

ARQUITECTURA DE SW Y MODELO DE UN SISTEMA PARA ACCESO POR CONTENIDO A IMÁGENES

Humberto A. Gómez C.
Universidad de los Andes
hu-gomez@uniandes.edu.co

RESUMEN

Consultar datos es una labor que debe cumplir toda aplicación de base de datos. Sin embargo, si éstos datos son imágenes la labor es mas complicada. ¿Cómo es el índice de acceso? ¿Cómo se hace una consulta? ¿Cómo se determina si una imagen es “similar” a otra?.

Para responder estas preguntas, se propone un modelo de un sistema de acceso por contenido a imágenes de textiles. Este modelo soporta el almacenamiento de imágenes 2D de textiles con varios colores. El índice se basa en el tramado de la representación monocromática de la imagen. El tramado es un indicador de la dirección o direcciones de los hilos más notorios del tejido. El acceso se lleva a cabo dando diferentes criterios que puede cumplir una imagen: tramado, color, rotación, ubicación de regiones.

Palabras claves: Bases de datos, acceso por contenido, registro de imágenes

CONTENIDO

1. INTRODUCCION
2. ARQUITECTURA DEL SISTEMA (Almacenamiento y Consultas)
3. LENGUAJE DE CONSULTAS
4. EVALUACIÓN DE CONSULTAS
5. MODELO DEL SISTEMA (Algoritmos y Complejidad)

1. INTRODUCCION

Desde que la multimedia hizo su aparición, la comunidad de la computación se ha visto enfrentada a diferentes retos para lograr a cabo un buen desempeño de las aplicaciones que usan este nuevo medio. Uno de éstos retos ha sido la fácil manipulación de imágenes en bases de datos. Aunque desde mucho tiempo atrás se han desarrollado técnicas para el tratamiento de imágenes, estas solamente se han explotado en el área de almacenamiento y manipulación. En estos aspectos se ha evolucionado bastante, logrando un muy buen desarrollo. En el área de consultas, sin embargo, la evolución ha sido diferente. Consultar directamente sobre la imagen y no sobre atributos definidos externamente con criterio de alguien es una necesidad presentada por usuarios que manejan imágenes.

El manejo dado hasta el momento para el acceso a imágenes se ha basado en acceso al contenido, no acceso por contenido. Acceso al contenido se refiere a utilizar mecanismos de indexación basados en propiedades descriptivas o de clasificación de la imagen. En este caso la imagen se clasifica de acuerdo a algún criterio, p.e. un título, o usando mecanismos de referencias cruzadas, los cuales pueden dar un acercamiento a la búsqueda de imágenes. El problema que surge con éstos métodos, es que la clasificación de la imagen la da quien ingresa esta descripción, y esta (la clasificación) es dependiente de los criterios propios de alguien. Si luego otro usuario va buscar una imagen con sus criterios (para cada persona una imagen tiene una interpretación diferente) es posible que no la encuentre. Si por el contrario se permite al usuario hacer un bosquejo (un dibujo tan exacto como una imagen de la base de datos, o tan ambiguo que todas las imágenes lo contengan, p.e. una simple línea) de la imagen, el sistema se encarga de buscar las imágenes similares a la dada por el usuario, y así no entran en juego los criterios de clasificación

Sistemas que permiten hacer ambos tipos de consultas ya han sido propuestos. En cuanto al acceso por contenido, un sistema de propósito general es el QBIC System [IBM95], el cual permite realizar accesos por contenido a imágenes y videos. Este permite hacer consultas sobre imágenes de textiles, pero basado en patrones ya preestablecidos. Este hecho limita al usuario el modelo de consultas deseado, ya que no es posible emitir consultas tales como *“recupere las imágenes que tengan una textura t1 en la parte superior izquierda y una textura t2 en la parte inferior derecha”*. Para solucionar el problema de ubicación espacial, existen modelos como el usado por el algoritmo SIM [Gud95] el cual permite consultar imágenes por similitud espacial entre los objetos que componen la imagen, siendo estos objetos íconos que representan cualquier cosa. El sistema se basa en el concepto de imágenes simbólicas, siendo una imagen simbólica el conjunto de íconos. En lo referente a acceso al contenido, el sistema Chabot [OS95] es un sistema de recuperación de imágenes multiresolución. Este modelo usa una base de datos objeto relación (Postgres) para el almacenamiento y manejo de las imágenes y sus datos asociados. Para la recuperación, Chabot integra el uso de texto almacenado y otros tipos de datos con análisis basado en el contenido. Este sistema permite hacer cualquier tipo de

consulta que sea soportada por el modelo de datos sobre el que está hecho. Como las clases tienen atributos que proveen información sobre la imagen, la consulta es dependiente del contenido de los atributos y no de la imagen misma.

