



UNIVERSIDAD TECNOLÓGICA DE LA MIXTECA

DIVISIÓN DE ESTUDIOS DE POSGRADO

MAESTRÍA EN ELECTRÓNICA
Y COMPUTACIÓN

SEGMENTACIÓN DE IMÁGENES DIGITALES
EN COLOR MODIFICANDO
EL SISTEMA DE COLOR I1I2I3

TESIS

QUE PARA OBTENER EL GRADO
DE MAESTRO EN ELECTRÓNICA Y COMPUTACIÓN

PRESENTA:

MARÍA LUISA ANTONIETA
GUERRERO RAMÍREZ

ASESOR:

DR. JOSÉ JAVIER BÁEZ ROJAS

HUAJUAPAN DE LEÓN, OAX., FEBRERO DE 2001

Contenido

INTRODUCCIÓN	1
PLANTEAMIENTO DEL PROBLEMA	3
OBJETIVO	4
1 COLOR DIGITAL	5
1.1 Visión del color	5
1.2 Fundamentos de color	6
1.3 Color digital	8
1.4 Tabla de color	9
1.5 Modelos de color.	11
1.5.1 Modelo de color <i>RGB</i>	11
2 SEGMENTACIÓN	13
5.1 Técnicas de segmentación basadas en encontrar directamente las regiones.	14
5.1.1 Crecimiento por agregación de píxeles	14
5.1.2 División y fusión de regiones	15
5.2 Otras técnicas de segmentación	15
3 SISTEMA DE COLOR I1I2I3	16
3.1 Sistema de Color I1I2I3	17
3.2 Modificación del Sistema de Color I1I2I3	20
4 PROCEDIMIENTO DE SEGMENTACIÓN	23
4.1 Generación de las imágenes digital I^*	23
4.2 Elección de la imagen digital I^*	23
4.3 Obtención de una imagen digital binaria	24
4.4 Eliminación del ruido	24
4.4.1 Detección de puntos	24
4.4.2 Detección de líneas de dos puntos	25
4.5 Multiplicación de la imagen binaria con la imagen digital original	26

5	PSEUDOCÓDIGOS	27
5.1	Generación de imágenes digitales I^* (I_1 , I_2 , I_3 , I_2_GR , I_2_GB)	28
5.2	Obtención de la imagen digital binaria.	31
5.3	Multiplicación de la imagen digital binaria con la imagen digital original.	33
6	PRUEBAS	36
6.1	Componentes $I_1I_2I_3$	38
6.2	Fórmulas modificadas	41
6.3	Procedimiento de segmentación	43
6.3.1	Región de color verde	43
6.3.2	Región de color rojo.	46
6.3.3	Región de color café	49
6.3.4	Región de color amarilla	52
6.3.5	Región de color azul	55
6.3.6	Región de color negro	61
6.4	Segmentación de varias regiones..	64
7	RESULTADOS Y CONCLUSIONES	71
7.1	Resultados	71
7.1.1	Tabla de imágenes digitales I^* , umbrales y colores	71
7.2.1	Resultados	74
7.2	Conclusiones	84
A	MULTIPLICACIÓN DE IMÁGENES DIGITALES	86
B	FORMATO.H	88
C	MANUAL DE USO DEL PROCEDIMIENTO DE SEGMENTACIÓN DE IMÁGENES DIGITALES	91
	BIBLIOGRAFÍA	94

Lista de figuras y tablas

Tablas

1	Ejemplo de la heterogeneidad de los tonos de color de los píxeles del mapa de trabajo	4
1.1	Adición de los colores primarios	7
1.2	Cubo de valores RGB	12
3.1	Ajuste de los valores de píxel, de negativos a positivos por medio de un corrimiento	21
4.1	Máscara utilizada para detectar puntos aislados de un fondo constante	25
4.2	Máscaras utilizadas para detectar líneas de dos puntos	25
6.1	Imagen digital de trabajo (fragmento de la carta topográfica de Oaxaca)	37
6.2	Imagen resultado de aplicar la componente I1	38
6.3	Imagen resultado de aplicar la componente I2	39
6.4	Imagen resultado de aplicar la componente I3	40
6.5	Imagen resultado de aplicar la fórmula (15), de la componente I2 modificada (I2_GR)	41
6.6	Imagen resultado de aplicar la fórmula (16), de la componente I2 modificada (I2_BG)	42
6.7	Imagen binaria generada a partir de la imagen de la fórmula I2_BG	43
6.8	Imagen que muestra la segmentación de la región verde	44
6.9	Imagen original con la región verde de diferente color	45
6.10	Imagen binaria de la región de color rojo	46
6.11	Segmentación de la región de color rojo	47
6.12	Imagen original con la región roja de diferente color	48

6.13	Imagen binaria de la región de color café	49
6.14	Segmentación de la región de color café	50
6.15	Imagen original con la región de color café de un color distinto	51
6.16	Imagen binaria de la región de color amarillo	52
6.17	Segmentación de la región de color amarillo	53
6.18	Imagen original con la región amarilla de diferente color	54
6.19	Imagen binaria de la región azul correspondiente a los ríos	55
6.20	Segmentación de la región de color azul correspondiente a los ríos	56
6.21	Cambio de color de la región de color azul correspondiente a los ríos en la imagen original.	57
6.22	Imagen binaria de la región de color azul correspondiente al lago	58
6.23	Segmentación de la región de color azul correspondiente al lago	59
6.24	Cambio de color de la región de color azul correspondiente al lago sobre la imagen original	60
6.25	Imagen binaria de las regiones de color negro	61
6.26	Segmentación de las regiones de color negro	62
6.27	Cambio de color de la región de color negro sobre la imagen original	63
6.28	Figura 6.28: Imagen resultado de cambiar el color de los ríos a la imagen original.. . . .	64
6.29	Figura 6.29: Imagen resultado de cambiar a la imagen de la figura número 6.28 el color del lago.. . . .	65
6.30	Imagen resultado de cambiar a la imagen de la figura número 6.29 el color de la región amarilla.. . . .	66
6.31	Imagen resultado de cambiar a la imagen de la figura número 6.30 el color de la región roja.. . . .	67
6.32	Imagen resultado de cambiarle a la imagen de la figura número 6.31 el color de la región de color negro.. . . .	68
6.33	Imagen resultado de cambiar a color blanco la región café de la imagen original.. . . .	69

Tablas

1.1	Ejemplo de una tabla de color	10
-----	---	----

1.2	Ejemplo de una tabla de color en tonos de gris	11
6.1	Simbología	36
7.1	Tabla de imágenes digitales I*, umbrales y colores.	72
7.2	Tabla de color.	72

INTRODUCCIÓN

La presente tesis forma parte de un proyecto en conjunto planteado por el Dr. José Javier Báez Rojas, que pretende la creación de un sistema computacional que maneje, modifique y cree mapas, para que sirva de herramienta a los Institutos de Minería, Diseño e Hidrología de esta Universidad. Este sistema incluirá algoritmos del Procesamiento Digital de Imágenes, además de una adecuada interfaz gráfica y *drivers* para la utilización de *plotters* e impresoras.

El tema de esta tesis es la **Segmentación de Imágenes Digitales en Color**. La *segmentación* se encuentra dentro de la etapa del análisis [8] en el Procesamiento Digital de Imágenes, y se define como: “Subdividir una imagen en sus partes constituyentes u objetos“ [8].

A su vez, la *Segmentación de Imágenes en Color* permite subdividir una imagen digital en color de acuerdo a la similitud o diferencia existente entre las características de color que presenten los píxeles que la forman, es decir, en determinadas regiones de color.

Por medio de la *segmentación* es posible modificar y crear mapas, ya que permite extraer determinadas regiones y cambiarles el color, con lo cual se genera un mapa con información diferente al mapa original.

Las técnicas de segmentación de imágenes digitales más conocidas se refieren a imágenes en blanco y negro, y se han adaptado para segmentar imágenes digitales en color.

Al investigar sobre una técnica de segmentación en color apropiada al tipo de mapas que se pretende manejar, fue notorio descubrir que no existe mucha información sobre el tema, sobretodo en lo referente al Procesamiento Digital de Imágenes. En cambio, se encontraron técnicas utilizadas en el campo de la Inteligencia Artificial (por ejemplo, ver referencia [10]), área fuera de los límites de esta tesis.

Como producto de esta investigación se encontró la tesis: *Color Image Invariant Segmentation and Retrieval Ph.D. Thesis* (1996), de Theo Gevers, de la Universidad de Ámsterdam [7]. En esta tesis se introduce el concepto de características sobresalientes del color digital para obtener información del color.

De los sistema de color mencionados en dicha tesis, se eligió el **Sistema de Color I1I2I3**, expuesto por los Doctores Yuichi Ohta, Takeo

Kanade y Toshiyuki Sakai, en el artículo *Color Information for Region Segmentation*, publicado en *Computer Graphics and Image Processing* (1980).

El *Sistema de Color IIII2I3* original permite realizar una segmentación de imágenes digitales en color por medio de las componentes I1, I2 e I3. Pero solo dos de las componentes propuestas (I1 e I3), involucran a las tres bandas de color (rojo, verde y azul), que forman una imagen digital en color.

Esto significa que en la componente I2 no se involucran a todos los posibles colores formados al combinar las tres bandas, y esto limita el rango de colores que se pueden segmentar.

Es por ello se modificó el *Sistema de Color IIII2I3*, en la parte de la componente I2 para incluir a las tres bandas de color y permitir realizar una mejor segmentación. Esta modificación es el tema central de esta tesis.

El desarrollo de esta tesis se hizo bajo el sistema operativo Linux (versión 6.3) con ambiente gráfico Gnome, con el fin de contar con un ágil manejo y manipulación de las imágenes digitales, además de aprovechar las ventajas que presenta, como la estabilidad y robustez. Se utilizó el lenguaje de programación en C, dado que es el lenguaje que he manejado anteriormente en la manipulación de imágenes digitales.

Como mapa de trabajo para esta tesis se utilizó la carta topográfica de Oaxaca, clave E14-6, en escala 1: 250 000, publicado por el Instituto Nacional de Estadística, Geografía e Informática (INEGI). Dadas las proporciones de esta carta solo se tomó un fragmento que contiene información representativa de las regiones en color del mapa total. Al escanearse, a las imágenes digitales en color se les dio el formato PPM (*portable pixel map*), y el formato PGM (*portable gray map*) a las imágenes digitales en blanco y negro .

En la siguiente sección se expone el problema que dio pie a la realización de esta tesis.

PLANTEAMIENTO DEL PROBLEMA

Dado que la segmentación de imágenes digitales en color es una técnica importante para modificar y crear mapas, en un principio se probaron las técnicas comúnmente utilizadas, entre ellas, el uso de la desviación estándar.

Pero, al analizar el mapa de trabajo ya digitalizado, se descubrió que no era posible agrupar los píxeles de un color como una sola región porque mostraban una heterogeneidad grande. Esto significa que en una región que a simple vista parecía un determinado color, en realidad se trataba de la píxeles de diversos tonos de color, similares entre sí, pero no iguales. Además de que en algunas regiones de color se encontraban píxeles de color negro y blanco.

Para ejemplificarlo, se incluye un fragmento del mapa de trabajo donde se muestran dos regiones, una de color verde y otra café. Para distinguir la gran variedad de tonos de color se ha ampliado 10 veces su tamaño original (ver figura número 1).

Además, cabe resaltar que los píxeles de la frontera entre las dos regiones no muestran gran diferencia del color. En este caso, por ejemplo, sería difícil decidir de manera simple si un determinado píxel de la frontera pertenece a la región verde o a la región de color café.

Unido a lo anterior, no fue posible aplicar los métodos tradicionales de segmentación por el hecho de no conocer el rango de la variedad de tonos de color, ya sea para utilizar un tipo de umbral o comparativo al momento de segmentar. Para esto habría sido necesario hacer un censo o tomar una muestra muy grande de los píxeles de la región para conocerlo, lo cual implica una inversión extra de tiempo.

Por ello, era necesario encontrar una técnica que permitiera realizar la segmentación de imágenes en color de manera óptima, empleando el menor número de recursos posibles, entre ellos el tiempo empleado.

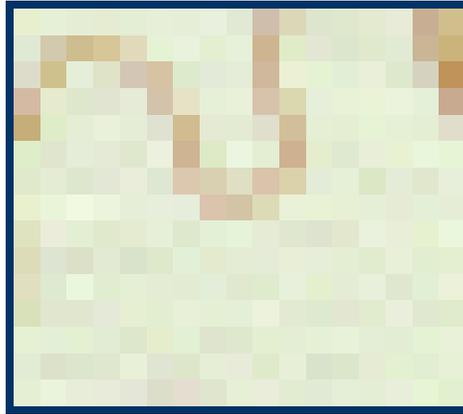


Figura 1: Ejemplo de la heterogeneidad de los tonos de color de los pixeles del mapa de trabajo.

OBJETIVO

Generar un procedimiento de segmentación de imágenes digitales en color que presenten un alto grado de heterogeneidad en los tonos de color de los pixeles de una determinada región.

La organización de esta tesis se divide en seis capítulos. El primer capítulo se refiere al Color Digital, exponiendo su concepto así como del modelo de color *RGB*. El segundo capítulo expone de manera breve el concepto de segmentación así como algunas de sus técnicas. El tercer capítulo presenta el *Sistema de Color III2I3* y la modificación realizada. En el capítulo cuarto se explica el procedimiento de segmentación utilizado, y el capítulo quinto su pseudocódigo. El capítulo sexto muestra las pruebas realizadas. Para finalizar, en el capítulo 7 se agregan los resultados obtenidos con cinco imágenes digitales diferentes y las conclusiones de esta tesis.

Capítulo 1

COLOR DIGITAL

Para entender el por qué el ojo humano puede ver determinados colores se incluye una breve explicación de su funcionamiento.

1.1 VISIÓN DEL COLOR

La retina es la parte del ojo en donde se forman las imágenes para poder verlas con nitidez. En ésta se encuentran los bastones y conos, conocidos como las moléculas de pigmento de los foto-receptores de los ojos, que convierten los rayos de luz en señales nerviosas [8].

Los conos de la retina son de tres tipos: uno es sensible a la región roja del espectro, esto es, a las longitudes de onda correspondientes a la región roja; otro, sensible a la región verde; y el tercero, sensible a la región azul-violeta. Cada tipo de cono es sensible a una amplia gama de longitudes de onda: los conos sensibles al rojo, lo son también al anaranjado, amarillo y verde; los elementos sensibles al verde responden a una gama desde el rojo al azul; y, los conos sensibles al azul reaccionan también al violeta y al verde.

La teoría de *Thomas Young – Hermann von Helmholtz* [8] establece que la luz es una perturbación ondulatoria, y que el color está relacionado con la longitud de onda de la luz.

La luz de casi cualquier longitud de onda aislada tiende a excitar, en alguna medida, a por lo menos dos conjuntos de conos, a causa de sus gamas de sensibilidad sobrepuestas. La luz proveniente de las regiones roja y verde del espectro, recibida como una sola impresión, se percibe como amarillo. Si los tres conjuntos de conos retinales se estimulan por igual, la sensación resultante es la de la luz blanca.

1.2 FUNDAMENTOS DE COLOR

En 1666 Isaac Newton descubrió que cuando un rayo de sol pasa a través de un prisma de cristal, el rayo de luz que emite no es blanco, sino que está formado por un espectro continuo de colores que van desde el violeta al rojo, y ninguno de ellos termina bruscamente, sino que se mezcla suavemente con el siguiente espectro. Este espectro de color se divide en seis regiones: violeta, azul, verde, amarillo, naranja y rojo [8].

Los colores que los seres humanos percibimos en un objeto están determinados por la naturaleza de la luz reflejada por el objeto. Un cuerpo que refleja luz relativamente equilibrada, en todas las longitudes de onda visibles, aparece blanco para el observador. Si un cuerpo tiene una mayor reflectancia en una determinada banda del espectro visible, muestra color.

Si la luz no tiene color, su único atributo es la intensidad o cantidad de luz, esto se conoce como luz acromática y es, por ejemplo, la que emite un televisor en blanco y negro. El término *niveles de grises* se refiere a una medida escalar de la intensidad que va del negro al blanco pasando por los grises.

