paper [64] extends these results by considering filtered patches with the filters being (up to third order) derivatives of Gaussian kernels. Figure 5 right panel shows the estimated probability density as a function of the distance to the edge manifold for multiscale representations of natural images. The density is approximately scale invariant and seems to converge toward the functional form $p(\text{dist}) \sim 1/\text{dist}^{0.7}$.

5. Applications of Statistical Models

Main reason for developing formal statistical models is to apply them in image processing/analysis applications. There is a large array of applications that continually benefit from advances in modeling of image statistics. Here we have selected an important subset.

5.1. Texture Synthesis

Use of newer statistical models has revolutionized the area of texture analysis. In 1980 Faugeras et al. [29] suggested using the marginals of filtered images for texture representations. Bergen and Landy [7], Chubb et al. [21], and Heeger and Bergen [41] also advocated the use of histograms. Zhu et al. [99] showed that marginal distributions of filtered images, obtained using a collection of filters, sufficiently characterize homogeneous textures. Choice of histograms implies that only the frequencies of occurrences are retained and the locations are discarded. Exploiting periodicity of these textures, one extracts features using wavelet decompositions at several scales and orientations, and uses them to represent images. Many schemes have

been proposed, two of them are: (i) Julesz ensemble that impose equivalence on all images that lead to the same histograms, and (ii) Gibbs model stated earlier in Section 3. Shown in Fig. 6 are some examples of texture synthesized under these models: the top row shows the real images and the bottom row shows corresponding synthesized textures. It must be noted that this synthesis framework holds well even when the raw histograms are replaced by parametric analytical forms, as shown in [80].

Beyond the use of marginals, Portilla and Simoncelli [65] developed a model based on a characterization of joint statistics of filter responses. Specifically, they measured the correlations of raw coefficients as well as the correlations of their magnitudes, and developed an efficient algorithm for synthesizing random images, subject to these constraints, by iteratively updating the image and forcing the constraints.

5.2. Image Compression

Compression seems to be a natural application for emerging statistical image models. In a typical implementation, the image is decomposed using a linear basis, the coefficients of this representation are quantized to discrete values, and these values are then efficiently encoded by taking advantage of their relative probability of occurrence. In this context, the statistical models considered here provide estimates of these probabilities. Most widely used compression scheme for images is the JPEG standard, which is based on a block-by-block frequency decomposition. In early 80s, it was recognized that multi-scale wavelet-style representations offer more flexibility and better compression

performance [15, 87]. Early coders were based on marginal models of coefficients, but this changed abruptly with the development of the first contextual coder by Shapiro [73]. This coder, and many that followed [20, 67], took heuristic advantage of the joint statistical properties of wavelets. Some subsequent coders have been based more explicitly on such models [14, 54].

5.3. Image Classification

An interesting application is to classify images into some pre-defined categories. If certain lower order statistics of images are found sufficient for this purpose, some level of efficiency can be achieved. We start the discussion with the classical methods for classification.

5.3.1. Multiple Hypothesis Testing. A classical technique for classification, given the probability models associated with the classes, is hypothesis selection [37]. Given an observation, the goal is to select the hypothesis that is most likely to have generated that observation. Let $H_1, H_2, \dots, H_n$ correspond to the n image classes, and let $P(I | H_i)$ be the likelihood of image I belonging to class H_i , then hypothesis selection can be accomplished as a sequence of binary hypothesis tests: $\frac{P(I|H_i)}{P(I|H_j)} \underset{H_j}{\overset{H_i}{\geq}} v_{ij}$, where v_{ij} is a threshold value generally taken to be one. Neyman-Pearson lemma provides optimality to hypothesis testing in a certain sense. In case of a Bayesian selection, the threshold v_{ij} is given by the ratio of priors $v_{i,j} = P(H_j)/P(H_i)$. For all these tests, one needs the likelihood function $P(I | H_i)$ which will depend upon the choice of statistical models. For example, if a parametric form is chosen to model the image probability, then the classes can be directly related to the parameter values. Typical values (“average values”) of the parameter represent a class and the likelihood $P(I | H_i)$ is written in the parametric form with the typical parameter values for each class. Deformable templates (Section 4), has been successfully applied to object recognition. The inference is obtained by hypothesis testing for α in presence of nuisance variables $s \in S$. Here, the likelihood is computed via the nuisance integral $P(I | H_i) = \int_S P(I, s | H_i) \gamma(ds)$. Nuisance variable estimation on a group S and estimation error bounds are derived in [35], while the hypothesis testing for α and recognition error bounds are derived in [37].