La idea entonces es, poder consultar imágenes de textiles (IT) sin basarse en patrones predefinidos usando diferentes criterios: tramado, color, ubicación, rotación, ubicación espacial, escalamiento y forma. El tramado puede ser diagonal, horizontal, vertical o transversal. Esto permite hacer el acceso por contenido a la imagen, ya que se especifica una característica propia de la imagen. Este tramado se hace por regiones de la imagen, haciendo que cada región sea un objeto de la IT. De esta manera se soportan consultas por similitud espacial. El tramado también permite consultar imágenes soportando la rotación de éstas. Para la consulta de imágenes, se usa un lenguaje de consultas visual, el cual permite definir los diferentes criterios que formarán una consulta, usando los operadores lógicos AND, OR y NOT para unir criterios así como paréntesis para agruparlos.

2. ARQUITECTURA DEL SISTEMA

Para llevar a cabo la funcionalidad propuesta, el sistema tiene dos componentes principales: el de poblado de la base de datos (creación del índice que identifica una imagen) y el de consulta a la base de datos. En el primero se extraen las diferentes regiones que conforman la imagen, las cuales vienen identificadas por tener cada una un tramado diferente de las demás. Así mismo se extraen los diferentes colores que están presentes en las regiones de la imagen. Cada una de estas características es entonces almacenada en la base de datos. En el segundo, el usuario genera una consulta desde una interfaz gráfica, la cual contiene un conjunto de herramientas básicas que le permiten generar la consulta: En el segundo, el usuario genera una consulta desde una interfaz gráfica, la cual contiene un conjunto de herramientas que le permiten armar una consulta: crear patrones de texturas con color, ubicar patrones como regiones de una imagen, unir diferentes imágenes para formar expresiones lógicas e importar imágenes. Expresión lógica significa que el usuario puede tener operadores lógicos como AND y OR, los cuales unen operandos que vienen a ser las diferentes imágenes. La consulta dada por el usuario, es entonces, un conjunto de objetos cada uno con sus características propias. A este conjunto se le aplica el mismo proceso que al del poblado: se extraen sus características y se pasan al motor de búsqueda, el cual se encarga de determinar si la consulta (imagen) dada por el usuario si es similar a una o varias imágenes de la base de datos. La figura 1 presenta la arquitectura del sistema.

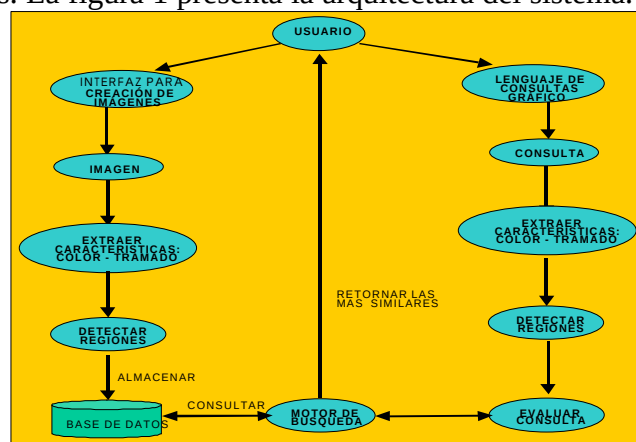


Figura 1 Arquitectura del sistema

2.1. POBLADO DE LA BASE DE DATOS

La creación del índice que sirve como identificador de la imagen se lleva a cabo en varios pasos. Primero se convoluciona la imagen con el filtro de convolución de Laws [Pra91], el cual acentúa el tejido de la imagen, es decir hace mas o menos notorio el tramado. Esto se hace con el fin de descartar partes de la imagen cuyo tramado sea débil y reafirmar partes cuyo tramado sea fuerte. Fuerte y débil se refieren al hecho de si pasan o no el umbral usado en el filtro de convolución. Al tiempo que se hace esto, se hace el cálculo del color usado en la imagen. Los colores que usa una IT son pocos. Este modelo almacena los 16 colores (soportando cambios en la intensidad) mas frecuentes. Dado que una imagen puede tener hasta 262144 colores (2^8), los cuales son combinaciones RGB, el mecanismo de detección de colores se basa en 16 colores base, los cuales tienen cada uno una combinación RGB. El proceso, es como sigue: cuando se encuentra un color en la IT, se revisa si existe en una tabla de colores mas frecuentes de la IT. Si está en la tabla, se incrementa un contador asociado a este, si no está, se ingresa el nuevo color a la tabla.