La luz con color tiene un espectro electromagnético de luz visible que va de 400 a 700 nm (10^{-9} m). Las características de una fuente cromática de luz son tres magnitudes básicas: radiancia, luminancia y brillo.

La **radiancia** es la cantidad total de energía que emite una fuente luminosa, se mide en watios (W). La **luminancia** proporciona una medida de la cantidad de energía que percibe el observador, se mide en lúmenes (lm). El **brillo** es un descriptor subjetivo, y se refiere a la noción acromática de la intensidad y tiene que ver con las sensaciones del color.

El ojo humano, dada su anatomía, percibe los colores como una combinación de los tres colores primarios: rojo, verde y azul. En adelante, se utilizará la notación inglesa *RGB (Red, Green, Blue)*, al referirse a cada uno de los tres planos de color.

Los colores primarios se pueden sumar y al hacerlo, se producen los colores secundarios:

magenta = rojo + azul
cian = verde + azul
amarillo = rojo + verde

Si se mezclan los tres colores primarios o un secundario con su color primario opuesto en las intensidades correctas, se obtendrá luz blanca:

$$\mathbf{blanco = rojo + verde + azul}$$

Estas combinaciones son las propiedades aditivas del modelo *RGB* (ver figura núm. 1.1).

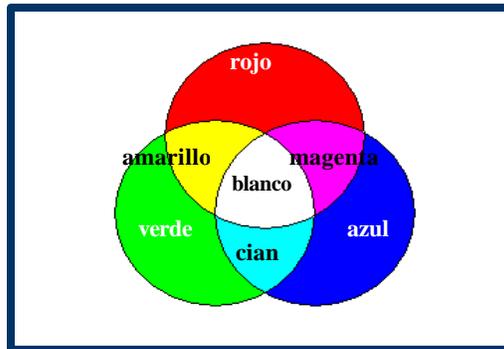


Figura 1.1: Adición de los colores primarios.

Para distinguir un color de otro se utilizan las características brillo, tono y saturación. El **brillo** se relaciona con la noción cromática de intensidad. El **tono** es un atributo asociado con la longitud de onda dominante en una mezcla de ondas luminosas, y representa el color dominante tal y como es percibido. La **saturación** se refiere a la pureza relativa o cantidad de luz blanca mezclada con otro tono. El tono y la saturación constituyen la cromaticidad.

Para formar un color se utilizan los valores triestímulo, que son cantidades de rojo, verde y azul, y se indican por *X*, *Y* y *Z*, respectivamente. Un color se especifica por sus coeficientes tricromáticos, definidos como sigue:

$$x = \frac{X}{X + Y + Z} \quad (1)$$

$$y = \frac{Y}{X + Y + Z} \quad (2)$$

y

$$z = \frac{Z}{X + Y + Z} \quad (3)$$

por lo tanto:

$$x + y + z = 1$$

(4)

1.3 COLOR DIGITAL

Una imagen digital consiste en un arreglo de dos dimensiones de valores muestreados tomados de la escena o impresión [8]. En las imágenes digitales a color cada uno de estos valores especifica la información del color de un punto espacial en la imagen. Para caracterizar la luz incidente en los detectores de la cámara o escáner de donde se obtiene la imagen digital, es necesario especificar la radiancia de las longitudes de onda, una especificación mínima que permita recrear la sensación de color de la imagen original al momento del despliegado de la misma [1].

Una imagen digital a color se representa por un arreglo de tríada, donde cada una es un número finito binario. Cada tríada representa el color dentro de un área finita de una escena original, centrada en un punto en una trama de puntos muestreados. Si existe cualquier variación o se pierde algún detalle dentro de esta área durante el muestreo, pueden observarse algunos bordes dentados o borrosos, o incluso, la pérdida de algún detalle fino del original.

Como cada tríada se representa como un número finito de dígitos o bits, solo un número finito de distintos colores puede ser representados. Si no existiese un color entre los colores representables, dentro de la gama del espacio del color, igual al original, se le da el color más cercano representable.

Un color se especifica usando los valores triestímulos, ya que permiten, aproximadamente, representar un color. Esto es consecuencia del hecho de que la percepción humana del color es mediante la respuesta de tres diferentes tipos de fotorreceptores en la retina, es decir, los conos. Por lo general, los valores triestímulos son enteros de 0 a 255 (para cada bandas de color *RGB*).

Al conjunto de colores descritos por las tríada de los valores de una imagen, producto de un dispositivo de captura específico, se le conoce como el *espacio nativo* de tal dispositivo. Para ser posible el procesamiento y manejo de imágenes a color capturadas por diferentes dispositivos, es necesario convertir del espacio nativo de cada dispositivo al espacio de color de referencia del dispositivo donde se esté trabajando, así como también en el caso de impresoras o *plotters*.

1.4 TABLA DE COLOR

Una forma de igualar el espacio nativo del dispositivo con el espacio de color de donde se trabaja, es generar una tabla que provea la correcta tríada de salida para cada tríada de entrada [1].

Para especificar estas transformaciones entre espacios es necesario estimar los parámetros de las entradas de la tabla. Con los dispositivos de entrada, esto puede hacerse capturando una imagen de una prueba meta, que contenga zonas de colores cuyas coordenadas *XYZ* del espacio de color deban ser medidas. Con el dispositivo de salida, se despliega o imprime una prueba meta para producir una copia prueba, y se comparan los colores de entrada y de salida. Crear una tabla de color requiere de un gran número de mediciones, al igual que calibrar una impresora.

Al crear una tabla de color se considera cada valor de la tríada que formara un color determinado, es decir, el valor correspondiente a la banda de color rojo, el valor de la banda de color verde y el valor de la banda azul. Dado que existen 256 valores por cada una de las bandas de color, es posible crear más de 16 millones de colores. La tabla 1.1 muestra un ejemplo de una tabla de color, ordenada de menor a mayor tomando como base la banda de color rojo.

En el caso de colores en tonos de grises, el valor de cada una de las bandas de color es el mismo, tomando valores de 0 a 255, siendo los valores menores los tonos de gris más oscuro y los valores más grandes, los tonos de gris claro. Los valores de 0 forman el color negro, y los valores de 255 el color blanco. La tabla 1.2 representa una tabla de color en tonos de gris.

ROJO	VERDE	AZUL	COLOR
0	0	0	Negro
0	0	128	Azul marino
0	0	255	Azul
0	255	0	Verde
0	255	255	Cian
46	139	87	Verde mar
64	224	208	Turquesa
70	123	180	Azul acero
107	142	35	Oliva
127	255	220	Aguamarina
135	206	235	Azul cielo
138	43	226	Azul violeta
160	32	240	Púrpura
165	42	42	Café
238	130	238	Violeta
245	245	220	Beige
250	128	114	Salmón
255	0	0	Rojo
255	0	255	Magenta
255	114	86	Coral
255	165	0	Naranja
255	192	203	Rosa
255	215	0	Oro
255	255	0	Amarillo
255	255	255	Blanco

Tabla 1.1: Ejemplo de una tabla de color.

ROJO	VERDE	AZUL	COLOR
0	0	0	Negro
1	1	1	
58	58	58	
75	75	75	
127	127	127	
164	164	164	
205	205	205	
226	226	226	
233	233	233	
255	255	255	Blanco

Tabla 1.2: Ejemplo de una tabla de color en tonos de gris.

1.5 MODELOS DE COLOR

El propósito de un modelo de color es facilitar la especificación de colores de algún formato estándar. Un modelo de color es la especificación de un sistema de coordenadas tridimensionales y de un subespacio dentro de ese sistema donde cada color se representa por un punto único [8].

1.5.1 MODELO DE COLOR *RGB*

En el modelo *RGB* cada color aparece en sus componentes primarias espectrales rojo, verde y azul [8]. Este modelo se basa en un sistema de coordenadas cartesiano. El subespacio de color de interés es un cubo en el que los valores *RGB* están en tres vértices; el cian, magenta y amarillo en otros tres vértices; el negro en el origen y el blanco en el vértice opuesto al origen.

En esta ejemplificación, la escala de grises se extiende del negro al blanco a lo largo de una diagonal del cubo, y los colores son puntos del cubo o en su interior definidos por vectores que se extienden desde el origen.

Se parte de la suposición de que todos los valores de color han sido normalizados, tal que el cubo representa la unidad, lo que significa que todos los valores *RGB* están en el rango [0,1].

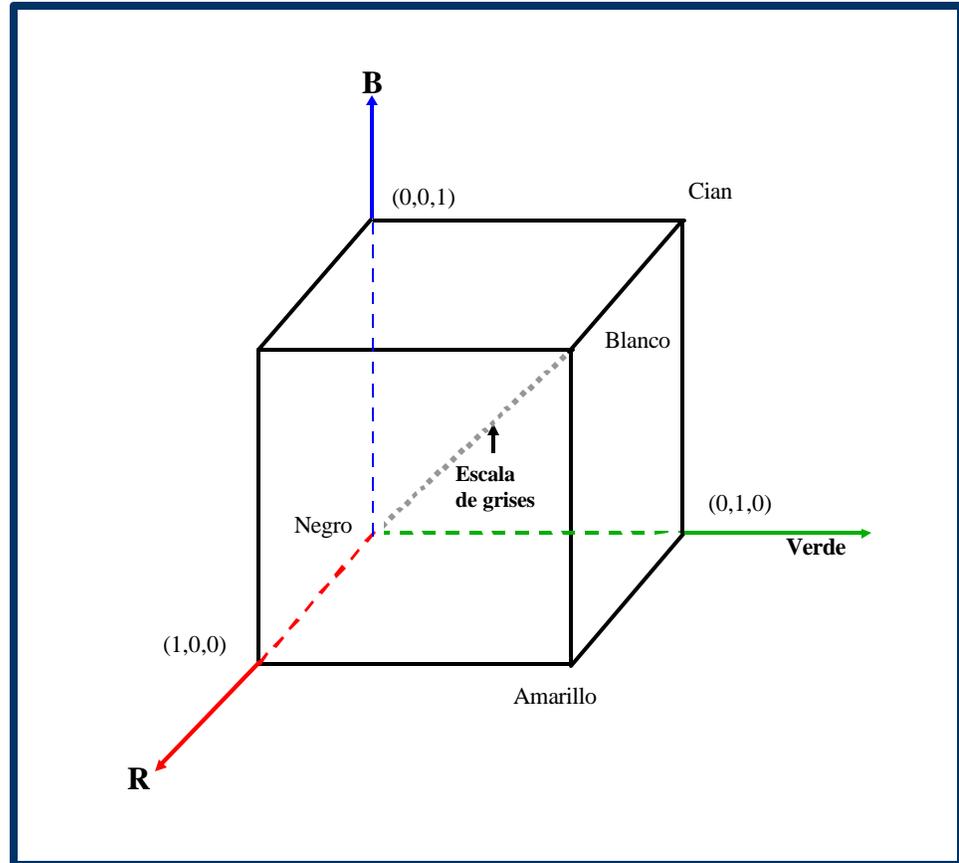


Figura 1.2: Cubo de valores RGB.

Las imágenes en este modelo están formadas en tres planos de imágenes independientes, uno por cada uno de los colores primarios. Cuando las imágenes son introducidas en un monitor *RGB*, las tres imágenes se combinan en la pantalla fosforescente para producir una imagen de color compuesta.

Capítulo 2

SEGMENTACIÓN

Dentro del Procesamiento Digital de Imágenes existe una parte conocida como análisis de la imagen [8]. La antecede el preprocesamiento de la imagen digital.

El primer paso del análisis de imágenes digitales por lo general consiste en segmentar la imagen. La *segmentación* es “subdividir una imagen en sus partes constituyentes u objetos”[8]. El grado de la subdivisión depende del tipo de problema que se pretende resolver.

En el caso de imágenes digitales en blanco y negro los algoritmos de segmentación se dividen de acuerdo a dos propiedades básicas de los valores del nivel de gris, que son discontinuidad y similaridad.

La *discontinuidad* divide una imagen en base a los cambios bruscos de nivel de gris. Esta propiedad a su vez puede dividirse en:

- Detección de puntos aislados.
- Detección de líneas y bordes de una imagen.

Las áreas de interés son la detección de puntos aislados, la detección de líneas y bordes de una imagen digital .

La segunda propiedad es la *similaridad* que se basa en dividir una imagen de acuerdo a las similitudes de nivel de gris. Los métodos de segmentación de esta categoría se basan en:

- Umbralización.
- Crecimiento de región.
- División y fusión de regiones.

De acuerdo al tipo de problema que se plantea en esta tesis, el tipo de segmentación de interés es la de *similaridad*, de donde se deriva la segmentación orientada a encontrar directamente las regiones, la cual se menciona a continuación.

5.1 TÉCNICAS DE SEGMENTACIÓN BASADAS EN ENCONTRAR DIRECTAMENTE LAS REGIONES

Una *región* se define como “un conjunto de píxeles conectados, tal que es un conjunto en donde todos los píxeles son adyacentes”[4].

La conectividad se refiere a que entre dos píxeles en un conjunto, existe un camino conectado dentro del mismo, tal que este *camino conectado* es un camino que siempre se mueve dentro de los píxeles vecinos [8].

La formulación básica [8] de las técnicas de segmentación que se basan encontrar directamente las regiones es la siguiente:

Sea R la representación de la región completa de una imagen. Se considera que la segmentación es el proceso que divide a R en n subregiones, R_1, R_2, \dots, R_n , tal que:

- a) $U_{i=1}^n R_i = R$. La segmentación debe ser completa, esto significa que cada uno de los píxeles debe estar en una región.
- b) R_i es una región conexas, $i=1,2,\dots, n$.
- c) $R_i \cap R_j = \Phi$ para todo i y j , $i \neq j$. Las regiones deben ser disjuntas.
- d) $P(R_i) = VERDADERO$ para $i = 1,2, \dots, n$.
- e) $P(R_i \cup R_j) = FALSO$, para $i \neq j$. R_i y R_j son diferentes en el sentido del predicado P .

donde $P(R_i)$ es un predicado lógico sobre los punto del conjunto R_i y Φ es el conjunto vacío.

Entre las técnicas que integran este tipo de segmentación se encuentran *el crecimiento por agregación de píxeles*, y *la división y fusión de regiones*.

5.1.1 CRECIMIENTO POR AGREGACIÓN DE PÍXELES

El *crecimiento de regiones* es un procedimiento que agrupa píxeles u subregiones dentro de regiones de mayor tamaño [8], por medio de la agregación de píxeles.

El procedimiento inicia con un conjunto de puntos “generadores” a partir de los cuales van creciendo las regiones, ya que a cada uno de estos puntos se les agregan los píxeles próximos con propiedades similares. Estas propiedades pueden ser el nivel de gris, la textura o el color.

Los problemas a presentarse son la selección de los generadores iniciales que representen correctamente a las regiones de interés y la selección de las propiedades adecuadas para la agregación de puntos de las diversas regiones durante el proceso de crecimiento. Además del adecuado planteamiento de una regla de parada del crecimiento.

5.1.2 DIVISION Y FUSIÓN DE REGIONES

Una variante del procedimiento de crecimiento por agregación de pixeles es la *división y fusión de regiones* [8].

Esta consiste en subdividir la imagen digital inicialmente en un conjunto de regiones arbitrarias disjuntas y después fusionar y/o dividir las regiones, siempre y cuando satisfagan las condiciones que plantea la formulación básica.

5.2 OTRAS TÉCNICAS DE SEGMENTACIÓN

Dentro de los textos que tratan el Procesamiento Digital de Imágenes de manera general, la segmentación de imágenes digitales en color consiste en la adaptación de las técnicas utilizadas para imágenes digitales en blanco y negro.

Lo que proponen es aplicar las técnicas a cada una de las bandas de color que forman una imagen digital en color (*rojo, verde, azul*).

Otra de las técnicas es utilizar métodos estadísticos [3], como la *desviación estándar*. Esta consiste en muestrear un determinado número de pixeles, y obtener la desviación estándar, la cual proporciona información sobre el rango de valores del nivel de gris (o del color rojo, verde y azul) que integran la región.

Este método, de obtener resultado óptimos, puede servir de base para aplicar el de *crecimiento de regiones por agregación de pixeles*.