5.3.2. Metrics for Image Comparison. A broader goal in image analysis is to quantify differences

between two given images. Given such a metric one can perform image clustering, image retrieval, image classification, and even recognition of objects in given images. If the probability models are parametric, one can derive a distance measure on the image space that takes a parametric form. If $f(x | p_1, c_1)$ and $f(x | p_2, c_2)$ are two univariate density functions parameterized by (p_1, c_1) and (p_2, c_2) respectively, then the distance measure takes the form: $d(p_1, c_1, p_2, c_2) = \tilde{d}(f(x | p_1, c_1), f(x | p_2, c_2))$, where $\tilde{d}$ is a metric on the space of univariate densities. Several forms have been proposed for $\tilde{d}$ including geodesic length (Riemannian metric), Earth Mover’s distance [22], Kullback-Leibler divergence, Renyi’s α -divergence [42], Jensen-Renyi divergence [38, 40], χ^2 -distance, and the L^p norm for $p = 1, 2, \dots$. The choice of metric depends upon the application and the desired computational simplicity. Srivastava et al. [80] have derived a parametric form of L^2 metric between two Bessel K forms. Two densities can be compared directly using their estimated parameters without requiring to compute full histograms.

5.4. Image Denoising

A common approach here is to decompose image into bands of spatial frequency and to threshold the coefficients after some nonlinear transformation, proposed first by Bayer and Powell [4] and later by Donoho [26]. The nonlinear transformation is used essentially to shrink all wavelet coefficients towards zero. This shrinking is based on thresholding which can be implemented as a hard threshold or a soft threshold. For applying statistical approaches, such as Bayesian or MAP techniques, an explicit prior model on the image pixels is required. Simoncelli and Adelson [76] studied image denoising in a Bayesian framework while Moulin and Liu [60] reported the use of generalized Laplacian models in statistical denoising approaches. The fact that statistics of decomposed images are easier to characterize, than that of the original images, has led to many pyramid-based approaches to image denoising. Images are decomposed into multi-scale representations and statistics of coefficients are used for denoising [53, 66, 75] in individual frequency bands. Also, the Gaussian scale mixture model for joint statistics can be used in a Bayesian framework [66], producing results that are significantly better than those achieved with a marginal model.

5.5. Other Applications

There are a number of other image analysis/synthesis applications that have benefited from statistical ideas. Yendrikhovskij [94] uses a clustering of color statistics to model the perceived color environment and to compute categories of coloring. A bidirectional radiosity distribution function (BRDF) of an object completely specifies its appearance under arbitrary illumination conditions. Dror et al. [28] have shown that statistics of these illumination maps are similar to the statistics of natural images, and hence the proposed image models can apply there as well [92].

6. Discussion

In this paper we have discussed some recent advances in statistical modeling of natural images. Not only these models provide a better match than the traditional models but also have lead to significant improvements in a number of imaging applications.

Although substantial progress has been made over the past twenty years in understanding complex statistical properties of natural images, we are still quite far from a full probability model. For example, samples drawn from existing models are unable to capture the variety and complexity of real world images, except in the restricted case of homogeneous textures. Beyond univariate and bivariate densities of the image statistics, the computational complexity increases exponentially. An important question is: In the context of a specific imaging application, say face recognition from video images, what order densities are required to ensure a reasonable success? For homogeneous textures, univariate models have been successful but how much more is needed for more general applications. The notion of sufficient statistics needs to be made precise for different application contexts. Even among the proposed models, several issues remain open. For instance, one issue in dead leaves model is how to incorporate texture and dependencies of objects. At this point, there are few analytical results for realistic dead leaves models.

Many of the models described in this paper model statistics of an ensemble of images; their applications for analysis of individual images needs to be clarified.

Beyond applications, such as synthesis and compression of images, an important reason for developing statistical models is image understanding, an area

with many outstanding problems. One such problem is: Given an image outside a rectangle, find the most probable interpolation of it inside of the rectangle. Lack of Markovity disallows the classical harmonic analysis, and points to more powerful pattern-theoretic structures underlying the image ensembles.