Después se procede a dividir la IT en máximo 64 regiones, usando un método similar al de los cuatrilos [Cha93]. La diferencia es que solo se baja hasta un tercer nivel en el árbol. (1er. Nivel 4 regiones, 2º. Nivel 16, 3er. Nivel 64). Para determinar si una región es similar a otra, se lleva a cabo una normalización de la región, la cual consiste en calcular el tramado de esta. Este tramado puede ser de cuatro tipos: horizontal, vertical, diagonal y transversal. El proceso toma una región y le hace los cuatro recorridos (para cada tipo de tramado). Para cada recorrido se calculan tres valores, los cuales vienen a ser las características relevantes. Estos recorridos se muestran en la figura 2.

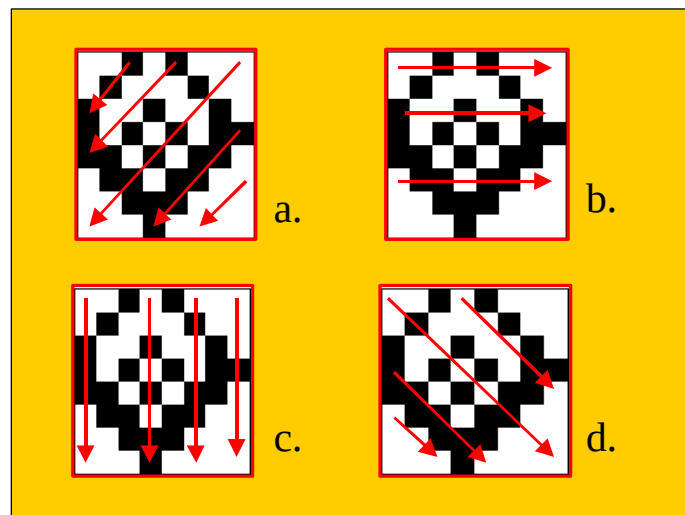


Figura 2. Recorridos de una región.

(a) Diagonal, (b) Horizontal, (c) Vertical, (d) Transversal

Los tres valores, se calculan así: el primer valor se obtiene calculando la cantidad de puntos representativos en cada una de las líneas que conforman el recorrido. Para el ejemplo de la figura 2, los recorridos diagonal y transversal tienen 15 líneas, mientras que los recorridos horizontal y vertical tienen ocho líneas. Como no necesariamente cada línea ejerce suficiente influencia sobre el recorrido, cada una es filtrada por un umbral, y de esta manera, las líneas que pasan este umbral son sumadas, obteniendo así la cantidad de puntos representativos o *suma de participación filtrada del tramado*.

Luego se obtiene el porcentaje de esta suma respecto a la cantidad de puntos de la región, produciendo así *el porcentaje de participación filtrada del tramado*. Este último es el primero de los tres valores que se almacenan como información de la región.

El segundo valor se obtiene en forma similar al anterior, es decir se hacen los cuatro recorridos para la región, pero en vez de calcular la cantidad de puntos que caen en cada línea, lo que se hace es calcular la “sublínea” mas larga dentro de cada línea. Ej. en la figura 2.c, sobre la cuarta línea (la segunda en amarillo de izquierda a derecha, hay tres sublíneas, dos de longitud uno (bajando, en la tercera y en la quinta casilla) y una de longitud dos (séptima y octava casilla). Esta última es la que se elige como sublínea mas larga. Estas sublíneas son entonces sumadas, produciendo así la *suma de densidad filtrada del tramado*. A esta suma se le calcula el porcentaje respecto a la cantidad de puntos de la región, obteniendo así *el porcentaje de densidad NO filtrada del tramado*. El tercer valor se obtiene filtrando las sublíneas y luego sumando las que pasan el umbral. A esta se la calcula el porcentaje respecto a la cantidad de puntos de la región, obteniendo así *el porcentaje de densidad filtrada del tramado*. Un cuarto valor que podría surgir, el *porcentaje de participación NO filtrada del tramado* no se almacena, ya que esta es igual para los cuatro recorridos¹. El objetivo de esta es poder filtrar las diferentes líneas de los recorridos. La figura 3 muestra el cálculo de los tres valores característicos para la textura (región) de la figura 2.