Cabe mencionar que el tipo de imagen digital en color utilizada en esta tesis presenta un alto grado de heterogeneidad en los tonos de color de los pixeles en una determinada región, por lo que las técnicas antes mencionadas no dieron resultados óptimos. Así que se decidió investigar sobre otros métodos de segmentación. En el siguiente capítulo se trata el método de segmentación finalmente utilizado.

Capítulo 3

SISTEMA DE COLOR I1I2I3

La parte central de esta tesis es la *Segmentación de Imágenes en Color*. Este tipo de segmentación permite dividir una imagen digital en color de acuerdo a la similitud o diferencia que exista entre las características de color que presenten los píxeles.

Las dos grandes áreas interesadas en la segmentación de imágenes digitales en color son el Procesamiento Digital de Imágenes y la Inteligencia Artificial.

En el Procesamiento Digital de Imágenes, área base de esta tesis, los algoritmos para segmentar imágenes en color más utilizados son adaptaciones de algoritmos para imágenes en blanco y negro.

Al investigar sobre algoritmos propios para imágenes digitales en color se encontró la tesis: *Color Image Invariant Segmentation and*

Retrieval Ph.D. Thesis (1996), de Theo Gevers, de la Universidad de Ámsterdam [7].

La tesis de Theo Gevers introduce el concepto de características sobresalientes del color digital para obtener información del color. Las características se seleccionan de acuerdo a las que proporcionen los mejores resultados de segmentación. Para ello, implementa dos algoritmos de segmentación: *clustering*, y crecimiento de regiones (*region growing*).

Esta tesis menciona varios sistemas de color con los cuales se experimentó, y de ellos se tomó el **Sistema de Color I1I2I3** para resolver el problema que se tenía planteado para la realización de esta tesis.

3.1 SISTEMA DE COLOR $I_1I_2I_3$

El **Sistema de Color $I_1I_2I_3$** [7] fue expuesto por los Doctores Yuichi Ohta, Takeo Kanade y Toshiyuki Sakai, en el artículo *Color Information for Region Segmentation*, publicado en *Computer Graphics and Image Processing* (1980).

Este sistema de el color se ha empleado, entre otras aplicaciones, en la segmentación en imágenes digitales en color para darle realismo a objetos en tercera dimensión así como para su reconstrucción; en el reconocimiento visual de objetos; y en la segmentación y medición de áreas en imágenes digitales médicas. Para más detalles, ver en la Bibliografía de este documento la sección dedicada a ligas en internet.

El **Sistema de Color $I_1I_2I_3$** se describe por medio de las coordenadas I_1 , I_2 , e I_3 [7], las cuales se explican a continuación.

El punto de inicio del cálculo de las características del color de este sistema es la aplicación de la transformación de Karhunen-Loève. Esta transformación se calcula de la matriz de covarianza.

De la matriz de covarianza se determinan los eigenvectores y los eigenvalores. La característica del color correspondiente al eigenvalor más grande es el que tiene la varianza más grande. Las características del color que presenten varianzas grandes son más útiles que las características del color con varianzas pequeñas, ya que expresan mejor la

separación de diferentes *cluster* (regiones) bajo la hipótesis de que el ruido es igual.

Específicamente: Sea I la imagen entera. Se calcula la matriz de covarianza C de las distribuciones de R , G y B , en I . Se calculan los eigenvectores y eigenvalores $\lambda_1, \lambda_2, \lambda_3$ de C , y $\lambda_1 \geq \lambda_2 \geq \lambda_3$. Sea v_i , para $i=1,2$ y 3 , los eigenvectores de C correspondientes a λ_i .

La característica de color no correlacionada se calcula como sigue:

$$X_i = v_i^T a \quad (\|v_i\|=1, i=1,2,3)$$

(5)

y la siguiente matriz es un pixel

$$a = \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

(6)

La mejor característica en este sentido, es la que tiene la varianza más grande X_1 .

Las tres características del color fueron descubiertas experimentando y analizando los resultados de ocho escenas a color de Otha et al. (1980).

Estas escenas a color fueron digitalizadas con una resolución espacial de 256×256 y una resolución de intensidad de 6-bits por cada R , G , y B . Otha calculó los eigenvectores de C de cada una de las ocho escenas a color, y descubrió que:

$$\vartheta_1 \text{ era aproximadamente: } \begin{bmatrix} \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \end{bmatrix} \quad (7)$$

$$\vartheta_2 \text{ era aproximadamente: } \begin{bmatrix} \frac{1}{2} \\ 0 \\ -\frac{1}{2} \end{bmatrix} \quad (8)$$

$$\text{ó: } \begin{bmatrix} -\frac{1}{2} \\ 0 \\ \frac{1}{2} \end{bmatrix} \quad (9)$$

y v_3 era aproximadamente:

$$\begin{bmatrix} 1 \\ -\frac{1}{4} \\ \frac{1}{2} \\ -\frac{1}{4} \end{bmatrix}$$

(10)

De donde surgieron las tres características ortogonales del color:

$$I1 = \frac{(R + G + B)}{3}$$

(11)

$$I2 = \frac{(R - B)}{2}$$

(12)

$$I3 = \frac{(2G - R - B)}{4} \quad (13)$$

las cuales son componentes importantes en orden de representar información del color, donde además, $I1$ corresponde a la intensidad.

Por lo tanto, el **Sistema de Color I1I2I3** se define como sigue:

$$\begin{bmatrix} I1 \\ I2 \\ I3 \end{bmatrix} = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{2} & 0 & -\frac{1}{2} \\ -\frac{1}{4} & \frac{1}{2} & -\frac{1}{4} \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (14)$$

3.2 MODIFICACIÓN DEL SISTEMA DE COLOR I1I2I3

Durante el desarrollo de esta tesis, al realizar la segmentación en color se implementó el Sistema de Color I1I2I3, aplicando los componentes $I1$, $I2$, e $I3$, es decir, las fórmulas (11), (12) y (13) respectivamente.

Después de hacer varias pruebas aplicando dichas componentes en la imagen digital en color del mapa de trabajo, se concluyó que la componente $I2$ es la que mejor resultados presentaba, ya que mostraba gran cantidad de información útil para segmentar. En el capítulo siguiente se muestran de manera gráfica los resultados obtenidos.

La información obtenida a partir de la componente $I2$ se refiere a los tonos de los colores rojo y azul, es decir, de los tonos de colores más cercanos al rojo y al azul. Los tonos del color verde no aparecen como tales. Esto limita el rango de colores que se pueden segmentar con esta componente.

Al revisar la componente $I2$, fórmula (12), es notorio observar que solo involucra a las bandas de color rojo (R) y azul (B), sin incluir a la banda verde (G). Lo que implica que muestra información incompleta de una imagen a color RGB .

Con el fin de obtener información más completa de una imagen a color, se agregó la banda de color verde (G), respetando la formulación matemática original. Lo que se hizo fue crear dos fórmulas más:

$$I2_GR = \frac{(G - R)}{2}$$

(15)

y

$$I2_BG = \frac{(B - G)}{2}$$

(16)

En estas fórmulas también se puede invertir el orden de las bandas de color involucradas, así por ejemplo, en la fórmula (15) se puede restar

(R-G). Para la realización de esta tesis se decidió solo utilizar este par de fórmulas y no las cuatro que pudieran utilizarse si se invirtieran las bandas de color. La razón es porque los resultados son similares, la diferencia consiste en que las imágenes digitales generadas por un tipo de combinación presentan en tonos claros ciertas regiones, y si se utilizara la combinación inversa los presenta en tonos oscuros.

Finalmente, las fórmulas que se utilizaron en esta tesis son: (11), (12), (15) y (16), referentes a las componentes $I1$, $I2$, y a las dos fórmulas nuevas respectivamente.

La aplicación de las fórmulas (12), (15) y (16) en imágenes digitales en color permite realizar la segmentación de colores diferentes a tonos de gris.

Para segmentar líneas ó áreas de color negro ó en tonos de gris, se utiliza la fórmula (11), de la componente $I1$, obteniendo mejores resultados que si se usaran las fórmulas (12), (15) y (16).

Al aplicar cada una de las fórmulas anteriores se realiza una resta, por lo que es muy probable obtener como resultado valores negativos.

Las imágenes digitales trabajan con valores enteros positivos, y el valor rango de valores que puede presentar un pixel en cualquier tríada RGB es de 0 a 255.

Así que, para evitar valores negativos, a los resultados obtenidos al aplicar las fórmulas se les aplicó un corrimiento, de tal manera que si el valor mínimo de un pixel era de -127 , ahora, el nuevo valor del pixel sería 0.

De igual manera, si el valor máximo original era de $+127$, ahora sería de 255. Para los valores de gris cuya diferencia es cero, se mapean al valor 127, lo que implica que todas las variaciones en tonos de gris siempre serán el mismo valor. Los puntos intermedios se ajustan de acuerdo al corrimiento propuesto.

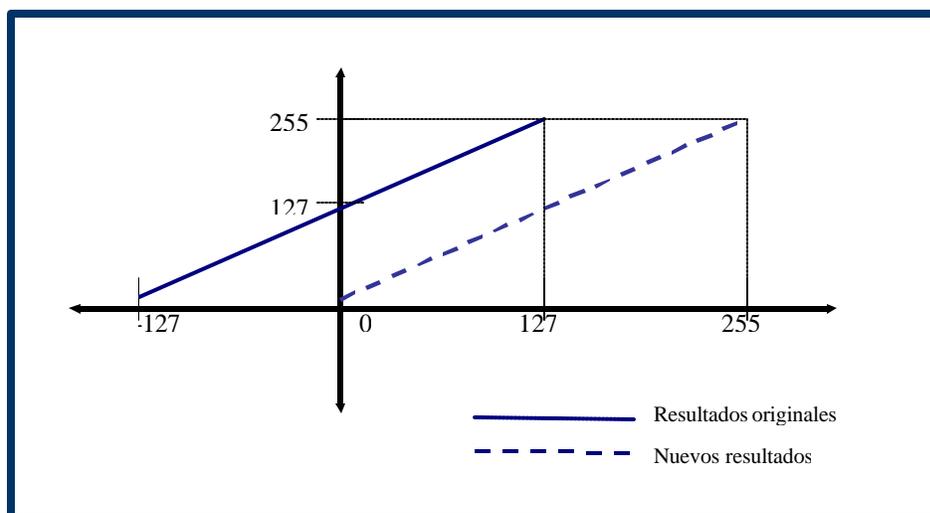


Figura 3.1: Ajuste de los valores de pixel de negativos a positivos por medio de un corrimiento.

El haber generado estas dos últimas fórmulas ha permitido la segmentación de un mayor número de colores en las imágenes digitales, ya que el sistema original al aplicarlo en el tipo de imagen usada en esta tesis, no incluye el color amarillo, y en algunas ocasiones lo presenta como si fuera de color rojo.

Lo que significa que al utilizar en conjunto las fórmulas originales del **Sistema de Color I1I2I3** y las fórmulas modificadas, es mayor el número de colores posibles a segmentar, tentativamente, todos los del modelo de color *RGB*.

Además, resuelve el problema de segmentar de manera óptima imágenes digitales en color con un alto grado de heterogeneidad en los tonos de color de los píxeles de una determinada región.

En el capítulo cuarto se explica de manera detallada el procedimiento de segmentación de imágenes digitales en color que propone esta tesis, y en el capítulo sexto las pruebas que se obtuvieron en su aplicación.

Capítulo 4

PROCEDIMIENTO DE SEGMENTACIÓN

La obtención de las imágenes digitales a partir de la aplicación del **Sistema de Color I1I2I3** y de las fórmulas derivadas de las componente I2, representa la parte inicial del procedimiento de segmentación. Las imágenes digitales utilizadas son las generadas a partir de las fórmulas (11), (12), (15) y (16). Para distinguir estas imágenes digitales se les denotará a partir de ahora como I*.

El procedimiento empleado para segmentar es el siguiente.

4.1 GENERACIÓN DE LAS IMÁGENES DIGITALES I*

De las fórmulas de las componentes I1, I2 y I3 y de las fórmulas modificadas de I2, se generan las imágenes digitales I* en tonos de gris con formato PGM. Se emplean las fórmulas (11), (12), (15) y (16).

4.2 ELECCIÓN DE LA IMAGEN DIGITAL I*

De acuerdo a la región que se interese segmentar, y de las regiones que resalten en las imágenes digitales I*, se elige la más conveniente. De las bandas de color involucradas en la fórmula que se aplique dependen los resultados mostrados en la imagen digital I*.

Por lo general, las regiones que son más fáciles de segmentar son aquellas que se diferencian fácilmente del resto de las regiones de la imagen digital. Por ejemplo, las líneas en tonos claros resaltan en un conjunto de regiones de tonos oscuros, y viceversa. Estas serían las imágenes digitales adecuadas para segmentar.

4.3 OBTENCIÓN DE UNA IMAGEN DIGITAL BINARIA

El siguiente paso en la segmentación es generar una imagen digital binaria a partir de la imagen digital I* elegida.

A la imagen digital I* se le aplica un umbral global simple, de tal forma que la nueva imagen digital muestre en color blanco la región de interés, es decir, la región segmentada, y en color negro el resto de la imagen.

El valor del umbral se elige con el auxilio del histograma de la imagen digital I^* , evitando ensayos innecesarios al momento de elegir un umbral.

En una primera instancia, la elección de la imagen digital binaria es de manera arbitraria, e involucra pruebas de ensayo y error hasta encontrar el umbral adecuado, es decir, una imagen digital con la mayor información de la región de interés y el menor ruido.

Una vez encontrado el umbral óptimo, este valor se puede utilizar como un estándar para ser utilizado en la segmentación de regiones del mismo color en otras imágenes digitales, permitiendo un rango de valores de la tríada que lo forman.

4.4 ELIMINACIÓN DE RUIDO

Muchas veces las imágenes binarias generadas pueden contener *ruido*. En este caso, el ruido significa los puntos blancos en donde no debería de haber. Esto se distingue a simple vista.

La eliminación del ruido se hace en base a la detección de puntos aislados y de dos puntos adyacentes de color blanco [8].

4.4.1 DETECCIÓN DE PUNTOS

La detección de puntos aislados de una imagen digital es directa. Se utiliza una máscara (ver figura 4.1), en donde con un valor de 1 se centra el pixel que significa ruido [8], es decir, el pixel blanco, y el resto de la máscara con valores de cero que representan los pixeles de color negro.

0	0	0
0	1	0
0	0	0

Figura 4.1: Máscara utilizada para detectar puntos aislados de un fondo constante.

4.4.2 DETECCIÓN DE LÍNEAS DE DOS PUNTOS

Tomando como base la máscara utilizada en la detección de un punto, se modificó para poder detectar dos puntos. Esto es, se buscaban dos puntos adyacentes de color blanco aislados del resto de la imagen digital .

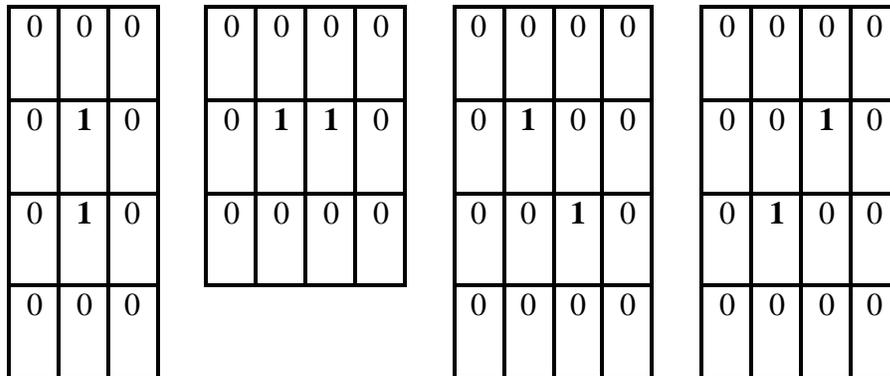


Figura 4.2 Máscaras utilizadas para detectar líneas de dos puntos.

Para ello se aplican máscaras de detección de dos puntos. Las máscaras se muestran en la figura 4.2. Cada una de estas máscaras recorre toda la imagen digital binaria.

Se eligió de manera arbitraria que las líneas de ruido fueran de solo dos puntos por considerar que si las líneas contenían un mayor número de puntos podría afectar el resultado final y borrar, por considerarse como ruido, píxeles correspondientes a alguna región de interés.