Acknowledgments

We thank the creators of van Hateren database for making their database available to public. We express gratitude to our collaborators Ulf Grenander, Kim S. Pedersen and David Mumford for their help in this research. Many thanks to Hamid Krim for his invitation and support in writing this paper. AS has been supported in part by the grants NSF DMS-0101429, ARO DAAD19-99-1-0267, and NMA 201-01-2010. EPS was supported in part by NSF CAREER grant MIP-9796040, the Alfred P. Sloan Foundation, and the Howard Hughes Medical Institute.

Note

1. see [46] for a discussion on “forward” and “backward” dead leaves algorithms.

References

1. L. Alvarez, Y. Gousseau, and J.-M. Morel, “The size of objects in natural and artificial images,” in *Advances in Imaging and Electron Physics*, P.W.H. et al. (Eds.), Academic Press: New York, 1999.
2. D. Andrews and C. Mallows, “Scale mixtures of normal distributions,” *Journal of the Royal Statistical Society*, Vol. 36, pp. 99–102, 1974.
3. O. Barndorff-Nielsen, “Exponentially decreasing distribution for the logarithm of a particle size,” *Journal of Royal Statistical Society A*, Vol. 353, pp. 401–419, 1977.
4. B.E. Bayer and P.G. Powell, “A method for the digital enhancement of unsharp, grainy photographic images,” *Advances in Computer Vision and Image Processing*, Vol. 2, pp. 31–88, 1986.
5. P.N. Belhumeur, J.P. Hefanpha, and D.J. Kriegman, “Eigenfaces vs. fisherfaces: Recognition using class specific linear projection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 7, pp. 711–720, 1997.
6. A.J. Bell and T.J. Sejnowski, “The independent components of natural scenes are edge filters,” *Vision Research*, Vol. 37, No. 23, pp. 3327–3338, 1997.
7. J.R. Bergen and M.S. Landy, *Computational Modeling of Visual Texture Segregation*, MIT Press: Cambridge, MA, 1991, pp. 253–271.

