

FIGURA 2

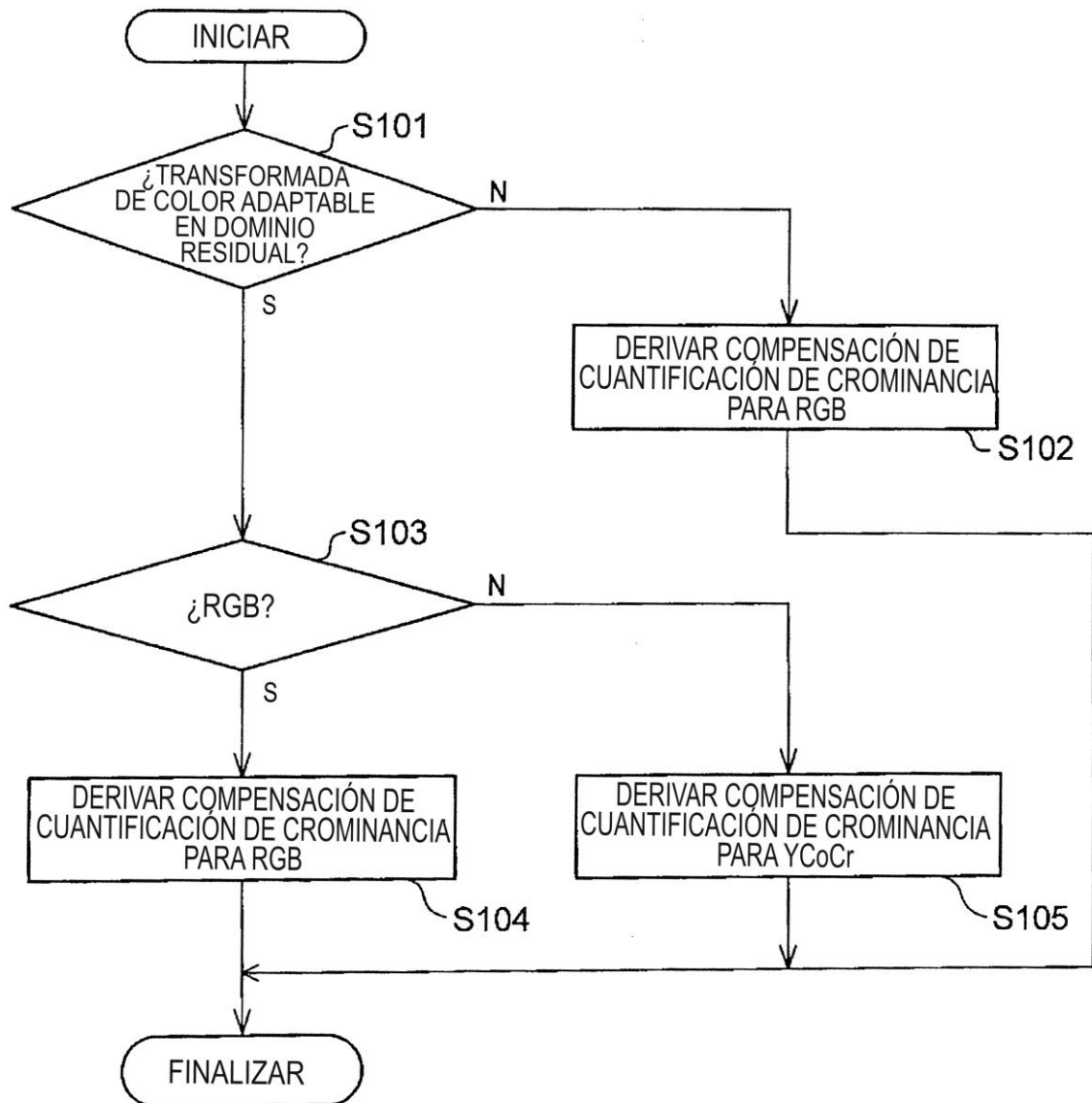


FIGURA 3

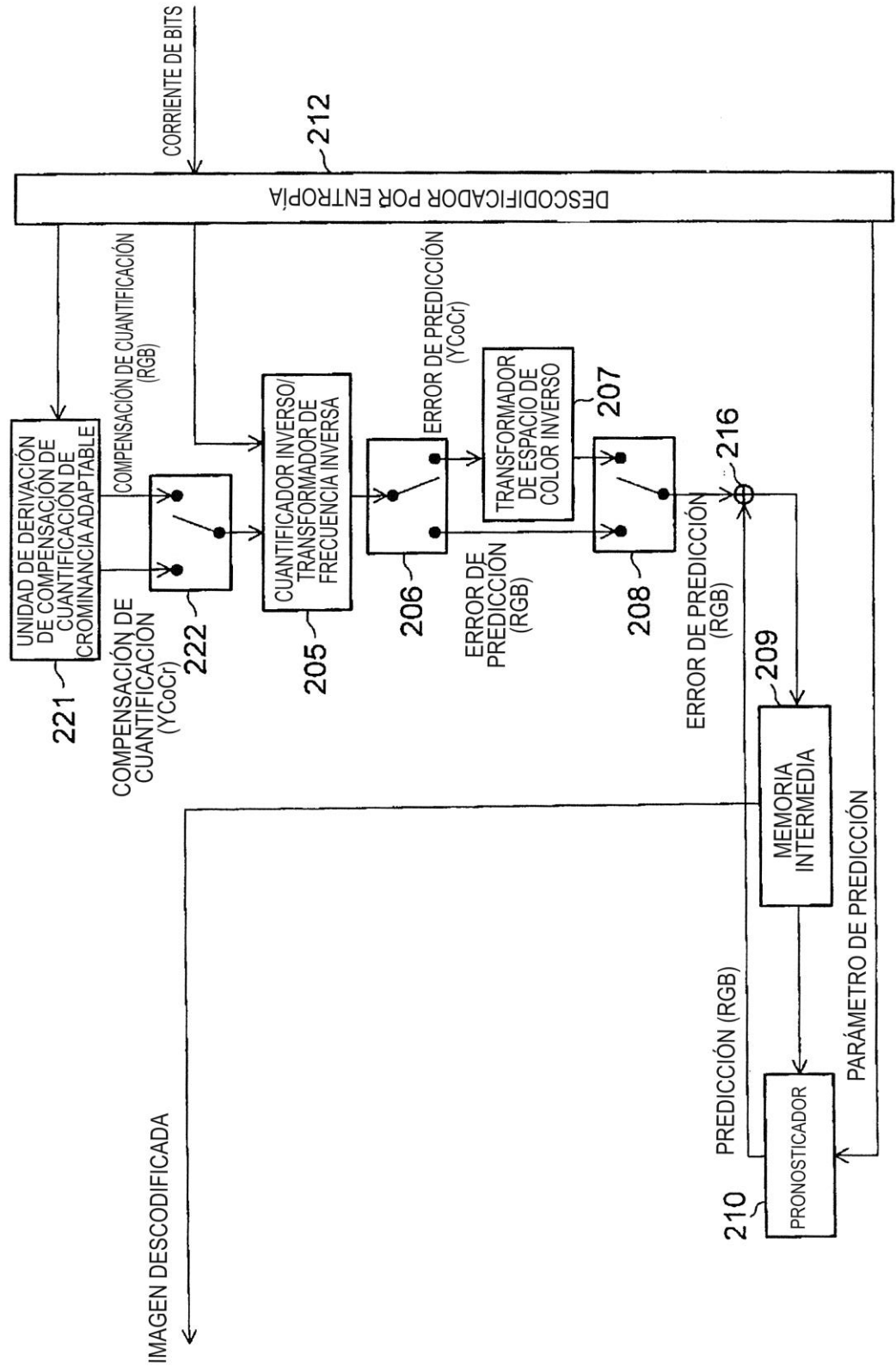


FIGURA 4

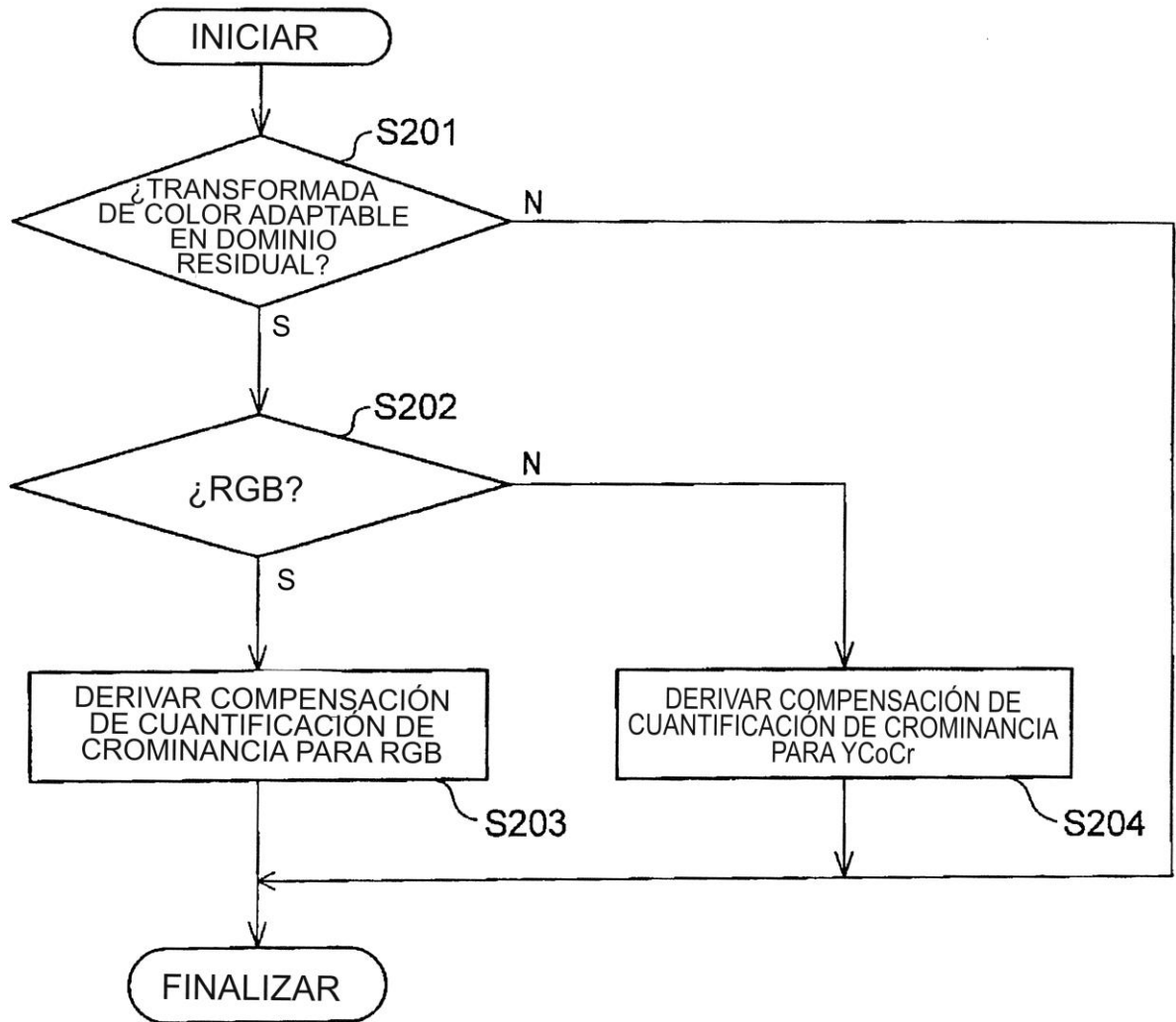


FIGURA 5

pps_range_extensions() {	Descriptor
if(transform_skip_enabled_flag)	
log2_max_transform_skip_block_size_minus2	ue(v)
cross_component_prediction_enabled_flag	u(1)
chroma_qp_offset_list_enabled_flag	u(1)
if(chroma_qp_offset_list_enabled_flag) {	
diff_cu_chroma_qp_offset_depth	ue(v)
chroma_qp_offset_list_len_minus1	ue(v)
for(i = 0; i <= chroma_qp_offset_list_len_minus1; i++) {	
cb_qp_offset_list[i]	se(v)
cr_qp_offset_list[i]	se(v)
}	
}	
log2_sao_offset_scale_luma	ue(v)
log2_sao_offset_scale_chroma	ue(v)
if(adaptive_color_trans_flag) {	
alt_pps_cb_qp_offset	se(v)
alt_pps_cr_qp_offset	se(v)
}	
}	