4.5 MULTIPLICACIÓN DE LA IMAGEN BINARIA CON LA IMAGEN DIGITAL ORIGINAL

Al multiplicar la imagen digital binaria por la imagen digital original es posible generar una imagen digital con la región segmentada, o bien, modificar el color de la región segmentada en la imagen digital original.

El procedimiento de la multiplicación de imágenes digitales se explica en el Apéndice A al final de este documento.

La segmentación de la región de interés se obtiene finalmente al multiplicar la imagen digital binaria ya preprocesada, es decir, sin ruido, por la imagen digital original.

El resultado es una imagen digital en donde se muestra, en su color original, a la región segmentada sobre un fondo negro. Esta es una imagen digital en color con formato PPM.

También al multiplicar la imagen digital binaria por la imagen digital original es posible cambiar el color de la región segmentada por uno nuevo en la imagen digital original.

El color nuevo se elige de una tabla de color prediseñada, o bien, de una tríada de color con valores nuevos.

El resultado es la imagen original con la región de interés de un color distinto.

Al modificar el color de la región de interés sobre la imagen original puede permitir homogeneizar el color, ya sea cambiando el color original por uno diferente, o por uno similar, teniendo la ventaja de que la imagen digital resultante tendrá toda la región de interés con todos los píxeles de igual valor. Es decir, ya no será una región con valores de píxel heterogéneos.

Capítulo 5

PSEUCÓDIGOS

La implementación del procedimiento expuesto en el capítulo 4 se hizo sobre el sistema operativo Linux (versión 6.3) con ambiente gráfico Gnome, y se utilizó el lenguaje de programación C.

De manera práctica, el procedimiento de segmentación se dividió en tres partes creando igualmente tres programas correspondientes.

Las partes son las siguientes:

- Generación de imágenes digitales I* (I1, I2, I3, I2_GR, I2_BG).
- Obtención de la imagen digital binaria .
- Multiplicación de la imagen digital binaria con la imagen digital original.

Cada uno de estos programas utiliza el archivo *formato.h* (ver Apéndice B), el cual identifica si la imagen digital de trabajo tiene formato PGM o PPM, además de extraer datos necesarios como el número de columnas y de renglones, y el valor máximo de pixel.

A continuación de incluyen los psudoalgoritmos de cada uno de estos programas.

5.1 GENERACIÓN DE IMÁGENES DIGITALES I* (I1, I2, I3, I2_GR, I2_BG)

Este pseudocódigo permite la generación de imágenes digitales I1, I2, I3, I2BG, o I2GR , una a la vez. Utiliza las fórmulas vistas en el capítulo 3 de este documento. Pide de entrada una imagen digital en color con formato PPM.

```
FILE *fnpuno
int      coluno, renuno, sizeone, maxuno, i, nsize, c1
float   R,G,B,i1,i2,i3,gg,menos,i2bg,i2gr
char    i1name[200]
u_char *uno,*datos
char    arg[200]
```

INICIO

```
imprimir "Uso: <nombre_archivo_entrada> <nombre de archivo_salida>"
lee argv[1]
lee argv[2]
```

```
datos = lee_ppm(argv[1],&coluno, &renuno,&maxuno)
sizeone = coluno*renuno*3
nsize = coluno*renuno
```

```
si ( (uno = (u_char *) calloc (nsize, sizeof(u_char ))) == NULL)
    imprimir "ERROR!. NO ES POSIBLE RESERVAR MEMORIA"
    salir
```

```
imprimir "OPCIONES:  1: I1  2: I2  3: I3  4:I2BG  5:I2GR "
lee c1
```

```
desde i=0 hasta i<renuno*coluno i++
```

```
    si c1= 1
```

```
        entonces
```

```
            R = (float)datos[3*i]
```

```
            G = (float)datos[3*i+1]
```

```
            B = (float)datos[3*i+2]
```

```
            i1 = (R+G+B)/3
```

uno[i] = (u_char) i1

fin si

si no c1= 2

entonces

R = (float)datos[3*i]

B = (float)datos[3*i+2]

i2 = (R-B)/2

i2= i2+ 127

uno[i] = (u_char)i2

fin si no

si no c1=3

entonces

R = (float)datos[3*i]

G = (float)datos[3*i+1]

B = (float)datos[3*i+2]

gg = 2*G

menos = gg-R-B

i3 = menos/4

i3 = i3 + 127

uno[i] = (u_char)i3

fin si no

si no c1=4

entonces

G = (float)datos[3*i+1]

B = (float)datos[3*i+2]

i2bg = (B-G)/2

i2bg = i2bg+ 127

uno[i] = (u_char)i2bg

fin si no

si no c1=5

entonces

R = (float)datos[3*i]

G = (float)datos[3*i+1]

i2gr = (G-R)/2

i2gr = i2gr+ 127

uno[i] = (u_char)i2gr

```

fin desde
escribe resultados en archivo salida
    "%d columnas ", coluno
    "%d renglones", renuno
    "%d maximo", maxuno
    "P5\n#I1\n%d %d\n%d\n", coluno, renuno, maxuno

si fnpuno = fopen (i1name,"w")==NULL)
    entonces
        imprime "ERROR: NO ES POSIBLE CREAR ARCHIVO DE
                SALIDA "
    salir

cierra fnpuno
libera uno

FIN

```

5.2 OBTENCIÓN DE LA IMAGEN DIGITAL BINARIA

La obtención de la imagen digital binaria se realiza por medio de la aplicación de un umbral global sobre una imagen digital en blanco y negro con formato PGM. Para ello, compara los valores de los píxeles con un umbral, si el umbral es mayor o igual al valor del píxel, el nuevo valor será cero, de lo contrario, será igual a 255.

Una vez que obtiene la imagen digital binaria, el pseudocódigo elimina el ruido, de acuerdo a los lineamientos expuestos en el capítulo 4.

```

FILE *fnpuno
int      coluno, renuno, sizeone, maxuno, i, j, k, l, cont, d1, d2, d3,
        d4, d5, d6, d7, d8, d9, num
u_char *datos,*datos3
char   arg[200]

```

INICIO

```

imprimir "Uso: <nombre_archivo_entrada> <nombre de archivo_salida>

```

```

        <numero_umbral>"
lee argv[1]
lee argv[2]
lee num
datos = lee_pgm(argv[1], &coluno, &renuno, &maxuno)
sizeone = coluno*renuno

si ( uno = (u_char *) calloc (sizeone, sizeof(u_char ))) == NULL)
    imprimir "ERROR!. NO ES POSIBLE RESERVAR MEMORIA"
    salir

desde i=0 hasta i<sizeone i++
    si datos[i] >= num
        entonces datos[i] = 0
    si no datos[i] = 255
fin desde

desde i=0 hasta i<renuno i++
    desde j=0 hasta j<coluno j++
        si datos[i*coluno+j] = 255 (punto blanco)
            entonces
                si existe solo un punto blanco cercano
                    entonces
                        datos[i*coluno+j] = (punto
blanco cercano) = 1
                fin si
            fin si
        si no datos[i*coluno+j] = datos[i*coluno+j]
    fin desde j
fin desde i

desde i=0 hasta i<renuno i++
    desde j=0 hasta j<coluno j++
        si datos[i*coluno+j] = 1
            entonces datos3[i*coluno+j] = 0
        si no datos3[i*coluno+j] = datos[i*coluno+j]
    fin desde j
fin desde i

```

escribe resultados en archivo salida

"%d columnas ", coluno

"%d renglones", renuno

"%d maximo", maxuno

"P5\n#I1\n%d %d\n%d\n", coluno, renuno, maxuno

si fnpuno = fopen (i1name,"w")==NULL)

entonces

imprime "ERROR: NO ES POSIBLE CREAR ARCHIVO DE
SALIDA "

salir

cierra fnpuno

libera datos3

FIN

5.3 MULTIPLICACIÓN DE LA IMAGEN DIGITAL BINARIA CON LA IMAGEN DIGITAL ORIGINAL

Para realizar la multiplicación de una imagen digital binaria con formato PGM por una imagen digital con formato PPM se aplica el procedimiento que se explica en el Apéndice A de este documento.

El procedimiento consiste en obtener los valores de pixel de la imagen digital binaria, guardarlos en una variable de tal forma que si el valor del pixel es cero, se le dará el valor de cero, y si es de 255 se le dará el valor de uno. Esta variable se multiplica por los valores de pixel de la imagen digital en color, por cada una de sus bandas de color.

Dependiendo de si se desea obtener solo una imagen digital que muestre la segmentación, entonces si el resultado de la multiplicación es cero, el valor del pixel la imagen digital final será cero, de lo contrario, será igual al valor de la imagen digital en color original.

Si se desea cambiar el color de la región segmentada en la imagen digital original, entonces si el resultado de la multiplicación es cero, el valor del pixel de la imagen digital final será igual al valor del pixel de la imagen digital en color original. Si el resultado es diferente de cero, al pixel de la imagen digital final se le dará los valores de la tríada del nuevo color.

```
FILE *fnpuno
int      coluno, renuno, sizeone, maxuno, i, val, mult, col, ren,
nrojo,
        nverde, nazul, c1
char  i1name[200]
u_char *datos, *datos2, *datos3
```

INICIO

```
imprimir "Uso: <nombre_archivo_ppm> <nombre_archivo_binario>
        <nombre de archivo_salida>"
```

```
lee argv[1]
lee argv[2]
lee argv[3]
```

```
datos = lee_ppm(argv[1], &coluno, &renuno, &maxuno)
datos2 = lee_pgm(argv[2], &coluno, &renuno, &maxuno)
sizeone = coluno*renuno*3
```

```
si ( (uno = (u_char *) calloc (sizeone, sizeof(u_char))) == NULL)
    imprimir "ERROR!. NO ES POSIBLE RESERVAR MEMORIA"
    salir
```

```
imprimir "1: Color original  2: Cambiar color y deja igual el resto de la
        imagen"
```

```
imprimir "Opción: "
lee c1
```

```
    desde i=0 hasta i<coluno*renuno i++
        si c1 =1
```

```
            entonces
                val = (int )datos2[i]
```

```
                si val = 255
```

```
                    entonces
```

```
                        mult = 1
```

```
                    si no
```

```
                        entonces
```

```
                            mult = 0
```

```

                                datos3[3*i] = (datos[3*i]*mult )
                                datos3[3*i+1] =
(datos[3*i+1]*mult )
                                datos3[3*i+2] =
(datos[3*i+2]*mult )
                                fin si no
                                fin si
si no c1=2
    entonces
        imprime "NUEVO ROJO: "
        lee nrojo
        imprime "NUEVO VERDE: "
        lee nverde
        imprime "NUEVO AZUL: "
        lee nazul

        desde i=0 hasta i<coluno*renuno i++
            val = (int )datos2[i]

            si val= 255
                entonces mult = 1
                    si no mult = 0

                    si mult =0
                        entonces

datos3[3*i]=datos[3*i]
                                datos3[3*i+1]=datos[3*i+1]
                                datos3[3*i+2]=datos[3*i+2]
                                fin si

                                si no
                                    datos3[3*i]= nrojo
                                    datos3[3*i+1]= nverde
                                    datos3[3*i+2]= nazul

                                fin si no
                                fin desde j
                                fin desde i

```

escribe resultados en archivo salida

```
"%d columnas ", coluno  
    "%d renglones", renuno  
"%d maximo", maxuno  
"P5\n#I1\n%d %d\n%d\n", coluno, renuno, maxuno
```

```
si fnpuno = fopen (i1name,"w")==NULL)
```

```
    entonces
```

```
        imprime "ERROR: NO ES POSIBLE CREAR ARCHIVO DE  
                SALIDA "
```

```
        salir
```

```
    cierra fnpuno
```

```
    libera datos3
```

```
    FIN
```

Capítulo 6

PRUEBAS

En este capítulo se presentan los resultados obtenidos al implementar, primero, las fórmulas de las componentes *I1*, *I2* e *I3* del Sistema de Color *I1I2I3*, y después las fórmulas modificadas.

La figura 6.1 muestra la imagen digital en color en formato PPM que sirvió de base para el desarrollo de esta tesis. Es un fragmento de la carta topográfica de Oaxaca, clave E14-6, en escala 1: 250 000, publicado por el Instituto Nacional de Estadística, Geografía e Informática (INEGI).

Esta imagen digital contiene la información más representativa de la carta topográfica completa. A esta imagen digital en adelante se le llamará “imagen original”.

La información es la siguiente:

Simbología:	Región de Color:
Área de bosque o selva denso, agricultura	Región verde
Carretera pavimentada	Línea roja
Terracería	Línea roja y blanca
Curvas de nivel	Líneas café
Lugares poblados	Zona amarilla
Corrientes de agua	Líneas azules
Lagos	Región azul

Tabla 6.1: Simbología.

Para fines prácticos se omitirá la simbología de la carta, y en su lugar solo se utilizarán los nombres de los colores, es decir, de las regiones de interés por segmentar.

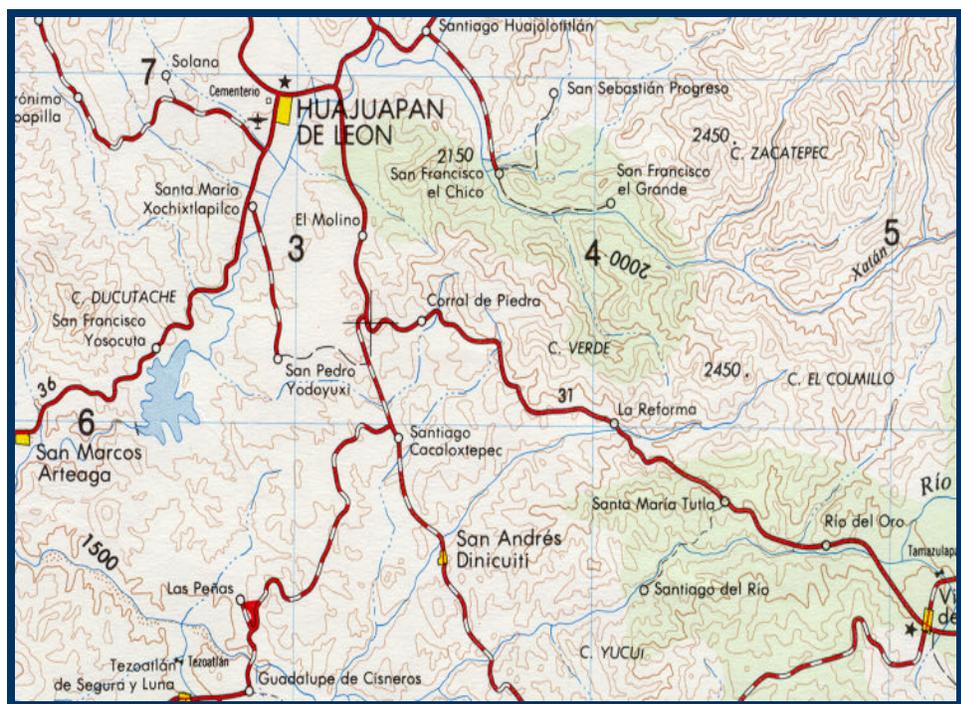


Figura 6.1: Imagen digital de trabajo (fragmento de la carta topográfica de Oaxaca).

6.1 COMPONENTES I1I2I3

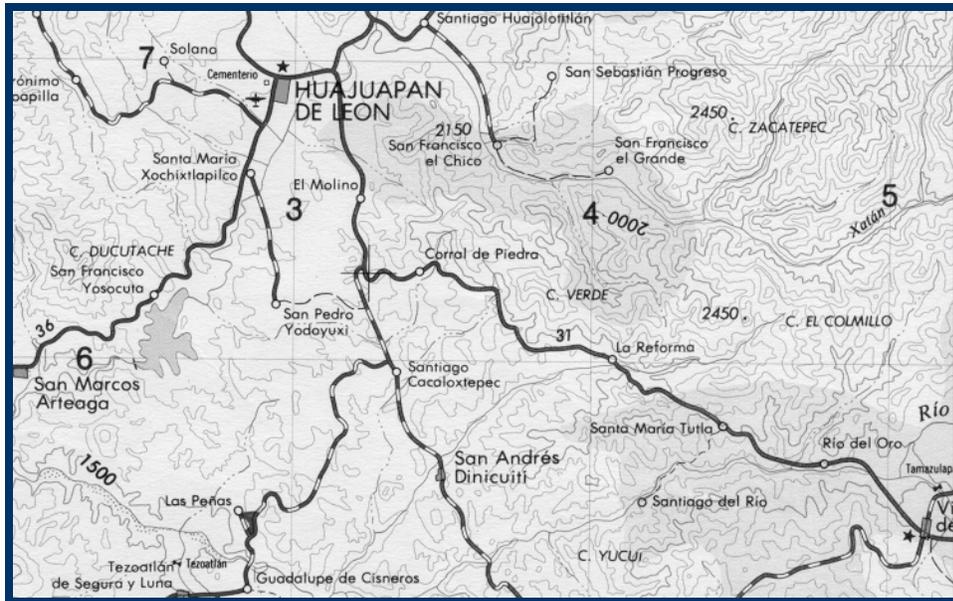


Figura 6.2: Imagen resultado de aplicar la fórmula de la componente I1.