8. J. Besag, "Spatial interaction and statistical analysis of lattice systems (with discussion)," *J. Royal Statist. Soc. B*, Vol. 36, pp. 192–326, 1974.
9. J. Besag, "On the statistical analysis of dirty pictures," *J. Royal Statistical Society B*, Vol. 48, No. 3, pp. 259–302, 1986.
10. P. Billingsley, *Probability and Measure*, Wiley Series in Probability and Mathematical Statistics, 1995.
11. D. Bitouk, U. Grenander, M.I. Miller, and P. Tyagi, "Fisher information measures for ATR in clutter," in *Proceedings of SPIE, Automatic Target Recognition XI*, 2001, Vol. 4379, pp. 560–572.
12. T. Bollerslev, K. Engle, and D. Nelson, *ARCH Models*, Vol. IV, North-Holland: Amsterdam, 1994, pp. 2959–3038.
13. H. Brehm and W. Stammers, "Description and generation of spherically invariant speech-model signals," *Signal Processing*, Vol. 12, pp. 119–141, 1987.
14. R.W. Buccigrossi and E.P. Simoncelli, "Image compression via joint statistical characterization in the wavelet domain," *IEEE Transactions on Image Processing*, Vol. 8, No. 12, pp. 1688–1701, 1999.
15. P.J. Burt and E.H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Transactions on Communications*, Vol. 31, No. 4, pp. 532–540, 1983.
16. G.J. Burton, N.D. Haig, and I.R. Moorhead, "A self-similar stack model for human and machine vision," *Biological Cybernetics*, Vol. 53, No. 6, pp. 397–403, 1986.
17. G.J. Burton and I.R. Moorhead, "Color and spatial structures in natural scenes," *Applied Optics*, Vol. 26, No. 1, pp. 157–170, 1987.
18. J.-F. Cardoso, "Source separation using higher order moments," in *Proceedings of ICASSP*, 1989, pp. 2109–2112.
19. Z. Chi, "Probability models for complex systems," Ph.D. Thesis, Division of Applied Mathematics, Brown University, 1998.
20. C. Chrysafis and A. Ortega, "Efficient context-based entropy coding for lossy wavelet image coding," in *Data Compression Conf.*, 1997, Snowbird, Utah.
21. C. Chubb, J. Econopouly, and M.S. Landy, "Histogram contrast analysis and the visual segregation of IID textures," *J. Opt. Soc. Am. A*, Vol. 11, pp. 2350–2374, 1994.
22. S. Cohen and L. Guibas, "The earth mover's distance under transformation sets," in *Proceedings of Seventh IEEE International Conference on Computer Vision*, Vol. 2, pp. 1076–1083, 1999.
23. P. Comon, "Independent component analysis, a new concept?" *Signal Processing*, Special Issue on Higher-Order Statistics, Vol. 36, No. 3, 1994.
24. J. Daugman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *Journal of the Optical Society of America A*, Vol. 2, No. 7, pp. 23–26, 1985.
25. N.G. Deriugin, "The power spectrum and the correlation function of the television signal," *Telecommunications*, Vol. 1, pp. 1–12, 1956.
26. D. Donoho, "Denoising by soft-thresholding," *IEEE Trans. Info. Theory*, Vol. 43, pp. 613–627, 1995.
27. D.L. Donoho and A.G. Flesia, "Can recent innovations in harmonic analysis 'Explain' key findings in natural image statistics," *Network: Computation in Neural Systems*, Vol. 12, No. 3, pp. 371–393, 2001.
28. R.O. Dror, T.K. Leung, E.H. Adelson, and A.S. Willsky, "Statistics of real-world illumination," in *Proceedings of 2001 IEEE Conference on CVPR*, 2001, Vol. 2, pp. 164–171.
29. O.D. Faugeras and W.K. Pratt, "Decorrelation methods of texture feature extraction," *IEEE Pat. Anal. Mach. Intell.* Vol. 2, No. 4, pp. 323–332, 1980.
30. D.J. Field, "Relations between the statistics of natural images and the response properties of cortical cells," *J. Optical Society of America*, Vol. 4, No. 12, pp. 2379–2394, 1987.
31. D. Gabor, "Theory of Communications," *Journal of IEE (London)*, Vol. 93, pp. 429–457, 1946.
32. D. Geman and A. Koloydenko, "Invariant statistics and coding of natural microimages," in *Proc. of the IEEE Workshop on Statistical and Computational Theories of Vision*, 1999.
33. S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 6, No. 6, pp. 721–741, 1984.
34. U. Grenander, *General Pattern Theory*, Oxford University Press, 1993.
35. U. Grenander, M.I. Miller, and A. Srivastava, "Hilbert-Schmidt lower bounds for estimators on matrix Lie groups for ATR," *IEEE Transactions on PAMI*, Vol. 20, No. 8, pp. 790–802, 1998.
36. U. Grenander and A. Srivastava, "Probability models for clutter in natural images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 4, pp. 424–429, 2001.
37. U. Grenander, A. Srivastava, and M.I. Miller, "Asymptotic performance analysis of Bayesian object recognition," *IEEE Transactions of Information Theory*, Vol. 46, No. 4, pp. 1658–1666, 2000.
38. A.B. Hamza, Y. He, and H. Krim, "An information divergence measure for ISAR image registration," in *Proc. of IEEE Workshop on Statistical Signal Processing*, 2001.
39. P.J.B. Hancock, R.J. Baddeley, and L.S. Smith, "The principal components of natural images," *Network*, Vol. 3, pp. 61–70, 1992.
40. Y. He, A.B. Hamza, and H. Krim, "A generalized divergence measure for robust image registration," *IEEE Transactions on Signal Processing*, 2002, to appear.
41. D.J. Heeger and J.R. Bergen, "Pyramid-based texture analysis/synthesis," in *Proceedings of SIGGRAPH*, 1995, pp. 229–238.
42. A.O. Hero, B. Ma, O. Michel, and J. Gorman, "Alpha-divergence for classification, indexing and retrieval," Communication and Signal Processing Laboratory, Technical Report CSPL-328, U. Mich, 2001.
43. J. Huang, "Statistics of natural images and models," Ph.D. Thesis, Division of Applied Mathematics, Brown University, RI, 2000.
44. A. Hyvarinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, John Wiley and Sons; New York, 2001.
45. D. Jeulin, I. Terol-Villalobos, and A. Dubus, "Morphological analysis of UO2 powder using a dead leaves model," *Microscopy, Microanalysis, Microstructure*, Vol. 6, pp. 371–384, 1995.
46. W.S. Kendall and E. Thonnes, "Perfect simulation in stochastic geometry," Preprint 323, Department of Statistics, University of Warwick, UK, 1998.
47. D. Kersten, "Predictability and redundancy of natural images," *Journal of Optical Society of America*, Vol. 4, No. 12, pp. 2395–2400, 1987.
48. M. Kirby and L. Sirovich, "Application of the Karhunen-Loeve procedure for the characterization of human faces," *IEEE*

- Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 1, pp. 103–108, 1990.
49. E.R. Kretzmer, “Statistics of television signals,” *Bell Syst. Tech. Journal*, Vol. 31, pp. 751–763, 1952.
 50. A.B. Lee and D. Mumford, “Occlusion models for natural images: A statistical study of scale-invariant dead leaves model,” *International Journal of Computer Vision*, Vol. 41, No. 1/2, 2001.
 51. A.B. Lee, K.S. Pedersen, and D. Mumford, “The nonlinear statistics of high-contrast patches in natural images,” *Int. J. Computer Vision*, 2002, in press.
 52. D.D. Lee and H.S. Seung, “Learning the parts of objects by non-negative matrix factorization,” *Nature*, Vol. 401, pp. 788–791, 1999.
 53. D. Leporini, J.-C. Pesquet, and H. Krim, “Best basis representations with prior statistical models,” in *Lecture Notes in Statistics: Bayesian Inferences in Wavelet Based Models*, P. Muller and B. Vidakovic (Eds.), Springer Verlag: Berlin, 1999.
 54. S.M. LoPresto, K. Ramchandran, and M.T. Orchard, “Wavelet image coding based on a new generalized gaussian mixture model,” in *Data Compression Conf.*, 1997, Snowbird, Utah.
 55. S.G. Mallat, “A Theory for multiresolution signal decomposition: The wavelet representation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 11, pp. 674–693, 1989.
 56. D. Marr, *Vision: A computational Investigation into the Human Representation and Processing of Visual Information*, W.H. Freeman and Company: New York, 1982.
 57. G. Matheron, *Random Sets and Integral Geometry*, John Wiley and Sons: New York, 1975.
 58. K.D. Miller, “A Model for the development of simple-cell receptive fields and the arrangement of orientation columns through activity dependent competition between on- and off-center inputs,” *Journal of Neuroscience*, Vol. 14, pp. 409–441, 1994.
 59. M.I. Miller and L. Younes, “Group actions, homeomorphisms, and matching: A general framework,” *International Journal of Computer Vision*, Vol. 41, No. 1/2, pp. 61–84, 2002.
 60. P. Moulin and J. Liu, “Analysis of multiresolution image denoising schemes using a generalized Gaussian and complexity priors,” *IEEE Trans. Info. Theory*, Vol. 45, pp. 909–919, 1999.
 61. D. Mumford and B. Gidas, “Stochastic models for generic images,” *Quarterly of Applied Mathematics*, Vol. 59, No. 1, pp. 85–111, 2001.
 62. B.A. Olshausen and D.J. Field, “Emergence of simple-cell receptive field properties by learning a sparse code for natural images,” *Nature*, Vol. 381, pp. 607–609, 1996.
 63. B.A. Olshausen and D.J. Field, “Sparse coding with an overcomplete basis set: A strategy employed by V1?” *Vision Research*, Vol. 37, No. 23, pp. 3311–3325, 1997.
 64. K.S. Pedersen and A.B. Lee, “Toward a full probability model of edges in natural images,” in *Proc. of ECCV’02*, 2002.
 65. J. Portilla and E.P. Simoncelli, “A parametric texture model based on joint statistics of complex wavelet coefficients,” *International Journal of Computer Vision*, Vol. 40, No. 1, pp. 49–70, 2000.
 66. J. Portilla, V. Strela, M. Wainwright, and E. Simoncelli, “Adaptive Wiener denoising using a Gaussian scale mixture model in the wavelet domain,” in *Proc 8th IEEE Int’l Conf on Image Proc.* Thessaloniki, Greece, IEEE Computer Society, 2001, pp. 37–40.
 67. R. Rinaldo and G. Calvagno, “Image coding by block prediction of multiresolution subimages,” *IEEE Trans Im Proc.*, 1995.
 68. S.T. Roweis and L.K. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” *Science*, Vol. 290, pp. 2323–2326, 2000.
 69. D.L. Ruderman, “The statistics of natural images,” *Network*, Vol. 5, pp. 517–548, 1994.
 70. D.L. Ruderman, “Origins of scaling in natural images,” *Vision Research*, Vol. 37, No. 23, pp. 3385–3398, 1997.
 71. D.L. Ruderman and W. Bialek, “Scaling of natural images: Scaling in the woods,” *Physical Review Letters*, Vol. 73, No. 6, pp. 814–817, 1994.
 72. J.P. Serra, *Image Analysis and Mathematical Morphology*, Academic Press: London, 1982.
 73. J. Shapiro, “Embedded image coding using zerotrees of wavelet coefficients,” *IEEE Trans. Sig. Proc.*, Vol. 41, No. 12, pp. 3445–3462, 1993.
 74. E.P. Simoncelli, “Statistical models for images: Compression, restoration and synthesis,” in *Proc. 31st Asilomar Conf on Signals, Systems and Computers*, Pacific Grove, CA. IEEE Computer Society, 1997, pp. 673–678.
 75. E.P. Simoncelli, “Bayesian denoising of visual images in the wavelet domain,” in *Bayesian Inference in Wavelet Based Models*, Lecture Notes in Statistics, Vol. 41, Springer Verlag: Berlin, 1999.
 76. E.P. Simoncelli and E.H. Adelson, “Noise removal via Bayesian wavelet coring,” in *Third Int’l. Conf on Image Proc.*, Lausanne, IEEE Sig Proc Society, 1996, Vol. 1, pp. 379–382.
 77. E.P. Simoncelli and R.T. Buccigrossi, “Embedded wavelet image compression based on a joint probability model,” in *Proceedings of ICIP (1)*, 1997, pp. 640–643.
 78. E.P. Simoncelli, W.T. Freeman, E.H. Adelson, and D.J. Heeger, “Shiftable multi-scale transforms,” *IEEE Trans Information Theory*, Vol. 38, No. 2, pp. 587–607, 1992.
 79. E. Simoncelli and B. Olshausen, “Natural image statistics and neural representation,” *Annual Review of Neuroscience*, Vol. 24, pp. 1193–1216, 2001.
 80. A. Srivastava, X. Liu, and U. Grenander, “Universal analytical forms for modeling image probability,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 28, No. 9, 2002.
 81. J.B. Tenenbaum, V. Silva, and J.C. Langford, “A global geometric framework for nonlinear dimensionality reduction,” *Science*, Vol. 290, pp. 2319–2323, 2000.
 82. M.G.A. Thomson, “Beats, kurtosis, and visual coding,” *Network: Computation in Neural Systems*, Vol. 12, No. 3, pp. 271–287, 2001.
 83. A. Turiel, G. Mato, and N. Parga, “The self-similarity properties of natural images resemble those of turbulent flows,” *Physical Review Letters*, Vol. 80, No. 5, pp. 1098–1101, 1998.
 84. A. Turiel and N. Parga, “The multi-fractal structure of contrast changes in natural images: From sharp edges to textures,” *Neural Computation*, Vol. 12, pp. 763–793, 2000.
 85. A. Turiel, N. Parga, D.L. Ruderman, and T.W. Cronin, “Multi-scaling and information content of natural color images,” *Physical Review E*, Vol. 62, No. 1, pp. 1138–1148, 2000.
 86. J.A. van Hateren, “Independent component filters of natural images compared with simple cells in primary visual cortex,” *Proc. Royal Statistical Society of London B*, Vol. 265, pp. 359–366, 1998.
 87. M. Vetterli, “Multidimensional subband coding: Some theory and algorithms,” *Signal Processing*, Vol. 6, No. 2, pp. 97–112, 1984.