Los tres valores característicos son entonces promediados, para obtener de esta manera el porcentaje total de densidad para cada uno de los recorridos, es decir se obtienen finalmente cuatro porcentajes. Estos valores son la base para unir cuatriles. Los que tengan valores similares, tienen un tramado similar, luego se deben unir. Valores similares quiere decir que los porcentajes de los valores sean bastante cercanos para los cuatro recorridos pero en forma independiente, es decir que sea similar el valor total horizontal de ambas cuatriles, el total vertical, el diagonal y el transversal. En estos casos se usa un umbral con el cual se decide la similitud entre estos. El tramado de la región es el del primer cuadrante que conforma la región, ya que este es el que se usa como punto de comparación con otros cuatriles.

Participación					Densidad				
Diag.	Horiz.	Vert.	Trans.		Diag.	Horiz.	Vert.	Trans.	
3	2	3	4		3	1	3	4	
1	2	3	4		1	1	2	4	
2	3	4	2		1	1	2	1	
3	5	4	3		3	2	2	3	
2	5	4	2		1	2	2	1	
3	4	3	3		3	2	2	3	
2	3	3	2		1	3	3	1	
4	1	1	1		4	1	1	1	
5			4		5		0	4	
25	25	25	25		22	13	17	22	
					88%	52%	68%	88%	
Filtro >= 4	9	14	12	12	9	0	0	12	
% (sobre 25)	36%	56%	48%	48%	36%	0%	0%	48%	
porcentaje de participación filtrada de la direccionalidad porcentaje de densidad no filtrada de la direccionalidad porcentaje de densidad filtrada de la direccionalidad									
25					25				

Figura 3 Cálculo de valores característicos para una región²

¹ Para el ejemplo de la figura 2, siempre será 100% o 25 puntos
² Algunos valores se omiten ya que son siempre cero, como en el caso de la diagonal, que aunque tiene 15 líneas, solamente nueve de ellas tienen un valor diferente de cero.

Como la IT se segmentará en 64 cuatrilos, cada uno de estos tiene asociado implícitamente un número único dentro de la imagen. Al unir los cuatrilos, estos números vienen a ser parte de una región y ayudan a dar una identificación aproximada de la forma de la región y de su ubicación. La figura 4 es un ejemplo de las regiones y su numeración implícita.

1	2	5	6	17	18	21	22
3	4	7	8	19	20	23	24
9	10	13	14	25	26	29	30
11	12	15	16	27	28	31	32
33	34	37	38	49	50	53	54
35	36	39	40	51	52	55	56
41	42	45	46	57	58	61	62
43	44	47	48	59	60	63	64

Figura 4 Regiones unidas con número de posición

En esta imagen hay siete regiones y cada región tiene varios números (que vienen a ser los cuatrilos que la conforman); p.e. la región ubicada en la esquina superior izquierda tiene cinco cuatrilos (1, 2, 3,4 y 9), y tiene como densidad (tramado) la del cuartil identificado con el valor uno (1), por otro lado la que está ubicada en la esquina inferior derecha tiene dos cuatrilos (63 y 64) y tiene como densidad la del cuartil identificado con el valor 63.

Por último se genera una matriz de cruce, de regiones frente a colores, la cual permite asociar una región con varios colores y un color con varias regiones.

2.2. CONSULTAS A LA BASE DE DATOS

Este modulo toma una consulta emitida por el usuario por medio de un lenguaje de consultas (usando una interfaz gráfica que lo implementa) y busca en la base de datos las imágenes que satisfacen la consulta dada. Este proceso de búsqueda de similitud entre una imagen I_Q y cualquier imagen de la base de datos consta de las siguientes fases: primero se lleva a cabo un proceso de normalización de la imagen I_Q , el cual es el mismo que se lleva a cabo en el proceso de almacenamiento de una imagen. Esto produce un número M de regiones para I_Q frente a un número N de regiones de I_{BD} (las cuales fueron calculadas en el momento de almacenar esta imagen). Una vez hecho esto, se procede a comparar cada región de I_Q con cada región de I_{BD} . La manera de llevar a cabo esta comparación es obteniendo la diferencia en valor absoluto entre el componente D (el valor del tramado para el recorrido Diagonal) de la región de I_Q y el componente D de cada una de las regiones de I_{BD} , entre el componente H de la región de I_Q y el componente H de cada una de las regiones de I_{BD} , y así mismo para los otros dos componentes, obteniendo así cuatro valores (un vector 4-valores o 4V) para cada región. Como se comparan regiones frente a regiones, para cada región de I_Q se van a hacer N comparaciones, obteniendo N vectores 4V. De éstos N vectores, se debe escoger el que tenga los

valores mas cercanos a cero³. El vector escogido viene a ser la calificación de r_{iQ} respecto a las N regiones de I_{BD} .