Cada una de las fórmulas de las componentes I1, I2 e I3 se aplicaron a la imagen original obteniendo una imagen digital por cada componente. Estas imágenes digitales son en tonos de gris, y tienen el formato PGM.

La región en tonos blancos en todas las imágenes digitales generadas se presenta como una región en tono gris equivalente a la tríada de valores alrededor de 128.

La componente I1 corresponde a la intensidad. El resultado de la aplicación de la fórmula (11) es una imagen digital en tonos de gris de la original (ver figura número 6.2).

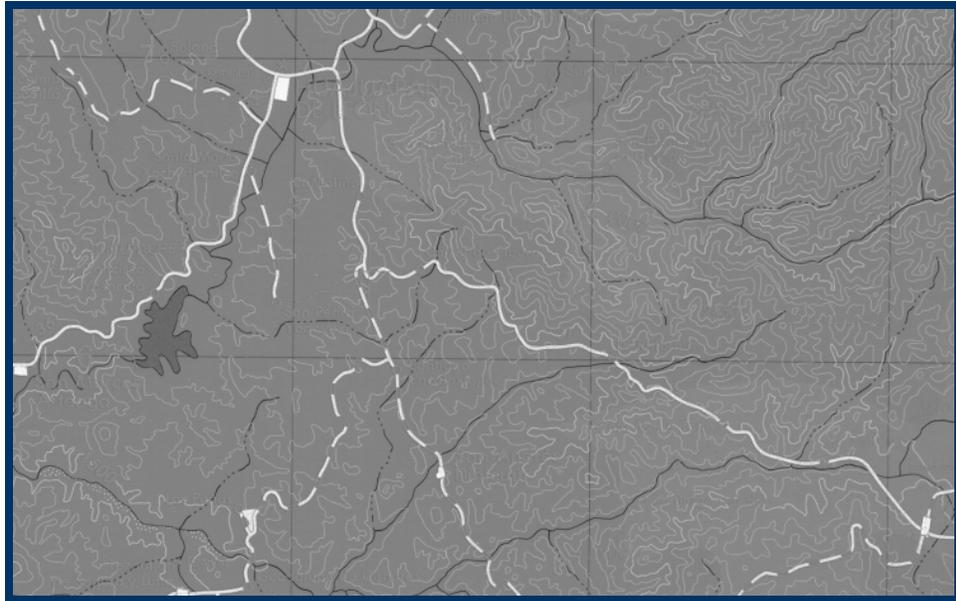


Figura 6.3: Imagen resultado de aplicar la fórmula de la componente I2.

Al aplicar la fórmula (12) de la componente I2 se genera una imagen digital que muestra en tonos claros las regiones correspondientes a los colores rojo, amarillo, y muy tenue, café. La región de color azul se encuentra en tonos oscuros (ver figura número 6.3).

De las tres componentes, esta es la imagen digital generada que presenta un mayor número de regiones definidas, es decir, mayor información, lo cual facilita su segmentación.



Figura 6.4: Imagen resultado de aplicar la fórmula de la componente I3.

La imagen digital generada por la fórmula (13) de la componente I3 muestra en tonos oscuros la región de color rojo; en tonos claros la región amarilla y de manera muy tenue la región verde (ver figura número 6.4).

6.2 FÓRMULAS MODIFICADAS



Figura 6.5: Imagen resultado de aplicar la fórmula (15), de la componente I2 modificada (I2_GR)

La imagen digital generada al aplicar la fórmula (15) de la componente I2 modificada (I2_GR), muestra en tonos claros las regiones de color azul, verde, y en tonos oscuros las regiones rojo, amarillo y café (ver figura número 6.5).

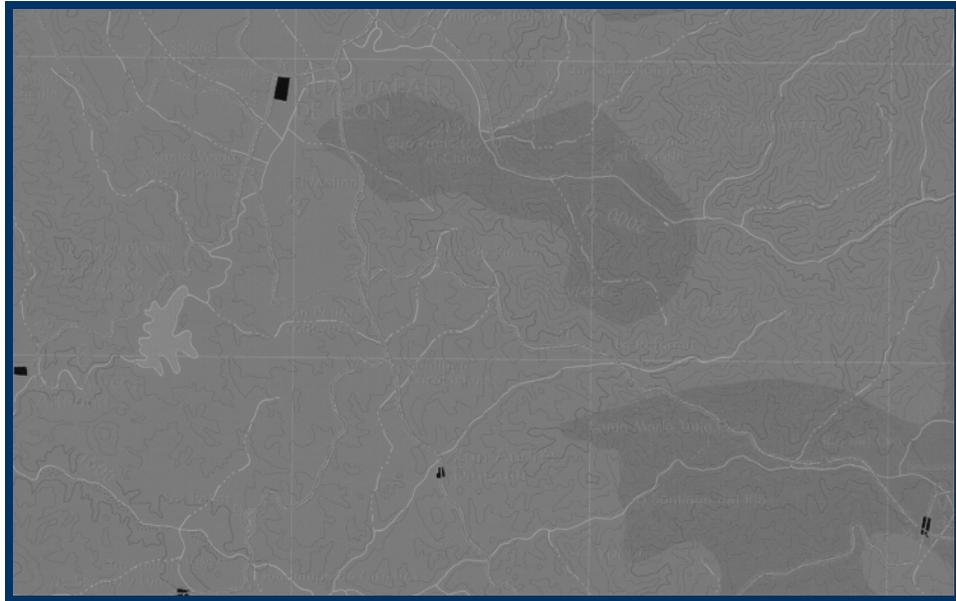


Figura 6.6: Imagen resultado de aplicar la fórmula (16), de la componente I2 modificada (I2_BG)

La imagen digital creada al aplicar la fórmula (16) de la componente I2 modificada (I2_GB), muestra en tonos claros las regiones de color azul y rojo; en tonos oscuros las regiones verde, amarillo y café (ver figura número 6.6).

A diferencia de la aplicación de la componente I3 que también muestra el color verde, esta imagen digital la define de mejor manera. Esto se observa al notar que se distinguen las líneas cafés que atraviesan la región verde. Es por ello que esta imagen digital se utilizó para segmentar la región verde.

Además, en esta imagen digital se separa el color amarillo del rojo, que en las imágenes digitales anteriores habían aparecido como una sola región. Resalta en tono oscuro la región amarilla, lo que permite su segmentación.

6.3 PROCEDIMIENTO DE SEGMENTACIÓN



Figura 6.7: Imagen binaria generada a partir de la imagen de la fórmula I2_BG (figura número 6.6).

6.3.1 REGIÓN DE COLOR VERDE

Como primer ejemplo, se presenta la región de color verde. La imagen digital I^* elegida es I_2 que se obtuvo al aplicar la fórmula modificada (16), la cual involucra a las bandas de color azul y verde.

Al igual que en todos los ejemplos, una vez encontrado el umbral adecuado, se eliminó el ruido. El umbral fue del valor 120. El resultado se puede ver en la figura número 6.7.



Figura 6.8: Imagen que muestra la segmentación de la región verde.

Una vez obtenida la imagen digital binaria (ver figura número 6.7), se multiplicó con la imagen original (ver figura número 6.1), para obtener la segmentación de la región de color verde.

El resultado es una imagen digital que muestra una región verde sobre un fondo negro (ver figura número 6.8).

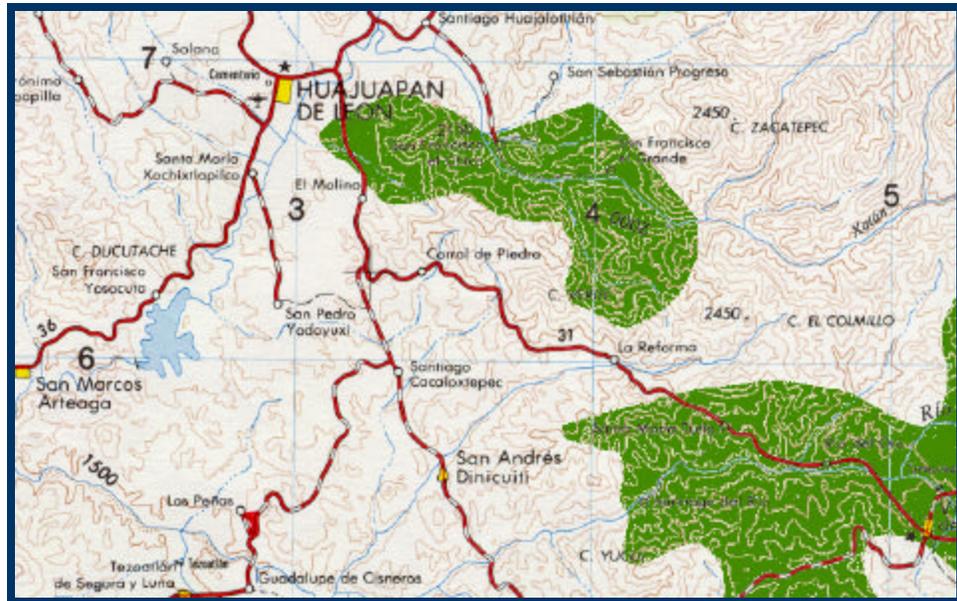


Figura 6.9: Imagen original con la región verde de diferente color.

Otro resultado que es posible obtener al multiplicar la imagen binaria con la imagen original, es cambiar el color de la región segmentado sobre la imagen original, obteniendo una nueva imagen.

En la figura número 6.9 se modificó el color de la región verde claro de la original por uno más oscuro. La tríada de valores de este nuevo color correspondientes a los valores de *RGB* son: 69, 139, 0.

6.3.2 REGIÓN DE COLOR ROJO



Figura 6.10: Imagen binaria de la región de color rojo.

En este caso, para segmentar la región de color rojo se empleó la imagen digital obtenida de la aplicación de la fórmula (15), que es modificación de la componente I2 (I2_GR), (ver figura número 6.5), en donde resalta la región de color rojo en tonos oscuros, con un umbral de 73. El resultado se muestra en la figura número (6.10).

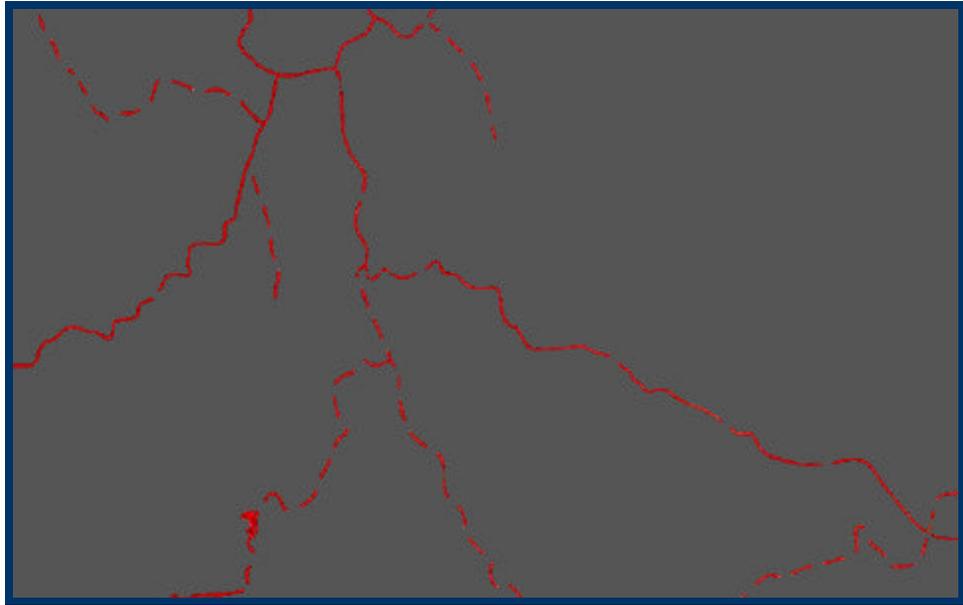


Figura 6.11: Segmentación de la región de color rojo.

Al multiplicar la imagen digital binaria de la región rojo (ver figura número 6.10) por la imagen digital original (ver figura número 6.1) se obtuvo la segmentación de la región de color rojo (ver figura número 6.11).



Figura 6.12: Imagen original con la región roja de diferente color.

Se modificó la región de color rojo en la imagen digital original al multiplicar la imagen digital binaria de la región roja por la imagen digital original. En este caso, se cambió por un color naranja (ver figura número 6.12), cuya tríada *RGB* es: 240, 128, 128.

6.3.3 REGIÓN DE COLOR CAFÉ

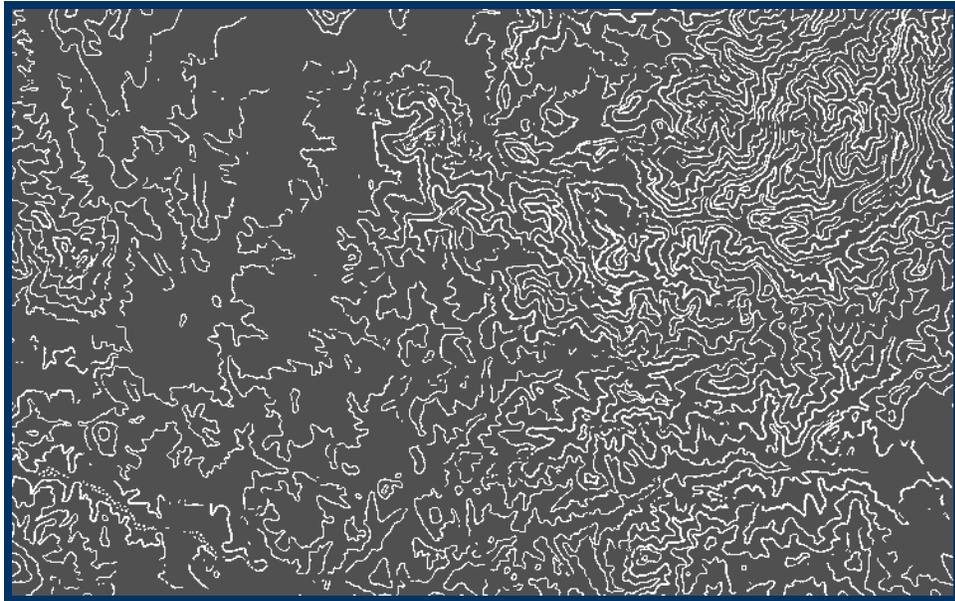


Figura 6.13: Imagen binaria de la región de color café.

Para segmentar la región de color café se eligió la imagen digital de la componente I2 (ver figura número 6.3), correspondiente a la fórmula (12), y un umbral con valor de 119.

La figura número 6.13 muestra la imagen binaria de la región de color café.