alt_pps_cb_qp_offset y ***alt_pps_cr_qp_offset*** especifican compensaciones para el parámetro de cuantificación de luminancia Qp'Y utilizado para derivar Qp'Cb y Qp'Cr, respectivamente. Los valores de *alt_pps_cb_qp_offset* y *alt_pps_cr_qp_offset* deben encontrarse en el margen de -12 a +12, incluso. Cuando ChromaArrayType es igual a 0, *alt_pps_cb_qp_offset* y *alt_pps_cr_qp_offset* no se utilizan en el proceso de decodificación y los decodificadores deberán

FIGURA 6

slice_segment_header() {	Descriptor
first_slice_segment_in_pic_flag	u(1)
if(nal_unit_type >= BLA_W_LP && nal_unit_type <= RSV_IRAP_VCL23)	
no_output_of_prior_pics_flag	u(1)
slice_pic_parameter_set_id	ue(v)
if(!first_slice_segment_in_pic_flag) {	
if(dependent_slice_segments_enabled_flag)	
dependent_slice_segment_flag	u(1)
slice_segment_address	u(v)
}	
if(!dependent_slice_segment_flag) {	
for(i = 0; i < num_extra_slice_header_bits; i++)	
slice_reserved_flag[i]	u(1)
slice_type	ue(v)
if(output_flag_present_flag)	
pic_output_flag	u(1)
if(separate_colour_plane_flag == 1)	
colour_plane_id	u(2)
if(nal_unit_type != IDR_W_RADL && nal_unit_type != IDR_N_LP) {	
slice_pic_order_cnt_lsb	u(v)
short_term_ref_pic_set_sps_flag	u(1)
if(!short_term_ref_pic_set_sps_flag)	
st_ref_pic_set(num_short_term_ref_pic_sets)	
else if(num_short_term_ref_pic_sets > 1)	
short_term_ref_pic_set_idx	u(v)
if(long_term_ref_pics_present_flag) {	
if(num_long_term_ref_pics_sps > 0)	
num_long_term_sps	ue(v)
num_long_term_pics	ue(v)
for(i = 0; i < num_long_term_sps + num_long_term_pics; i++) {	
if(i < num_long_term_sps) {	
if(num_long_term_ref_pics_sps > 1)	
lt_idx_sps[i]	u(v)
} else {	