En este punto se tienen M vectores $4V$, cada uno de los cuales indica la mejor calificación para cada una de las M regiones de I_Q . Como cada región de I_Q aporta (contiene) una cantidad diferente de pixeles al total de la imagen I_Q (un porcentaje de acuerdo al total de pixeles) cada valor de los vectores $4V$ se multiplica por este porcentaje, para así producir el vector $4V$ de calificación definitiva de cada uno de los r_{iQ} .

La primera parte del módulo de consultas, la cual es **independiente** de la consulta solicitada por el usuario, termina aquí, dejando los elementos necesarios para que la segunda parte del módulo, la cual es **dependiente** de la consulta, determine si se puede o no satisfacer la consulta. Para explicar la segunda parte del módulo, es necesario conocer el lenguaje de consultas, el cual genera una expresión lógica. Dependiendo del tipo de consulta, se llevan a cabo diferentes operaciones, las cuales finalmente permiten determinar la similitud entre dos imágenes.

3. LENGUAJE DE CONSULTAS

La definición de este lenguaje está basada en el Formalismo Extendido de BackusNaur (EBNF por sus siglas en inglés), el cual indica que:

- Corchetes '[' y ']' denotan opcionalidad de la expresión encerrada entre éstos.
- Llaves '{' y '}' denotan repetición (0 o más veces).

Los símbolos no terminales comienzan con una letra en mayúscula (ej. **Criterio**). Los símbolos terminales están escritos en minúscula (ej. **fijo**) o están denotados como cadenas (ej. **"H"**).

```
Query      ::= SELECT  Imagenes
              FROM    TiposImagenes
              WHERE    Imagen.Contenido LIKE Criterios

Criterios  ::= [not] Termino { or [not] Termino }
Termino    ::= Ubicación Rotación Patrón { and Ubicación Rotación Patrón }
Patrón     ::= Tramado | Color | Forma | Nada
Ubicación  ::= si | no
Rotación   ::= si | no
Tramado    ::= "H" | "V" | "D" | "T"
Color      ::= rgb (rojo, verde, azul)
Forma      ::= poligono
Nada       ::= Δ
```

³ Este valor proviene de una diferencia. Si esta diferencia es cercana a cero, es porque ambos valores son muy similares, y en consecuencia las regiones tienen un tramado similar.

Nota: Patrón se refiere a las características de una región. Sin embargo, una región puede llegar a ser una imagen completa. El símbolo *not* se refiere a que la imagen no cumpla los criterios dados.

El lenguaje permite construir expresiones lógicas tales como:

SELECT Imagenes WHERE ImagenContenido LIKE Rojo OR (H AND D),

que significa: “Traiga las imágenes que tengan una región con el color Rojo, o las imágenes que tengan dos regiones, una con tramado horizontal H y otra con tramado vertical V”.

Con base en el lenguaje se puede dar paso a la segunda parte del módulo de consultas de imágenes, el cual se basa en la consulta dada.

4. EVALUACIÓN DE CONSULTAS

Los diferentes criterios que componen el lenguaje son evaluados de la siguiente forma:

Color: este criterio indica que se requiere una o unas imágenes que tengan un color similar al dado por el usuario. Para este tipo la búsqueda se basa en la matriz de cruce de cuatrilos frente a colores.

Tipo: este criterio indica el tipo o los tipos de tejidos que tiene una región. La búsqueda se hace con base en el esquema de participación de cada una de las direcciones (H, V, T, D). Si el usuario escoge rotación (se tienen M vectores 4V), los cuatro valores de cada 4V se suman, es decir se aplica el producto punto $1 \bullet (H, V, D, T)$, generando así un valor escalar que da la calificación definitiva para cada región, es decir se tienen M valores escalares. Si el usuario escoge no-rotación, se mantienen los M vectores 4V. Para el primer caso se evalúa si alguna región de I_Q cumple el mismo tramado de alguna región de I_{BD} , para el segundo se busca el vector 4V de I_Q que más se asemeje a un vector 4V de I_{BD} .