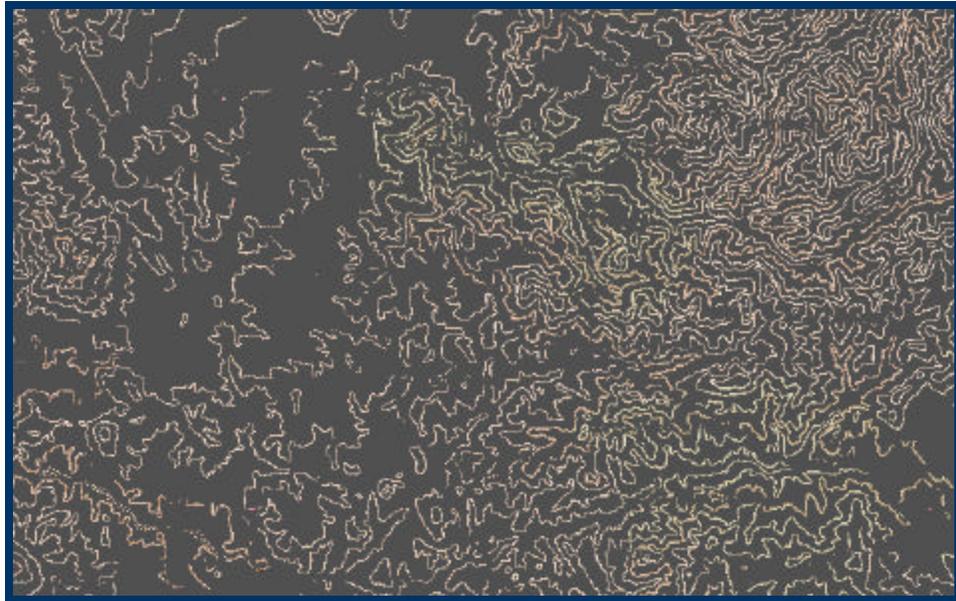


Figura 6.14: Segmentación de la región de color café.

La segmentación de la región café se realizó al multiplicar la imagen digital binaria de la región café (ver figura número 6.13) por la imagen digital original (ver figura 6.1).

El resultado se puede observar en la figura número 6.14.

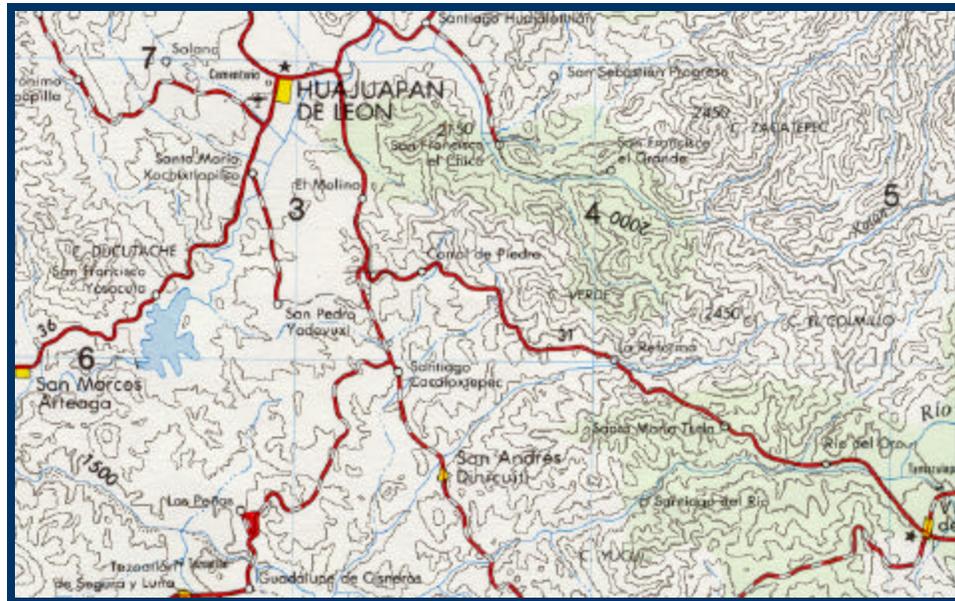


Figura 6.15: Imagen original con la región de color café de un color distinto.

Se modificó el color de la región café en la imagen digital original por uno más oscuro. La tríada *RGB* de el color nuevo es: 139, 131, 120. La figura número 6.15 muestra el resultado obtenido.

6.3.4 REGIÓN DE COLOR AMARILLA

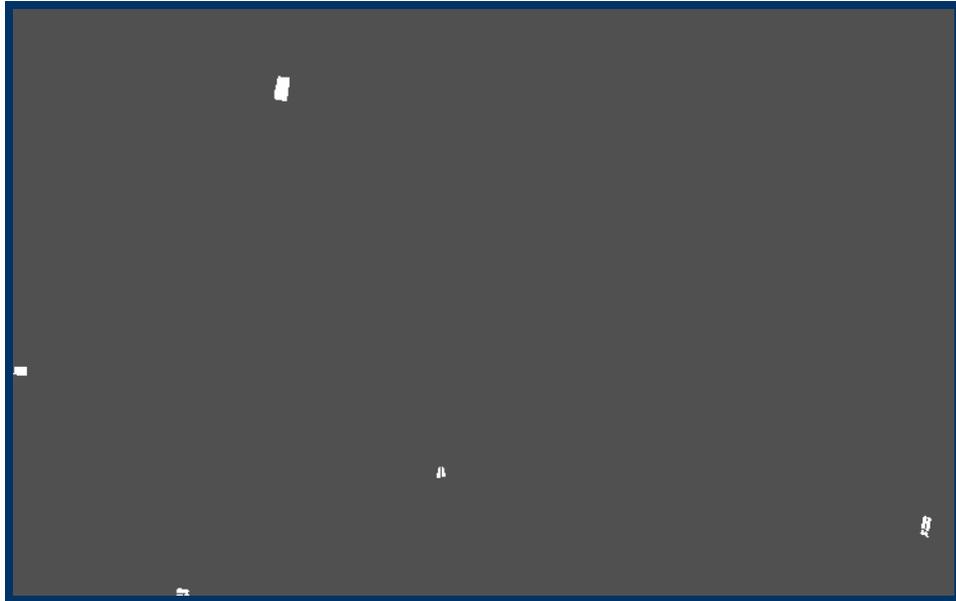


Figura 6.16: Imagen binaria de la región de color amarillo.

La región de color amarillo es la más pequeña de todas. En algunas imágenes digitales I^* la región amarilla aparece del mismo tono que la región de color rojo.

La imagen digital $I2_BG$ (ver figura número 6.6) muestra ya separada la región amarilla de la roja en un tono de color más oscuro que el resto de la imagen digital, por lo que se eligió para su segmentación. El umbral usado es el valor de 97.



Figura 6.17: Segmentación de la región de color amarillo.

Al igual que en los casos anteriores al multiplicar la imagen digital binaria correspondiente por la imagen digital original se obtuvo la segmentación de la región amarilla (ver figura número 6.17).

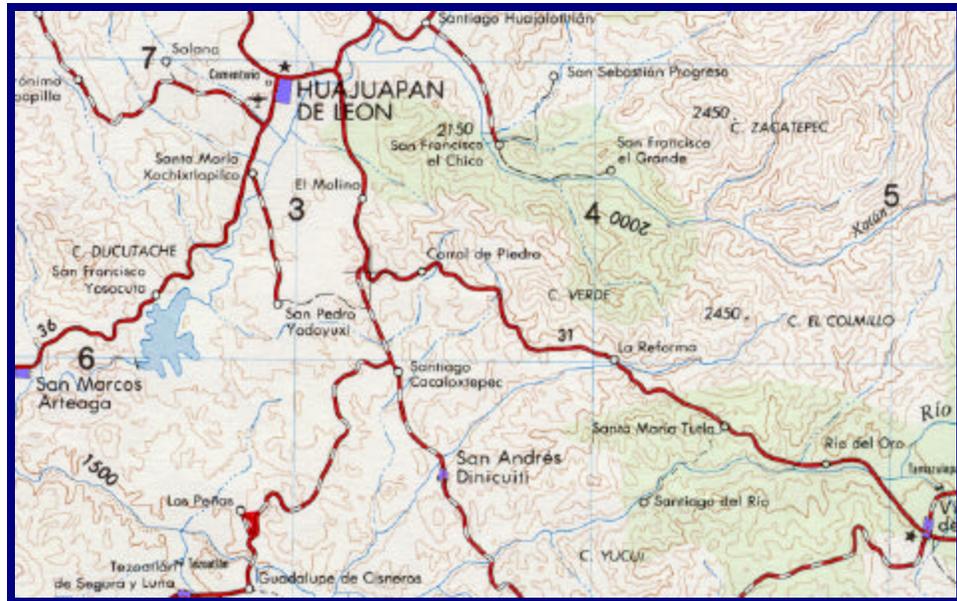


Figura 6.18: Imagen original con la región amarilla de diferente color.

En la figura número 6.18 se ha cambiado la región amarilla de color sobre la imagen digital original. El nuevo color es un tono de color morado, y los valores de la tríada *RGB* son: 139, 26, 26.

6.3.5 REGIÓN DE COLOR AZUL

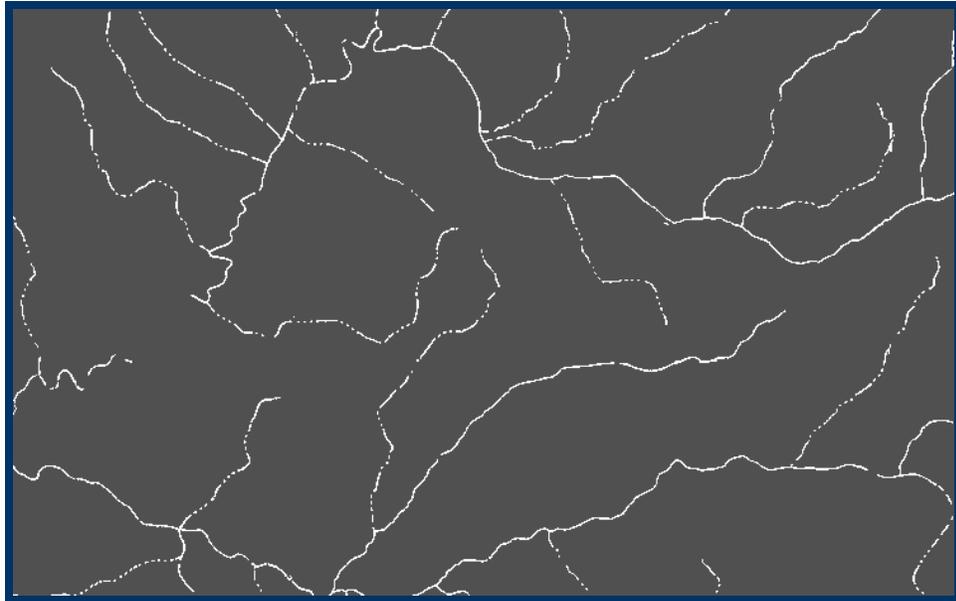


Figura 6.19: Imagen binaria de la región azul correspondiente a los ríos.

En el caso de la región de color azul, se distinguieron dos tonos de azul, uno referente a los ríos y otro, más claro, a un lago.

Para obtener la imagen binaria de la región azul se utilizó la imagen digital de la componente I2 (ver figura 6.3).

El umbral para obtener la imagen digital binaria de la región de los ríos(ver figura número 6.19) es diferente a la imagen digital binaria de la región del lago (ver figura número 6.22). Los umbrales fueron de 124 para los ríos y de 115 para la región del lago. A ambas imágenes digitales binarias se les eliminó el ruido.

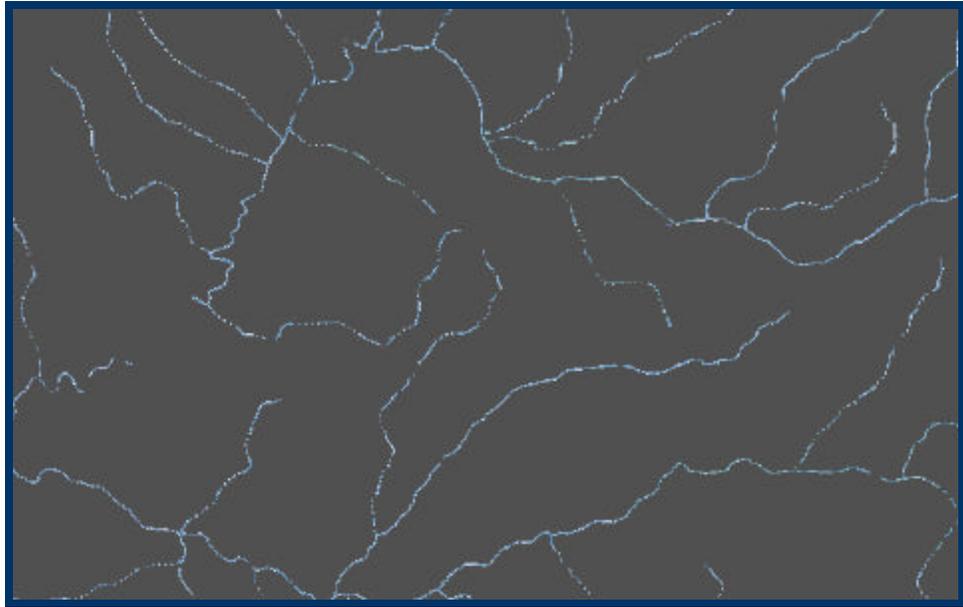


Figura 6.20: Segmentación de la región de color azul correspondiente a los ríos.

La segmentación de la región azul correspondiente a los ríos se puede ver en la figura número 6.20. La figura número 6.23 muestra la segmentación de la región azul del lago. El procedimiento en ambos casos ha sido el mismo que el empleado en los casos anteriores.



Figura 6.21: Cambio de color de la región de color azul correspondiente a los ríos en la imagen original.

A la región azul se cambió el color de azul. La figura 6.21 muestra el cambio hecho a la región de los ríos, poniéndole el color cuya tríada *RGB* es: 72, 118, 256.

En la figura número 6.24 se observa el cambio de color hecho a la región azul correspondiente al lago. La tríada *RGB* del nuevo color es: 100, 149, 237.



Figura 6.22: Imagen binaria de la región de color azul correspondiente al lago.

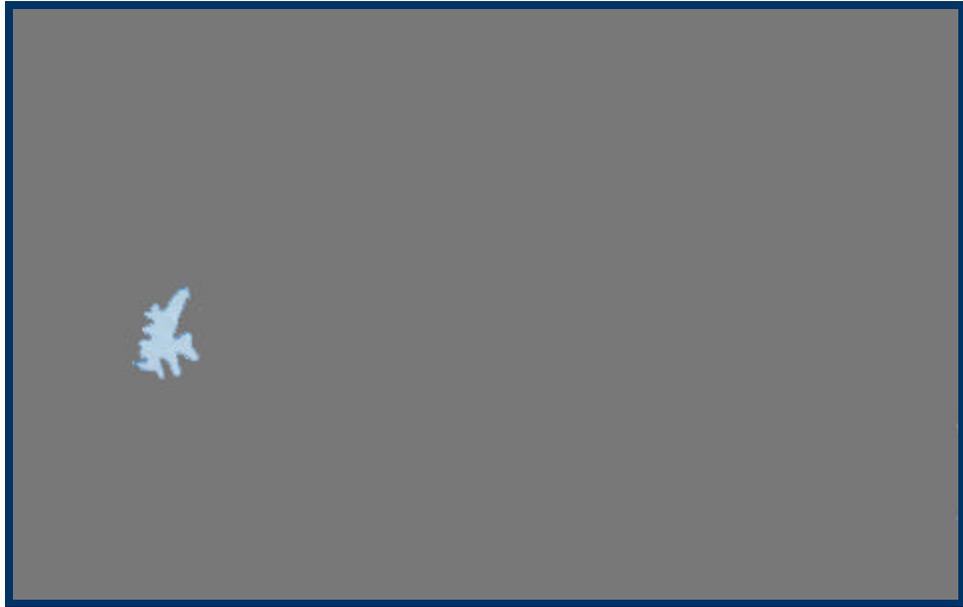


Figura 6.23: Segmentación de la región de color azul correspondiente al lago.



Figura 6.24: Cambio de color de la región de color azul correspondiente al lago sobre la imagen original.