88. M.J. Wainwright and E.P. Simoncelli, "Scale mixtures of Gaussians and the statistics of natural images," *Advances in Neural Information Processing Systems*, S.A. Solla, T.K. Leen, and K.-R. Muller (Eds.), 2000, pp. 855–861.
89. M.J. Wainwright, E.P. Simoncelli, and A.S. Willsky, "Random cascades on wavelet trees and their use in modeling and analyzing natural imagery," *Applied and Computational Harmonic Analysis*, Vol. 11, No. 1, pp. 89–123, 2001.
90. A.B. Watson, "The cortex transform: Rapid computation of simulated neural images," *Comp. Vis. Graphics Image Proc.*, Vol. 39, pp. 311–327, 1987.
91. B. Wegmann and C. Zetsche, "Statistical dependence between orientation filter outputs used in a human vision based image code," in *Proceedings of Visual Communication and Image Processing*, Society of Photo-Optical Instrumentation Engineers, Vol. 1360, pp. 909–922, 1990.
92. Y. Weiss, "Deriving intrinsic images from image sequences," in *Proceedings Eight IEEE International Conference on Computer Vision*, 2001, Vol. 2, pp. 68–75.
93. G. Winkler, *Image Analysis, Random Fields and Dynamic Monte Carlo Methods*, Springer: Berlin, 1995.
94. S.N. Yendrikhovskij, "Computing color categories from statistics of natural images," *Journal of Imaging Science and Technology*, Vol. 45, No. 5, pp. 409–417, 2001.
95. C. Zetsche, "Polyspectra of natural images," in *Presented at Natural Scene Statistics Meeting*, 1997.
96. C. Zetsche and F. Rohrbein, "Nonlinear and extra-classical receptive field properties and the statistics of natural images," *Network: Computation in Neural Systems*, Vol. 12, No. 3, pp. 331–350, 2001.
97. S.C. Zhu, C.E. Guo, Y.N. Wu, and Y.Z. Wang, "What are textures?," in *Proc. of European Conf. on Computer Vision*, 2002.
98. S.C. Zhu and D. Mumford, "Prior learning and Gibbs reaction-diffusion," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 19, No. 11, pp. 1236–1250, 1997.
99. S.C. Zhu, Y.N. Wu, and D. Mumford, "Minimax entropy principles and its application to texture modeling," *Neural Computation*, Vol. 9, No. 8, pp. 1627–1660, 1997.