Forma: este criterio indica que se debe tener en cuenta la forma que tiene la región dada por el usuario, la cual tiene Color y/o Tipo de tejido. Para cada región se genera su vector característico. Este vector es tomado con base en el vector de características del algoritmo SOM de [HM96], el cual es independiente de la escala, traslación y rotación de un objeto.

Ubicación: este criterio tiene o no en cuenta las posiciones de las regiones ya sean de tejido o de color dadas por el usuario. Si el usuario indica como criterio la ubicación, se arma un grafo de conectividad de las regiones de la I_Q , donde cada vértice contiene el tipo de tramado y/o los colores y/o la forma de la región. A la imagen I_{BD} se le genera un grafo similar, el cual luego es comparado con el de I_Q , para así determinar si las dos imágenes tienen regiones con ubicación espacial.

Nula: cualquier imagen satisface esta condición. Sin embargo el objetivo de esta cláusula es permitir consultas como:

SELECT Imagenes FROM ImagenesGeneradas WHERE ImagenContenido LIKE (Ubicacion (x, y) AND Rojo),

Que significa, “Traiga las imágenes de la base de datos de Imágenes generadas donde la mitad superior sea de color Rojo y el resto de cualquier manera”, asumiendo que Ubicacion (x, y) especifica la mitad superior de la imagen.

El manejo de los operadores es como sigue: el operador *and* es implícito en las diferentes características que el usuario coloca en la imagen de la consulta, es decir, no es necesario unirlos con un *and*; el sistema automáticamente lo hace. Para el manejo del operador *or*, el usuario puede generar varias subconsultas (o imágenes), cada una con sus características propias (unidas por un *and*); en este caso, todas las subconsultas son unidas con un *or*. El manejo del *not* es similar, salvo que el usuario prefija la subconsulta con este operador.

En general, el símbolo terminal *Termino* viene a ser una imagen de consulta y el sistema une éstos símbolos con un *OR*; el símbolo terminal *Patron* viene a ser una característica de cada imagen de consulta, y son unidos por un *AND*.

El modelo de consultas puede soportar el cambio de tamaño (escalamiento) de imágenes, así como el traslape entre regiones. El primero gracias a que el cálculo de densidad de tramado es hecho con base en el número de píxeles de la región y no de la imagen. El segundo tiene un límite (el 40% oculto), el cual viene dado por un umbral en el momento de evaluar los valores de ambas regiones. Si se pasa el umbral, es porque ambas regiones son similares, ya que la parte no traslapada alcanza a tener una calificación similar al de una región no traslapada.

5. COMPLEJIDAD DE LOS ALGORITMOS

El modelo del sistema tiene como base dos algoritmos: el de almacenamiento, el cual almacena la información descrita en el poblado de la base de datos, y el de consulta, el cual lleva a cabo la comparación entre dos imágenes. Esta sección presenta la complejidad y el espacio usado por ambos algoritmos.

5.1. ALMACENAMIENTO

Este proceso inicia aplicando un filtro de convolución con ventana de 3x3 sobre la imagen, cada tres píxeles.

```
for (Fil = 3 ; Fil < ImagenActual -> ALTO ; Fil+=3) //  $O(H/3) * O(A/3) = O(HA/9)$ 
    for (Col = 3 ; Col < ImagenActual -> ANCHO ; Col+=3) //  $O(A/3)$ 
    {
        Valor = FiltroConvolucion (Col, Fil) ;           //  $O(1)$ 
        ActualiceTexturaConv (Col, Fil, Valor) ;         //  $O(1)$ 
    }
```

El filtro de convolución aplicado en un punto tiene complejidad $O(1)$, ya que es simplemente aplicarlo en nueve puntos.

El ciclo mas interno (el que recorre las columnas) se ejecuta $ANCHO/3$ veces, es decir es $O(A/3)$, siendo A el mismo ANCHO. El ciclo mas externo (el que recorre las filas) se ejecuta $ALTO/3$ veces, es decir $O(H/3)$, siendo H el mismo ALTO.