6.3.6 REGIÓN DE COLOR NEGRO

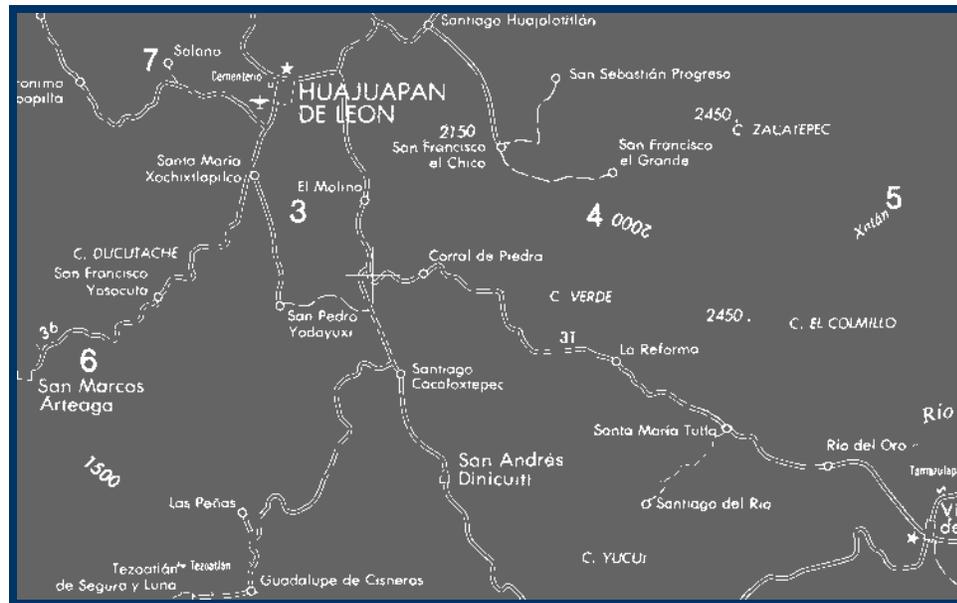


Figura 6.25: Imagen binaria de las regiones de color negro.

Para obtener la imagen digital binaria de las regiones de color negro se utilizó la imagen digital de la componente I1, aplicando la fórmula (11) (ver figura número 6.2), y un umbral de 105. La figura número 6.25 muestra la imagen digital binaria resultante.

6.4 SEGMENTACIÓN DE VARIAS REGIONES



Figura 6.28: Imagen resultado de cambiar el color de los ríos a la imagen original.

Una vez obteniendo las imágenes digitales binarias es posible generar otro tipo de mapas al combinar estas imágenes y multiplicarlas entre sí.

Las imágenes digitales binarias funcionan como plantillas, las que permiten segmentar la región que representan., y se pueden aplicar sobre cualquier imagen digital a color, ya sea de la original de la cual provienen, o alguna derivada de la original.

Como ejemplo se cambiarán de color las regiones referentes a los tonos azul (ríos y lago), amarillo, rojo y negro de la imagen digital original (ver figura número 6.1). Para ello, la segmentación se hará color por color, generando una nueva imagen digital en color que será la imagen digital base para la siguiente segmentación.

Cabe aclarar que es posible realizar la segmentación de cualquier color siempre y cuando se encuentre dentro del sistema de color *RGB*.



Figura 6.29: Imagen resultado de cambiar a la imagen de la figura número 6.28 el color del lago.

Primero, se ha multiplicado la imagen digital original por la imagen digital binaria de los ríos (ver figura número 6.19), y se les ha cambiado el color, siendo ahora de un tono más oscuro que el original. Los valores de la tríada *RGB* son: 16, 78, 139. La figura número 6.28 muestra el resultado.

Continuando con el ejemplo, la imagen digital de la figura número 6.28 se ha multiplicado por la imagen digital binaria de la región del lago (ver figura 6.22), cambiándole el color (ver figura número 6.29). El valor de la tríada *RGB* es: 100, 149, 237.



Figura 6.30: Imagen resultado de cambiar a la imagen de la figura número 6.29 el color de la región amarilla.

Ahora, la imagen digital de la figura número 6.29 se ha multiplicado por la imagen digital binaria de la región amarilla (ver figura 6.16), cambiando el color de la región (ver figura número 6.30). La tríada *RGB* del nuevo color naranja es: 255, 165, 0.

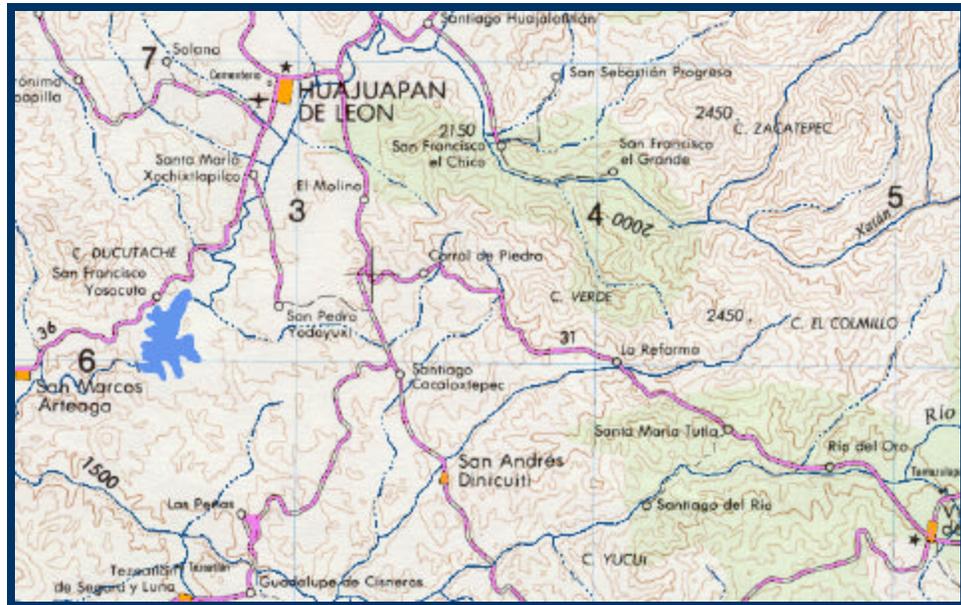


Figura 6.31: Imagen resultado de cambiar a la imagen de la figura número 6.30 el color de la región roja.

Para cambiar de color la región roja, se multiplicó la imagen digital de la figura número 6.30 por la imagen digital binaria de la región roja (ver figura número 6.10). El resultado se muestra en la figura número 6.31. El nuevo color es un tono de rosa, cuya tríada *RGB* es: 238, 130, 238.



Figura 6.32: Imagen resultado de cambiarle a la imagen de la figura número 6.31 el color de la región de color negro.

Para finalizar el ejemplo, la imagen digital de la figura número 6.31 se ha multiplicado por la imagen digital binaria de la región negra (ver figura 6.25), cambiando el color de la región por un tono de azul verde, cuya tríada *RGB* es: 0, 154, 206. La figura número 6.32 muestra el resultado.



Figura 6.33: Imagen resultado de cambiar a color blanco la región café de la imagen original.

Finalmente, además de poder cambiar el color de las regiones ya segmentadas, también es posible generar imágenes digitales en donde se “quiten” las regiones que se desee.

Esto se logra al realizar la multiplicación de igual forma, solo que el nuevo color que se le da debe ser igual al del fondo de la imagen original.

Como ejemplo, en la figura número 6.32 se han multiplicado las imágenes digitales original (ver figura 6.1) y la binaria del color café (ver figura número 6.13). El nuevo color ha sido el blanco, cuya tríada *RGB* es: 255, 255, 255.

Con ello se ejemplifica que combinando y multiplicando las imágenes digitales binarias y las de color se pueden obtener nuevas imágenes digitales, de acuerdo a las necesidades que se presenten.

Capítulo 7

RESULTADOS Y CONCLUSIONES

7.1 RESULTADOS

Para probar y verificar la eficacia del procedimiento de segmentación de imágenes digitales en color propuesto en esta tesis se aplicó en 5 imágenes diferentes.

De los resultados obtenidos se obtuvieron dos tablas, una que muestra las imágenes digitales I^* y los umbrales utilizados al segmentar.

En la sección 7.1.2 se muestran de manera gráfica los resultados obtenidos.

7.1.1 TABLA DE IMÁGENES DIGITALES I^* , UMBRALES Y COLORES

En esta sección se incluye la tabla 7.1 con el tipo de imágenes digitales I^* , los valores de los umbrales utilizados y el color segmentado. Se han ordenado de acuerdo al tipo de imagen I^* , y al valor del umbral de menor a mayor. De los colores segmentados solo dará el nombre común, de acuerdo a su apariencia a simple vista.

También se incluye una tabla 7.2, que es una tabla de color generada en la realización de esta tesis. Se han ordenado los valores de menor a mayor a partir de la banda de color rojo.

TABLAS

IMAGEN I*	UMBRAL	COLOR
I1	105	Negro
I1	170	Gris oscuro
I2	96	Azul
I2	115	Azul claro
I2	119	Azul
I2	124	Azul
I2	179	Rojo
I2	256	Naranja
I3	103	Rojo
I2GR	56	Rojo
I2GR	73	Rojo
I2GR	124	Tonos rojizos
I2GR	203	Verde
I2BG	97	Amarillo
I2BG	117	Rojo
I2BG	120	Verde
I2BG	123	Rojo oscuro
I2BG	178	Azul

Tabla 7.1: Tabla de imágenes digitales I*, umbrales y colores.

ROJO	VERDE	AZUL	COLOR
0	154	206	Azul violeta claro
16	78	139	Azul oscuro
69	139	0	Verde oscuro
72	118	255	Azul
100	149	237	Azul violeta
135	206	250	Azul
139	26	26	Morado
139	131	120	Gris azul
238	130	238	Rosa claro
240	128	128	Naranja
255	165	0	Naranja oscuro
255	228	196	Amarillo claro

Tabla 7.2: Tabla de color.

7.1.2 RESULTADOS



Imagen Original

en de una pintura al pastel, que tiene como gran diversidad de tonos de un solo color en una o cual podría significar la segmentación de una imagen dad de color de pixeles. Para resaltar este hecho, se ha pintado la región segmentada sobre la imagen original.



Segmentada

Color: Gris oscuro
Imagen I* utilizada: I1
Umbral utilizado: 170
Valor nueva tríada de color: 100, 100, 100



Segmentada

Color: Rojo intenso
Imagen I* utilizada: I2GR
Umbral utilizado: 56
Valor nueva tríada de color: 250, 250, 210



entada

Color: Azul oscuro

Imagen I* utilizada: I2GR

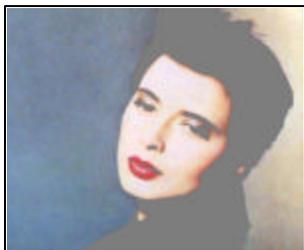
Umbral utilizado: 107

Valor nueva tríada de color: 135, 206, 250



Imagen Original

Al igual que la imagen anterior, ésta presenta cierta heterogeneidad en los colores de pixel, y también se pintado la región segmentada en la imagen original.



ada

Color: Negro

Imagen I* utilizada: I1

Umbral utilizado: 105

Valor nueva tríada de color: 125, 125, 125



ada

Color: Azul

Imagen I* utilizada: I2

Umbral utilizado: 124

Valor nueva tríada de color: 100, 100, 100



ada

Color: Rojo intenso

Imagen I* utilizada: I2GR

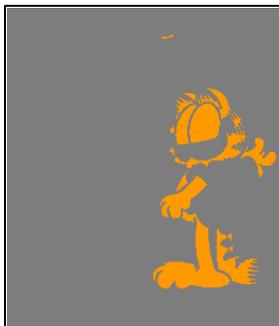
Umbral utilizado: 72

Valor nueva tríada de color: 255, 192, 203



Imagen Original

Esta imagen presentó un histograma con pocos valores de pixel, lo que facilitó la elección del valor de umbral a utilizar.



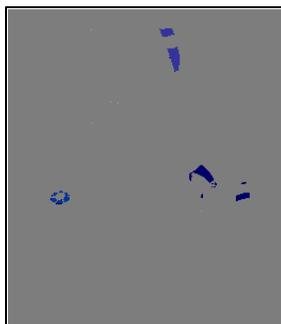
ntada

Color: Naranja
Imagen I* utilizada: I2
Umbral utilizado: 235



ntada

Color: Verde
Imagen I* utilizada: I2GR
Umbral utilizado: 203

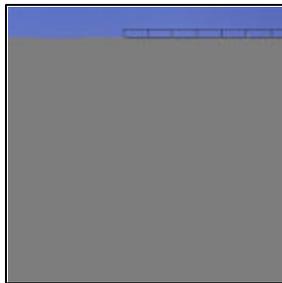


ntada

Color: Azul
Imagen I* utilizada: I2BG
Umbral utilizado: 178



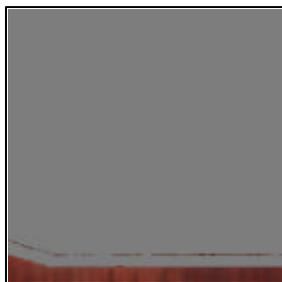
de esta imagen se distinguen dos tonos de color
consideraron para segmentar, así como la franja azul
en la parte superior.



ntada

Color: Azul
Imagen I* utilizada: I2

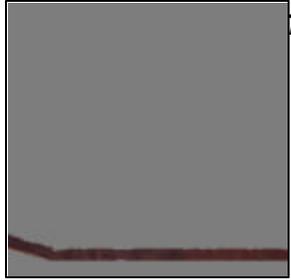
96



ntada

Color: Rojo
Imagen I* utilizada: I2BG

117



ntada

Color: Rojo oscuro
Imagen I* utilizada: I2BG

123



Imagen Original

Con esta imagen se ejemplifica que también es posible realizar la segmentación de objetos con este método, siempre y cuando presenten coloraciones similares.



Segmentada

Color: Azul

Imagen I* utilizada: I2

Umbral utilizado: 115

El valor del fondo se aclaró (tríada: 125,125,125) con el el tono de azul.

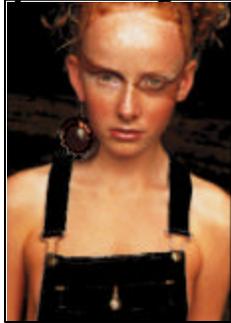


Segmentada

Color: Rojo

Imagen I* utilizada: I3

Umbral utilizado: 103



Segmentada

Color: Tonos rojizos

Imagen I* utilizada: I2GR

Umbral utilizado: 124

Se segmentaron los tonos rojizos y blancos, dando como resultado la segmentación del cuerpo de modelo de la foto.

7.2 CONCLUSIONES

Al concluir el desarrollo de esta tesis se cumplió con el objetivo de encontrar un procedimiento que permita realizar la segmentación de imágenes digitales en color. A este procedimiento se llegó a través del *Sistema de Color III2I3*, y de su modificación.

Esta modificación permite realizar la segmentación de colores que no se consideran al aplicar el sistema original, como es el caso del amarillo, y con una mejor definición el color verde y rojo. Esto significa que es posible abarcar cualquier color, siempre y cuando se encuentre dentro del sistema de color *RGB*.

Además, permite resolver el problema de segmentar imágenes digitales en color que presentan una gran heterogeneidad en los tonos de color de los píxeles que forman una región de color, que era el problema que se había presentado.

La implementación del procedimiento de segmentación en este momento se realiza de manera semi-automática, llamando desde línea de comandos a los programas de generación de imágenes digitales de I^* , de creación de imágenes digitales binarias (que elimina el ruido), y al programa de multiplicación y pintado de imágenes digitales.

Esto significa que existe gran interacción con el usuario, sobretodo al momento de elegir un umbral adecuado para generar la imagen digital binaria, así como al momento de cambiar el color de las regiones.

Lo que se propone para automatizarlo, es generar una base de datos de los colores involucrados en la imagen original base, y generar una tabla de color para los nuevos colores. Esto siempre y cuando el tipo de imagen digital sobre la cual se pretenda trabajar lo permita.

Por lo tanto, el trabajo realizado en esta tesis es solo un principio, se ha encontrado un procedimiento para segmentar imágenes digitales el cual se puede mejorar y optimizar.

Entre las posibles aplicaciones que se le pueden dar a este procedimiento se encuentran:

- Detección de caracteres y objetos de color en imágenes digitales en color.
- Modificación del color de caracteres y objetos de color en imágenes digitales en color.

- Detección de píxeles o grupos de píxeles de un determinado tono de color, sobretodo en el caso de regiones de apariencia homogénea.
- Segmentación de objetos de tonos de color similares.

La investigación de sobre la *segmentación de imágenes digitales en color* aún es un campo abierto y amplio. En esta tesis se ha encontrado una solución a un problema específico planteado. Aún queda camino por recorrer.