Anuj Srivastava is an Assistant Professor in Department of Statistics at Florida State University. He obtained his Ph.D. in electrical engineering from Washington University in 1996 and was a visiting researcher in Division of Applied Mathematics at Brown University during the year 1996–1997. He directs the Laboratory for Computational Vision at FSU that engages in research on image understanding using multiple imaging modalities. His research interests lie in the areas of statistical signal processing and image analysis, statistics on manifolds, and computational statistics. His research has been supported by ARO, NSF, NIMA, and FSU research funds.



Ann Lee received her M.Sc. degree in Engineering Physics at Chalmers University of Technology in Sweden, and her Ph.D. degree in Physics at Brown University. She is currently a visiting research associate in the Division of Applied Mathematics at Brown University and a research instructor in the Mathematics Department at Yale University. Her research is in the areas of machine and biological vision, statistical modeling, and pattern theory.



Eero Simoncelli is an Associate Professor of Neural Science and Mathematics at New York University. He received the B.A. in physics from Harvard University, studied Mathematics for one and a half years at Cambridge University, and then entered the Ph.D. program in electrical engineering and computer science at MIT. He joined the faculty of the Computer and Information Science department at U Pennsylvania in January 1993. In September of 1996, he moved to New York University as part of the Sloan Center for theoretical visual neuroscience. He received an NSF Faculty Early Career Development (CAREER) grant in 1996 for research and teaching in "Visual Information Processing", and a Sloan Research Fellowship in 1998. In August 2000, he became an Associate Investigator of the Howard Hughes Medical Institute, under their new program in Computational Biology.



Song Chun Zhu received his B.S. degree in computer science from the University of Science and Technology of China in 1991. He

received his M.S. and Ph.D. degrees in computer science from Harvard University (Harvard Robotics Lab) in 1994 and 1996 respectively. He was a research associate in the Division of Applied Math. (Pattern Theory Group) at Brown University during 1996–1997, a lecturer in the Computer Science Department at Stanford University during 1997–1998, and an assistant professor in the Department of Computer and Information Sciences at Ohio State University from

1998–2002. He is currently an associate professor in the Departments of Statistics and Computer Science jointly at University of California, Los Angeles. His research is focused on computer vision and learning, statistical modeling, and stochastic computing. He has published over 50 articles, and received various honors including a David Marr prize honorary mention, a NSF Career award, a Sloan Fellow, and a Navy Young Investigator Award.