FIGURA 7

slice_qp_delta	se(v)
if(pps_slice_chroma_qp_offsets_present_flag) {	
slice_cb_qp_offset	se(v)
slice_cr_qp_offset	se(v)
if(adaptive_color_trans_flag) {	
alt_slice_cb_qp_offset	se(v)
alt_slice_cr_qp_offset	se(v)
}	
}	

alt_slice_cb_qp_offset especifica una diferencia que se agrega al valor de alt_pps_cb_qp_offset cuando se determina el valor del parámetro de cuantificación de Qp'Cb. El valor de alt_slice_cb_qp_offset deberá encontrarse en el margen de -12 a +12, inclusive. Cuando alt_slice_cb_qp_offset no se presenta, se infiere que es igual a 0. El valor de alt_pps_cb_qp_offset + alt_slice_cb_qp_offset deberá encontrarse en el margen de -12 a +12, inclusive.

alt_slice_cr_qp_offset especifica una diferencia que se agrega al valor de alt_pps_cr_qp_offset cuando se determina el valor del parámetro de cuantificación de Qp'Cb. El valor de alt_slice_cr_qp_offset deberá encontrarse en el margen de -12 a +12, inclusive. Cuando alt_slice_cr_qp_offset no se presenta, se infiere que es igual a 0. El valor de alt_pps_cr_qp_offset + alt_slice_cr_qp_offset deberá encontrarse en el margen de -12 a +12, inclusive.

FIGURA 8

pps_range_extensions() {	Descriptor
if(transform_skip_enabled_flag)	
log2_max_transform_skip_block_size_minus2	ue(v)
cross_component_prediction_enabled_flag	u(1)
chroma_qp_offset_list_enabled_flag	u(1)
if(chroma_qp_offset_list_enabled_flag) {	
diff_cu_chroma_qp_offset_depth	ue(v)
chroma_qp_offset_list_len_minus1	ue(v)
for(i = 0; i <= chroma_qp_offset_list_len_minus1; i++) {	
cb_qp_offset_list[i]	se(v)
cr_qp_offset_list[i]	se(v)
}	
if(adaptive_color_trans_flag) {	
for(i = chroma_qp_offset_list_len_minus1+1; i <=	
2*chroma_qp_offset_list_len_minus1; i++) {	
cb_qp_offset_list[i]	se(v)
cr_qp_offset_list[i]	se(v)
}	
}	
log2_sao_offset_scale_luma	ue(v)
log2_sao_offset_scale_chroma	ue(v)
}	

FIGURA 9

pps_range_extensions() {	Descriptor
if(transform_skip_enabled_flag)	
log2_max_transform_skip_block_size_minus2	ue(v)
cross_component_prediction_enabled_flag	u(1)
chroma_qp_offset_list_enabled_flag	u(1)
if(chroma_qp_offset_list_enabled_flag) {	
diff_cu_chroma_qp_offset_depth	ue(v)
chroma_qp_offset_list_len_minus1	ue(v)
for(i = 0; i <= chroma_qp_offset_list_len_minus1; i++) {	
cb_qp_offset_list[i]	se(v)
cr_qp_offset_list[i]	se(v)
}	
if(adaptive_color_trans_flag) {	
for(i = 0; i <= chroma_qp_offset_list_len_minus1; i++) {	
alt_cb_qp_offset_list[i]	se(v)
alt_cr_qp_offset_list[i]	se(v)
}	
}	
log2_sao_offset_scale_luma	ue(v)
log2_sao_offset_scale_chroma	ue(v)
}	