Apéndice A

MULTIPLICACIÓN DE IMÁGENES DIGITALES

La multiplicación de imágenes digitales es la última parte del procedimiento de segmentación de imágenes digitales en color, y es la que permite generar la imagen digital que muestra la segmentación ya realizada.

Este procedimiento se refiere a multiplicar una imagen digital binaria con formato PGM por una imagen digital a color de formato PPM, ambas de igual tamaño [3].

PROCEDIMIENTO

Las imágenes digitales de entrada serán una imagen digital binaria con formato PGM y una imagen digital en color con formato PPM, ambas de igual tamaño. Como resultado de la multiplicación se obtendrá una imagen en color con formato PPM de dimensiones iguales a las imágenes digitales de entrada [3].

La multiplicación es de pixel a pixel, iniciando de la esquina superior derecha de las imágenes. Para facilitar las operaciones, cada valor de pixel de la matriz de datos de la imagen digital binaria se guarda en una variable de tipo entero. Así, cada matriz de la imagen digital en color se multiplica por esta variable. El resultado se guarda en la matriz de datos que representa a la imagen digital nueva.

Cabe aclarar que la imagen digital binaria muestra la región de interés, en este caso la región segmentada, en color blanco y el resto en color negro.

Los valores que contiene la matriz de datos de la imagen digital binaria son 1 para los píxeles blancos, y 0 para los píxeles negros, que al multiplicarse con los valores de las matrices de datos de la imagen en color dan los siguientes resultados:

- Al multiplicarse por 1, los valores de los píxeles permanecen sin cambio, manteniendo el color original de la imagen digital en color. Estos píxeles se refieren a la región segmentada.
- Al multiplicarse por 0, los valores de los píxeles son igual a 0, es decir, en color negro.

Con este procedimiento se obtiene una imagen digital que muestra los píxeles blancos de la imagen digital binaria en color, sobre una imagen digital en negro. El color corresponde a la región segmentada de la imagen digital original en color.

CAMBIO DE COLOR DE LA REGIÓN SEGMENTADA EN LA IMAGEN DIGITAL ORIGINAL

Al modificar el procedimiento anterior se obtiene un resultado diferente. La modificación consiste en aplicar una condición antes de guardar el producto de la multiplicación en la matriz de datos nueva, con el fin de cambiar la tríada de valores de color, es decir, en lugar de mostrar el color de la imagen digital en color original, se introducirá un color nuevo.

La condición es:

- Si el producto de la multiplicación es diferente de cero, se introducen los valores de la tríada del color nuevo.
- Si el producto es cero, entonces los valores de las matrices de datos de la imagen digital en color permanecen igual.

El resultado es la imagen digital original con la región segmentada (de la imagen digital binaria) con un color diferente al original.

Apéndice B

FORMATO.H

Los programas utilizados en la implementación del procedimiento de segmentación hacen uso del programa *formato.h*, el cual lee los encabezados de las imágenes digitales y verifica que sean de formato PPM o PGM.

```
/* Programa formato.h */

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <malloc.h>
#include <sys/types.h>
#include <unistd.h>
#include <sys/stat.h>
#include <fcntl.h>

/*FUNCION QUE LEE EL FORMATO *.PGM */
unsigned char *lee_pgm(archivo,coluno,renuno,maxuno)
char archivo[200];
int *coluno, *renuno, *maxuno;

{
    FILE *fnpi;
    long pos;
    int i,sizeone;
    char s1[5],s2[5];
    unsigned char *datos;

    fnpi=fopen(archivo, "r");
```

```

if(fnpi==NULL) {      fprintf(stderr,"\n\nERROR: NO ES POSIBLE ABRIR
    ARCHIVO %s \n",archivo);
    exit(0);
}

/* LEE ENCABEZADO */
fscanf(fnpi,"%s\n",s1);
if(strcmp(s1,"P5") != NULL){ fprintf(stderr,"\n ERROR: %s NO ES
    FORMATO *.PPM\n", archivo);
    exit(0);
}

fscanf(fnpi,"%s",s2);
if(strcmp(s2,"#")==NULL){
    i=getc(fnpi);
    while(i != 10){
        i=getc(fnpi);
    }
}

/* OBTIENE NUMERO DE COLUMNAS, RENGLONES Y EL MAXIMO */
fscanf(fnpi,"%d %d\n",coluno,renuno);
fscanf(fnpi,"%d\n",maxuno);
sizeone=(*coluno)*(*renuno);

datos=(unsigned char *)calloc(sizeone, sizeof(unsigned char ));
if(datos==NULL){
    fprintf(stderr,"\nERROR!. NO ES POSIBLE RESERVAR
MEMORIA
    %s\n",archivo);
    exit(1);
}

fread(datos,1,sizeone,fnpi);
fclose( fnpi );
return(datos);
}

/*FUNCION QUE LEE EL FORMATO *.PPM */
unsigned char *lee_ppm(archivo,coluno,renuno,maxuno)
char    archivo[200];
int      *coluno, *renuno, *maxuno;

{
    FILE      *fnpi;

```

```

    long pos;
        int          i,sizeone;
        char   s1[5],s2[5];
        unsigned char *datos;

        fnpi=fopen(archivo, "r");
        if(fnpi==NULL) {          fprintf(stderr,"\n\nERROR: NO ES POSIBLE
ABRIR
        ARCHIVO %s \n",archivo);
        exit(0);
    }

/* LEE ENCABEZADO */
    fscanf(fnpi,"%s\n",s1);
        if(strcmp(s1,"P6") != NULL){ fprintf(stderr,"\n ERROR: %s
NO ES
        FORMATO *.PPM\n", archivo);
        exit(0);
    }
    fscanf(fnpi,"%s",s2);
        if(strcmp(s2,"#")==NULL){
            i=getc(fnpi);
                while(i != 10){
                    i=getc(fnpi);
                }
        }
/* OBTIENE NUMERO DE COLUMNAS, RENGLONES Y EL MAXIMO */
    fscanf(fnpi,"%d %d\n",coluno,renuno);
    fscanf(fnpi,"%d\n",maxuno);
    sizeone=(*coluno)*(*renuno)*3;

    datos=(unsigned char *)calloc(sizeone, sizeof(unsigned char ));
        if(datos==NULL){ fprintf(stderr,"\nERROR!. NO ES POSIBLE
        RESERVAR MEMORIA %s\n",archivo);
        exit(1);
    }

    fread(datos,1,sizeone,fnpi);
    fclose( fnpi );
    return(datos);
}

```

Apéndice C

MANUAL DE USO DEL PROCEDIMIENTO DE SEGMENTACIÓN DE IMÁGENES DIGITALES

El procedimiento de segmentación de imágenes digitales en color se desarrolló sobre el sistema operativo Linux (versión 6.3), el cual cuenta con el ambiente gráfico Gnome.

El procedimiento consta de tres programas en lenguaje C, los cuales se ejecutan desde línea de comandos. Para ello, se usa el ejecutable del mismo.

Además de utilizar un shell para ejecutarlos, se hace uso de un visor de imágenes digitales, en este caso, se recomienda el Gqview, de Gnome, ya que permite hacer el despliegue secuencial de varias imágenes digitales a la vez. Dentro del ambiente gráfico Gnome también se encuentra el programa Gimp.

1. GENERACIÓN DE IMÁGENES DIGITALES I* (I1, I2, I3, I2_GR, I2_BG)

El primer programa a utilizar es el *i123bgr.c*, con ejecutable del mismo nombre. Este programa a partir de una imagen digital en color (formato PPM) genera una imagen digital en blanco y negro de cada una de las fórmulas del Sistema de Color I1I2I3 y las dos fórmulas derivadas. Crea una imagen a la vez.

Para ejecutarlo se introduce la siguiente línea:

```
./i123bgr nombre_imagen.ppm nombre_imagen_final
```

Primero se escribe el nombre de la imagen digital original, incluyendo la extensión que debe ser *.ppm. Enseguida se agrega el nombre que se desea dar a la imagen saliente sin extensión.

El programa muestra un menú de opciones del tipo de fórmula I* a usar:

1: I1 2: I2 3: I3 4: I2_BG 5: I2_GR

La imagen digital generada tendrá el nombre ya determinado más la extensión *.pgm, ya que esta imagen será en blanco y negro.

Para facilitar la identificación del tipo de fórmula usada para crear la imagen, después del nombre se agrega un subguión seguido de las iniciales de cada fórmula. Por ejemplo:

1: *nombre-imagen-final_I1.pgm*

2: *nombre-imagen-final_I2.pgm*

3: *nombre-imagen-final_I3.pgm*

4: *nombre-imagen-final_I2_bg.pgm*

5: *nombre-imagen-final_I2_gr.pgm*

La elección de la imagen digital I* se hace con el auxilio del visor Gqview, el cual permite desplegar las imágenes digitales.

2. OBTENCIÓN DE LA IMAGEN DIGITAL BINARIA

Una vez elegida la imagen digital I* se procede obtener una imagen digital binaria por medio de una segmentación global o un umbral.

El umbral se elige de tal forma que si el valor del pixel es mayor o igual que el umbral, lo mapea a 0, y si no, a 255. Se recuerda que el valor 0 significa un pixel de color negro y el valor 255 el pixel blanco.

La elección del umbral se realiza con el auxilio del histograma de la imagen digital. Si no se cuenta con un programa que lo cree, el Gimp cuenta con una función que lo realiza.

Para generar la imagen digital binaria se emplea el programa *binario.c*, con ejecutable del mismo nombre.

Al igual que en el programa anterior se introduce el nombre de la imagen digital original con su extensión (*.pgm) y el nombre que se desea poner a la imagen binaria generada sin extensión. Además, se debe agregar el número del valor del umbral.

La línea queda como sigue:

```
./binario nombre_imagen.pgm nombre_imagen_final número_umbral
```

Este programa además elimina el ruido, siendo este un pixel blanco aislado o dos pixeles adyacentes. La imagen resultante se podrá ver por medio de un visor de imágenes.

3. MULTIPLICACIÓN DE LA IMAGEN DIGITAL BINARIA CON LA IMAGEN DIGITAL ORIGINAL

Como parte final de este procedimiento, se genera la imagen digital en color que muestra la segmentación ya realizada, o bien si se desea, se cambia de color a la región segmentada sobre la imagen digital original.

Para ello es necesario ya tener la imagen digital binaria con la región segmentada y la imagen digital en color original.

El programa a usar es el *multipin.c*, cuyo ejecutable tiene el mismo nombre.

En este caso es necesario escribir en la línea de comandos tres nombres: el de la imagen digital original con su extensión (*.ppm), el de la imagen binaria con su extensión (*.pgm), y el nombre que se desea dar a la imagen resultante sin extensión. La línea queda como sigue:

```
./multipin nombre_imagen.ppm nombre_imagen_binaria.pgm  
nombre_imagen_final
```

El programa presentará dos opciones:

1: Color original

2: Cambiar color y deja igual el resto de la imagen

La elección 1 genera una imagen digital con la región segmentada en el color original sobre un fondo negro.

La elección 2, en cambio genera una imagen digital con la región segmentada con un color diferente al color original sobre la imagen digital original.

El programa pide la tríada *RGB* del nuevo color con que se desea “pintar” la región segmentada. Pedirá:

NUEVO ROJO:

NUEVO VERDE:

NUEVO AZUL:

Si no se cuenta con una tabla de color de donde se pueda elegir un nuevo color, el ambiente gráfico Gnome cuenta con un programa llamado “Paleta de colores”, que se puede encontrar en las “Utilerías” de la barra de inicio. La imagen final se podrá ver por medio de un visor de imágenes.

Bibliografía

- [1] ALLEBACH, Jan,
Processing digital color images: From capture to display,
Phisycs Today, Vol. 45 No. 12, Diciembre, 1992.
- [2] AYRES, Frank,
Teoría y problemas de matrices,
Serie de compendios de Shaum, México, Ed. McGraw-Hill, 1976.
- [3] BÁEZ R., José,
Notas del Curso de Procesamiento Digital de Imágenes
Universidad Tecnológica de la Mixteca, México,
Marzo-Junio de 1999.
- [4] CASTLEMAN, Kenneth R.,
Digital Image Processing,
USA, Ed. Prentice Hall, 1996.
- [5] DEITEL, H.M. y DEITEL, P.J.,
Como programar en C/C++,
Segunda Ed., México, Ed. Prentice-Hall Hispanoamericana, S.A.,
1995.
- [6] ENCICLOPEDIA DE LAS CIENCIAS,
El mundo del color,
Editorial Cumbre, S.A., Vol. 5, pg. 115-129, México, 1981.
- [7] GEVERS, Theo,
Color Image Invariant Segmentation and Retrieval,
Ph.D. Thesis, ISBN 90-74795-51-X, University of Amsterdam,
The Netherlands, 1996.
- [8] GONZÁLEZ, Rafael y WOODS, Richard,

Tratamiento Digital de Imágenes,
USA, Copublicación de Addison-Wesley Iberoamericana, S.A.
y Ediciones Díaz de Santos, S.A.,1996.

- [9] JAIN, R.; KASTURI, R.; SCHUNCK, B.
Machine Vision,
Singapore, McGraw-Hill International Editions, 1995.
- [10] ITO, Naoko; SHIMAZU, Yoshihisa,
**Fuzzy logic based non-parametric color image segmentation
with optional block processing,**
Department of Instrumentation Engineering, Faculty of Science and
Technology, Keio University.
- [11] KAMRAN, Husain,
Linux unleashed,
Segunda edición, USA, Sams Publishing, 1996.
- [12] LIM, Jae S.,
Two-dimensional signal and imagen processing,
USA, Ed. Prentice-Hall, 1996.
- [13] TEUBER, JAN,
Digital Image Processing,
USA, Ed. Prentice Hall, 1993.

LIGAS EN INTERNET

- [14] http://www.cs.cmu.edu/~maxwell/vision/AP_Syllabus.htmCMU
Advanced Perception Seminar
- [15] [http://www.ai.polymtl.ca/webLab/SMART/Facet2DocA/
Facet2DocA.html](http://www.ai.polymtl.ca/webLab/SMART/Facet2DocA/Facet2DocA.html)
**A Point-based Approach Based on Stereo Matching for the
Interposition Problem in Augmented Reality**
Claude Duchesne and Jean-Yves Hervé
- [16] <http://ls7www.cs.unidortmund.de/~schroete/paper/html/>

[erlangen97/paper.html](#)

Automatic Calibration of Lookup-Tables for Color Image Segmentation

Universität Dortmund, FB Informatik LS VII
Otto-Hahn Str. 16, D-44221 Dortmund, Germany
Sven Schröter

- [17] <http://wcca.ifas.ufl.edu/archive/7thProc/DECATRO/DECATRO.htm>

Automatic classification system of soil cover based on color information

- [18] <http://iris.usc.edu/Outlines/segment/color-seg.html>

Color segmentation high resolution aerial images

- [19] <http://www.comp.glam.ac.uk/pages/staff/tjones/ThesisOL/Title.htm>

Improving the precision of leg ulcer area measurement with active contour models

Timothy David Jones
Hospital for Rheumatic Diseases, Bath and the University Hospital of Wales, Cardiff, May 1999

- [20] <http://itzamna.uam.mx/alex/proyecto/proyecto1.htm>

Procesamiento digital de imágenes colposcópicas

- [21] www.nlm.nih.gov/research/visible/vhp_conf/imiels/nlmseg.htm

Semi-automated color segmentation of anatomical tissue

Celina Imielinska, Michael Downes, Swaroop Hosakere, Asim Khan, Wei Yuan
Dept. of Electrical Engineering and Computer Science
Stevens Institute of Technology

- [22] <http://www.palantir.swarthmore.edu/~maxwell/classes/e27/F98/syllabus.htm>

Syllabus for E93: Directed Project in Computer Vision (Fall '98)

[23] <http://www.nlm.nih.gov/research/visible/vhpconf98/AUTHORS/IMIELINS/IMIELINS.HTM>

Technical Challenges of 3D Visualization of Large Color Data Sets

Celina Imielinska, Ph.D., Lisa Laino-Pepper, M.S.&, Richard Thumann, M.S.&, Ryan Villamil

Department of Computer Science, Columbia University

Office of Scholarly Resources,

College of Physicians and Surgeons of Columbia University,

New York, NY 10032