Luego se procede a calcular la densidad de tramado para los 64 cuatrilos.

```

Inc = ImagenActual -> ANCHO/8 ;
for (Fil = 0 ; Fil < ImagenActual -> ALTO ; Fil += Inc)    //  $O(8) * O(AH/8) = O(AH)$ 
    //  $O(8) * O(AH/64) = O(AH/8)$ 
    for (Col = 0 ; Col < ImagenActual -> ANCHO ; Col += Inc)
    {
        ArmarRegiones (Col, Fil, Col+Inc, Fil+Inc) ;        //  $O(AH/64)$ 
        AcumuleColor      (Col, Fil) ;                      //  $O(1)$ 
    }

```

Para calcular la densidad de los cuatrilos, se llevan a cabo los cuatro recorridos (H, V, T y D)

```

PartVertical      ( ) ;    //  $O(AH/64)$ 
PartHorizontal    ( ) ;    //  $O(AH/64)$ 
PartDiagonal      ( ) ;    //  $O(AH/64)$ 
PartTransversal   ( ) ;    //  $O(AH/64)$ 

```

Cada uno de estos recorre una matriz que mide $A/8 \times H/8$, lo que produce el valor de $O(AH/64)$. La toma del color simplemente incrementa un valor en la tabla de colores de un cuartil.

Esto produce tres bloques principales en el algoritmo:

1. Filtro de convolución: $O(AH/9)$
2. Cálculo de densidades de regiones: $O(AH)$

lo cual da una complejidad total del max ($O(AH/9)$, $O(AH)$) = $O(AH)$

5.2. CONSULTA

Para poder comparar dos imágenes, se inicia aplicando el algoritmo de almacenamiento a la imagen de la consulta, dando una complejidad de $O(AH)$.

Luego se hace la comparación entre las M regiones de I_Q y las N de regiones de I_{BD} .

```

for (Rq = 1 ; Rq <= R_Con ; Rq++) // R_Con = M        //  $O(1) = O(64)$ 
{
    for (Rbd = 1 ; Rbd <= R_BD ; Rbd++) // R_BD = N //  $O(64)$ 
    {
        Dif [0] = fabs (PartBD [Rq] [0] - PartQ [Rbd] [0]);
        Dif [1] = fabs (PartBD [Rq] [1] - PartQ [Rbd] [1]);
        Dif [2] = fabs (PartBD [Rq] [2] - PartQ [Rbd] [2]);
        Dif [3] = fabs (PartBD [Rq] [3] - PartQ [Rbd] [3]);
    }
    CalRq [Rq-1] = Menor (Dif, ImagenQuery -> TotRegiones) ; //  $O(1)$ 
    CalRq [Rq-1] *= Participacion (Rq) ;
}

```

Menor devuelve el vector 4V de regiones de la BD que mas se parece a una región de la imagen de consulta. Como una imagen tiene máximo 64 regiones, $1 \leq M \leq 64$, $1 \leq N \leq 64$, y $1 \leq \text{TotRegiones} \leq 64$. *Participacion* retorna el porcentaje de participación de la región Rq en toda la imagen de consulta.

En este momento termina la parte independiente de la consulta.

La parte dependiente varía en complejidad (el símbolo $\|$ significa cardinalidad):

- Las que implican ubicación, se basan en el algoritmo SIM, [GR95], el cual tiene una complejidad de $O(\|E^q\| + E\|^{bd}\|)$, siendo $\|E^q\|$ y $E\|^{bd}\|$ la cantidad de vértices del grafo de la imagen de la consulta y de la base de datos respectivamente.
- Rotación tramado: se promedian máximo 64 regiones, lo cual da una complejidad de $O(1)$
- Tramado: como es región a región, en el peor de los casos se deben comparar todas las regiones de I_Q con las regiones de I_{BD} , dando así una complejidad de $O(1)$.
- Color: se busca color por color en la matriz de cruce de regiones y colores, (y como hay un número máximo de colores para todas las imagenes), esta búsqueda en el peor de las casos tiene una complejidad de $O(1)$
- Forma: se basa en la fase de construcción de un vector de caracterización de imágenes, del algoritmo SOM [HM96]. Esta fase consiste en calcular la redondez, rectangularidad, elipticidad, excentricidad e irregularidad de una región. Para el caso de las texturas, las regiones que se manejan no son irregulares (son un conjunto de cuadrantes), por lo tanto solo se calculan las cuatro primeras características. El cálculo de estas es una fórmula matemática sencilla, la cual tiene una complejidad de $O(1)$.

De esta manera, la parte dependiente tiene una complejidad de $\max(O(\|E^q\| + E\|^{bd}\|), O(1)) = O(\|E^q\| + E\|^{bd}\|)$. Sin embargo $\|E^q\|$ es máximo 64, y $E\|^{bd}\|$ también, lo que produce una constante final de $O(1)$.