FIGURA 10

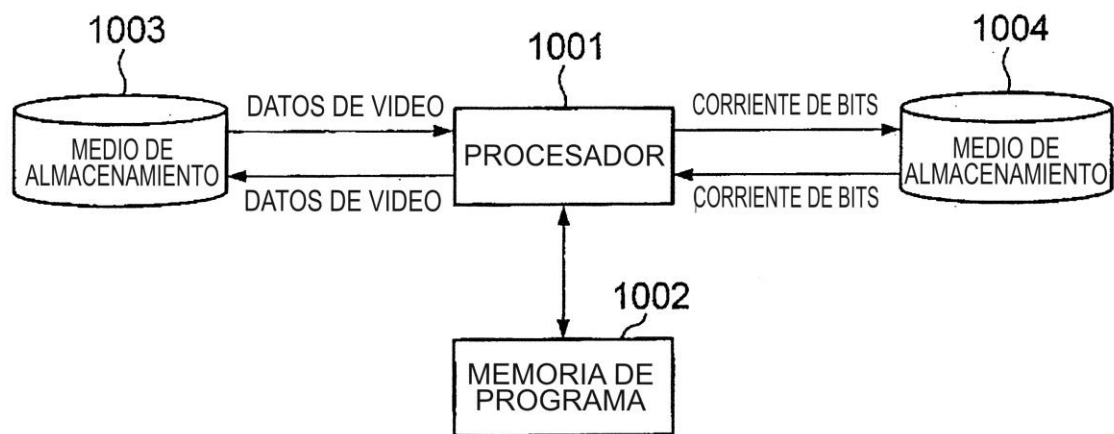


FIGURA 11

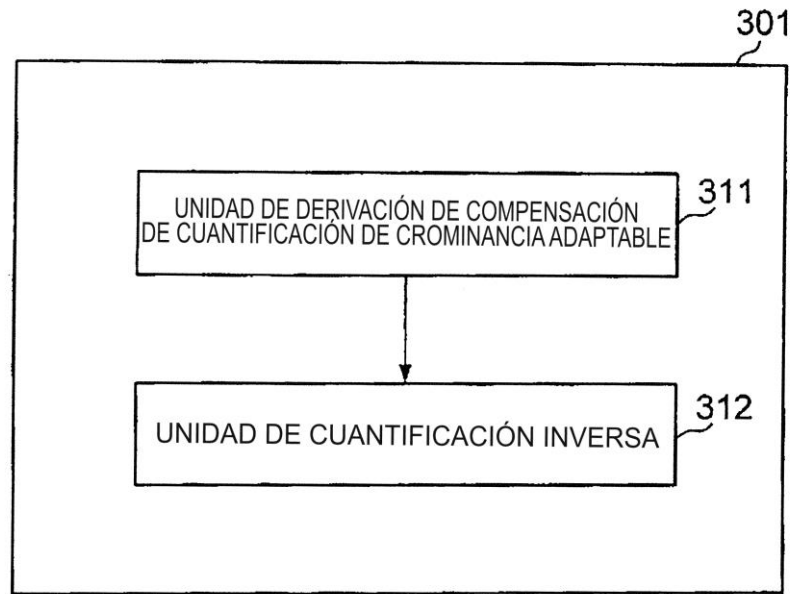


FIGURA 12

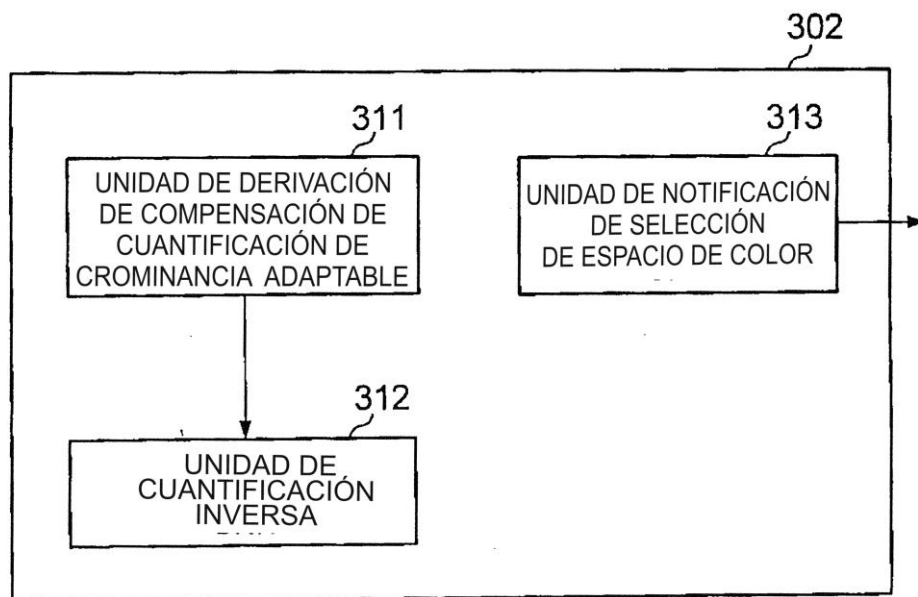


FIGURA 13

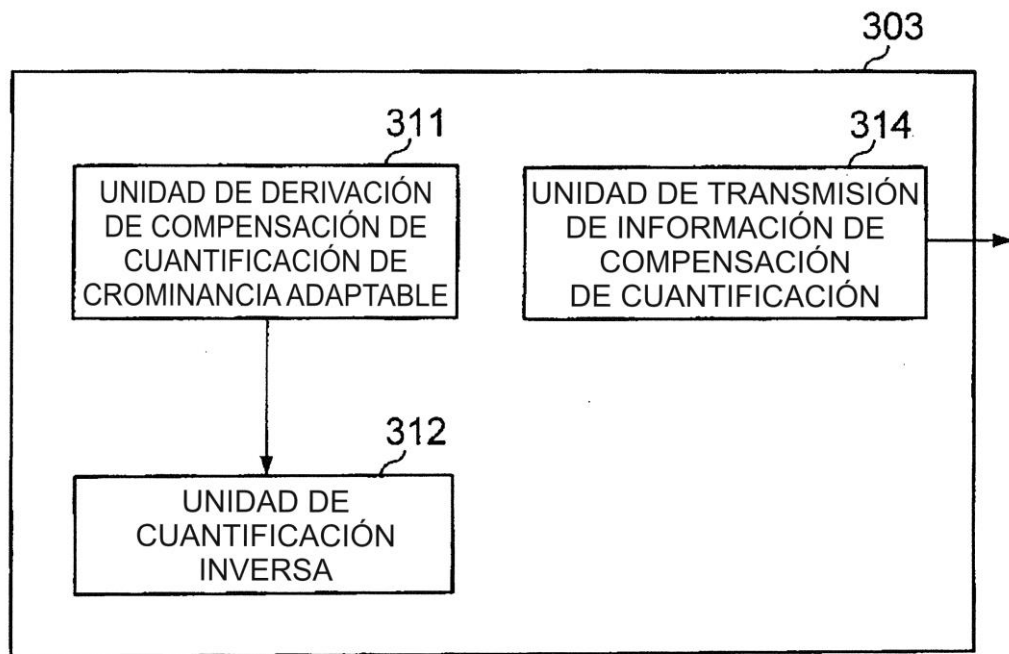


FIGURA 14

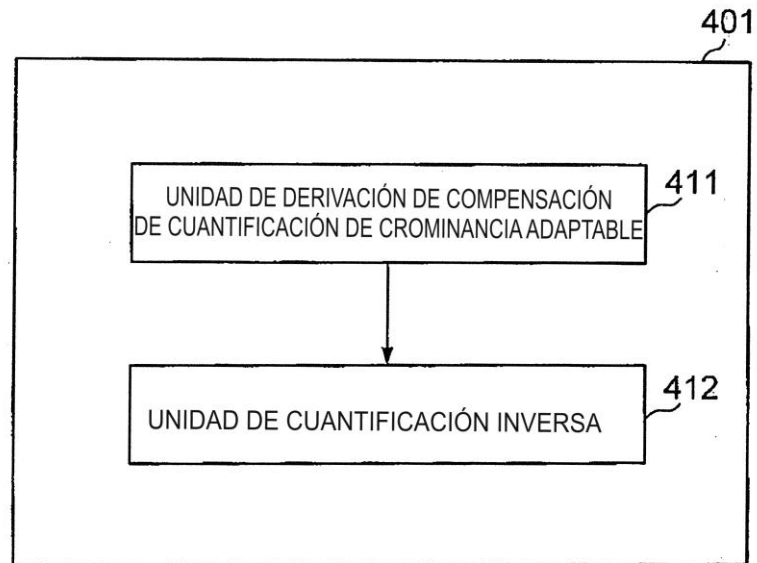