Finalmente, se tienen dos bloques en el algoritmo de consulta:

1. Comparación entre regiones: $O(AH)$
2. Evaluación de la consulta: $O(1)$

lo cual da una complejidad total del $\max(O(AH), O(1)) = O(AH)$.

CONCLUSIONES

Diferentes modelos y prototipos para el acceso por contenido a imágenes existen hoy en día. Así mismo existen variadas técnicas para la manipulación de éstas. Sin embargo la gran mayoría de éstos modelos adolecen de problemas comunes: se basan en imágenes que sirven como patrones y son alimentadas inicialmente en el sistema, o se basan en imágenes que el usuario importa, el sistema las clasifica y luego las consulta. Aunque también existen sistemas que no se basan en imágenes “iniciales” y permiten al usuario crear una imagen de consulta, no existe un modelo que permita crear una imagen de consulta de texturas con un tramado definido por el usuario y con especificación de regiones. El sistema QBIC de IBM es bastante amplio en el manejo de consultas por diferentes criterios. Sin embargo para el manejo de texturas se basa en imágenes predefinidas. Además no permite armar expresiones lógicas usando operadores tipo OR y NOT.

El modelo aquí presentado permite hacer accesos por contenido a imágenes de textiles, sin basarse en patrones previamente definidos. De esta manera el usuario puede enviar diferentes tipos de consultas a la base de datos, las cuales solo son conocidas en el momento de analizar la consulta. Estas consultas son creadas por medio de una interfaz gráfica, la cual provee al usuario con las herramientas necesarias o para armar un patrón de diseño que le genere una textura completa o para cargar imágenes con formatos especiales. Además el usuario puede armar expresiones lógicas compuestas, dándole así una capacidad mas poderosa de consulta. Las consultas no solamente están limitadas a especificar patrones o colores; el usuario puede especificar la forma que tiene una región de la imagen, o la ubicación de una o mas regiones dentro de la imagen de acuerdo a la ubicación espacial de éstas.

Los algoritmos que soportan el modelo, el de almacenamiento de una imagen y el de comparación de dos imágenes, tienen una complejidad de ejecución dada en términos del ancho y del alto de la imagen y el espacio ocupado por el índice generado por la imagen es mínimo comparado con el tamaño de la imagen. Cabe anotar, que el índice generado puede no ser el óptimo para imágenes con una cantidad muy grande de tramados diferentes y con un tamaño en bytes pequeño (menos de 5K). Sin embargo es de tener en cuenta que una imagen de un textil no tiene muchos cambios en un área pequeña como viene a ser la pantalla sobre la cual observa el usuario. Aunque el modelo soporta hasta 64 regiones con tramados diferentes, es poco probable llegar a ese límite; por el contrario para el promedio de las imágenes, el tamaño del índice está entre 300 y 800 bytes el cual es un valor bastante bajo comparado con el tamaño de la imagen de un textil. Debido a esto, se puede llevar a cabo una rápida comparación de dos imágenes. Este modelo pretende además, presentar una técnica en el reconocimiento de texturas por un método no supervisado usando características propias de un tejido.

REFERENCIAS

[IBM95] IBM ALMADEN RESEARCH CENTER. Query by Image and Video Content: The QBIC System. IEEE Computer Society Vol 28 Núm 9, págs. 23-32, Septiembre 1995

[Gud95] Venkat N. Gudivada. Multimedia Systems – An Interdisciplinary Perspective. ACM Computing Surveys. Vol 27, No. 4, págs. 201-256, Diciembre 1995

[OS95] Virginia Ogle y Michael Stonebraker. Chabot: Retrieval from a Relational Database of Images. IEEE Computer Society Vol 28 Núm 9, págs. 40-48, Septiembre 1995

[Pra91] William K. Pratt. Image Feature Extraction. Digital Image Processing. Wiley-Interscience, New York, 1991, págs. 557-595

[Cha93] A. Chapman, Image Digital Proccesing. John Willey & Sons. 1993

[HM96] Kyung-Ah Han Y Sung-Hyun Myaeng. Image Organization and Retrieval with Automatically Constructed Feature Vectors. 19th ACM SIGIR, Conference on Information Retrieval 1996, págs. 157-165

[GR95] Venkat Gudivada y Vijay Raghavan. Design and Evaluation of Algorithms for Image Retrieval by Spatial Similirity, acm TOIS, Vol 13 No. 2, págs. 115-144, Abril 1995