FIGURA 15

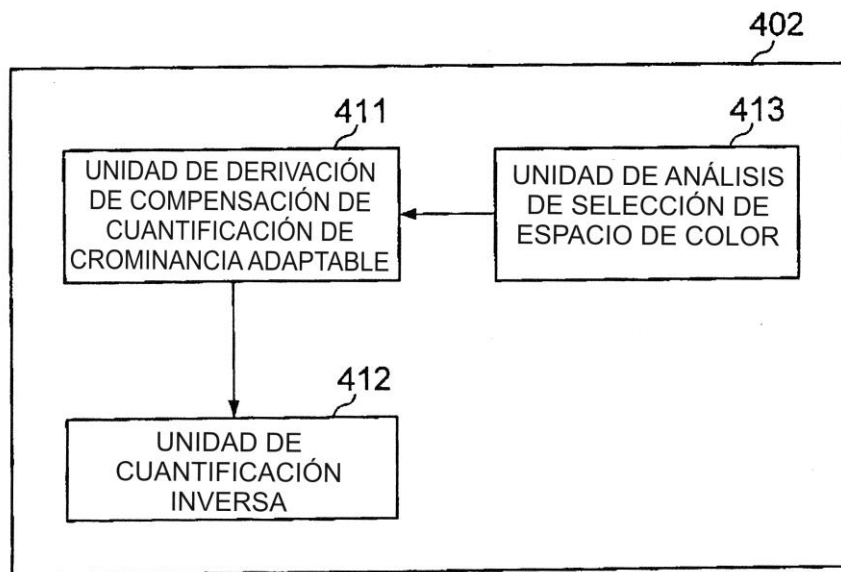


FIGURA 16

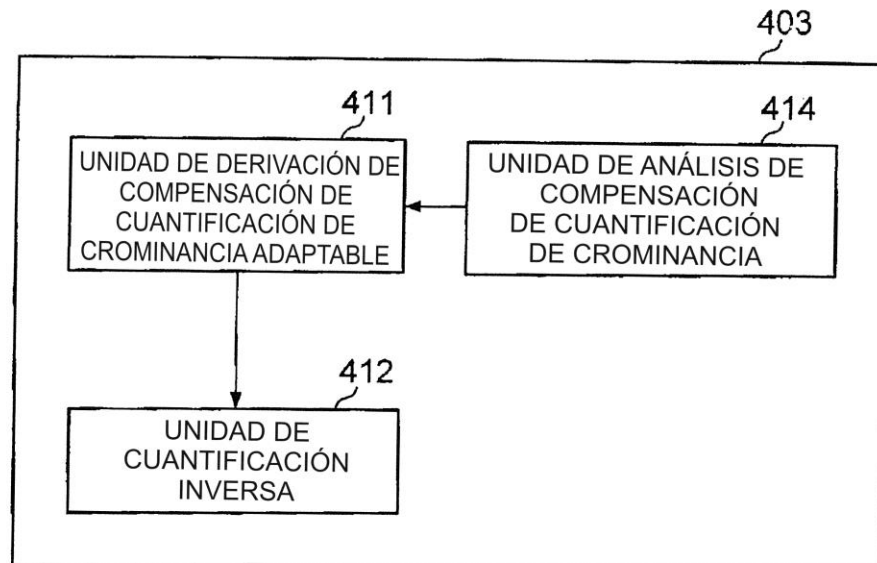


FIGURA 17

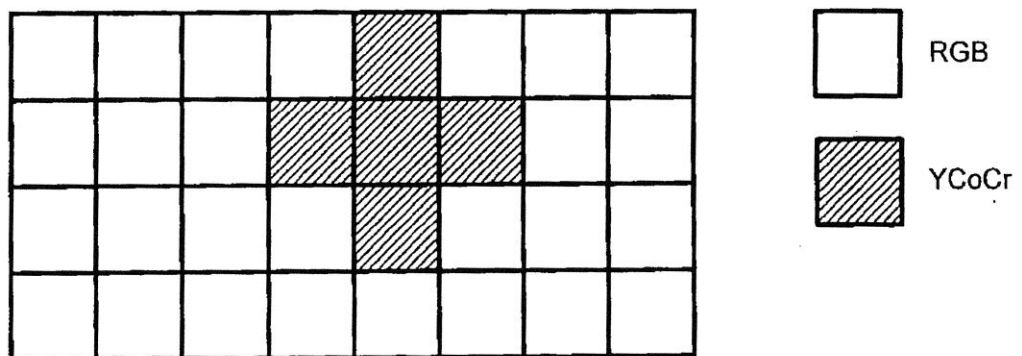


FIGURA 18

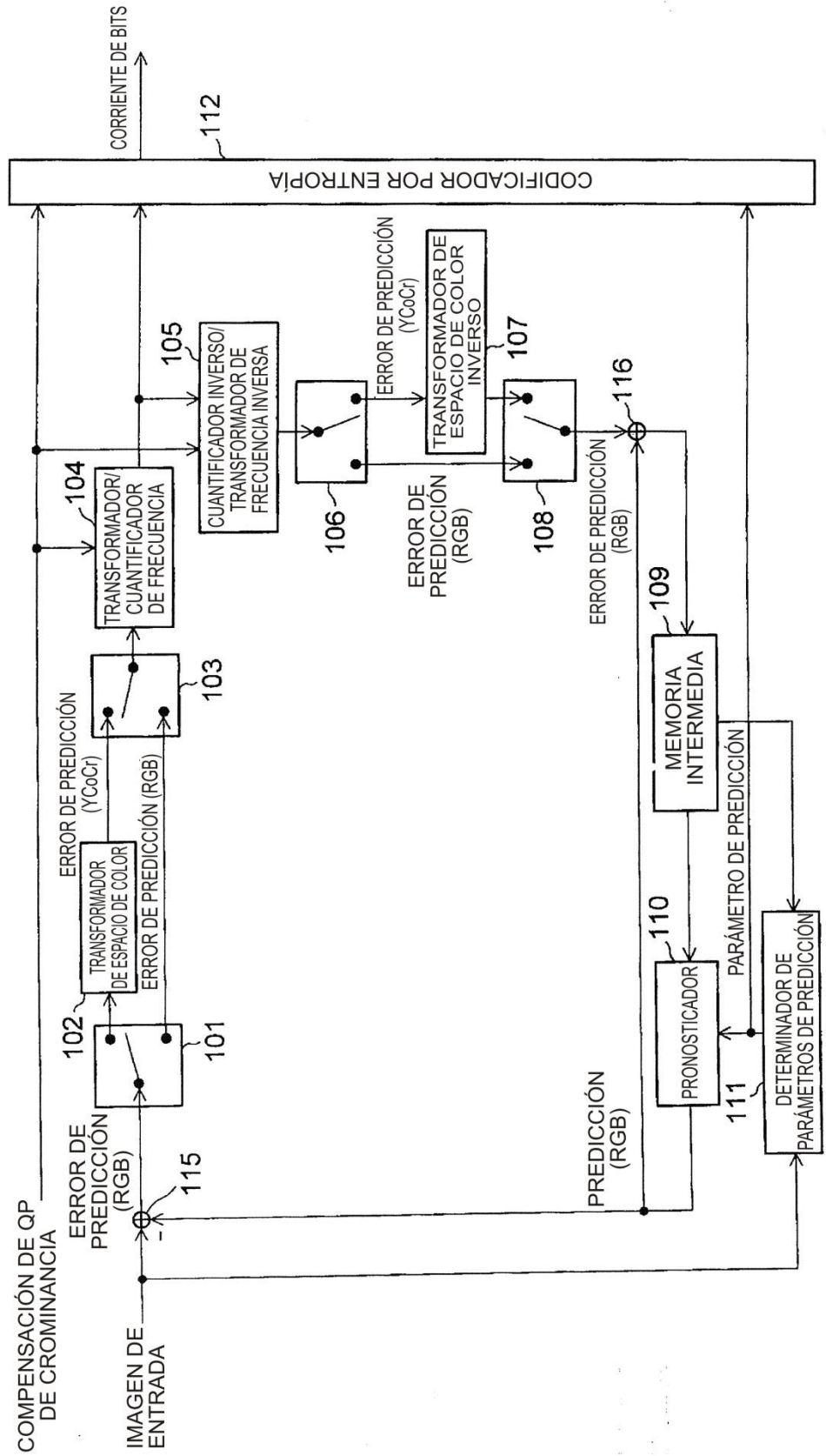


FIGURA 19

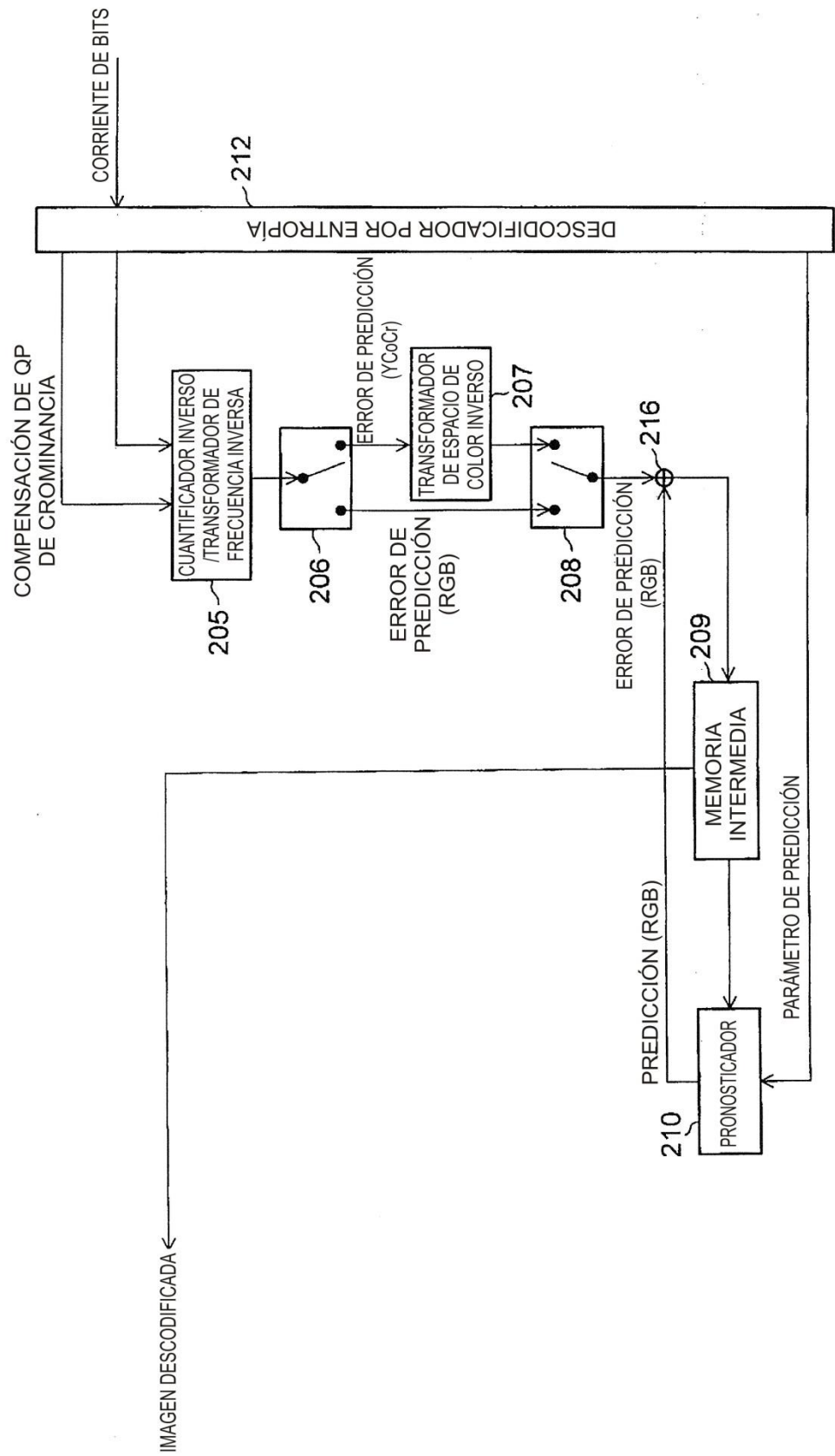


FIGURA 20

