

MIS: Un prototipo de un sistema de manipulación de imágenes similares*

Geneviève Jomier¹, Maude Manouvrier¹, Marta Rukoz², Juan Ramirez², Yexica Valero²

¹ Laboratoire LAMSADE, Université Paris-Dauphine-France
Place de M^{al} De Lattre de Tassigny
75775 Paris Cedex 16, Francia
e-mail: {jomier, manouvri}@lamsade.dauphine.fr

² Centro de Computación Paralela y Distribuida
Escuela de Computación
Facultad de Ciencias
Universidad Central de Venezuela
Apartado 47002 Los Chaguaramos, Caracas Venezuela
e-mail : mrukoz@flynn.ciens.ucv.ve

Resumen

Diferentes sistemas de tratamiento de imágenes, como los sistemas de tratamiento de imágenes médicas, geológicas, artísticas o educativas requieren la conservación de imágenes que representen los diferentes cambios realizados en el tratamiento aplicado. Estas imágenes pueden ser consideradas como similares ya que normalmente existe poca diferencia entre ellas, con respecto a su representación física. La representación de imágenes similares ha sido el objeto de diversos trabajos de investigación. Esos trabajos utilizan mecanismos de solapamiento de estructuras de datos o estructuras genéricas para representarlas, utilizando así menor cantidad de espacio que si esas imágenes se almacenaran completamente. En este trabajo presentamos un prototipo de un sistema de almacenamiento y manipulación de imágenes que permite el almacenamiento eficiente y la aplicación de operaciones sobre imágenes similares. Utilizamos el enfoque de árboles cuaternarios genéricos, el cual permite además de un almacenamiento eficiente de imágenes similares, la aplicación de operaciones sobre las diferentes imágenes.

Palabras Claves: Bases de datos de imágenes, árboles cuaternarios (Quad-Trees), representación de imágenes, optimización de almacenamiento, prototipos.

* Este trabajo fue financiado parcialmente por el CNRS en Francia, y por el CONICIT y CDCH-UCV en Venezuela

1.- Introducción

En el tratamiento de imágenes es necesario que el usuario disponga de un sistema que le permita aplicar una serie de operaciones particulares a una imagen o a partes de una imagen, así como almacenar, acceder y manipular eficientemente las imágenes resultantes de esos tratamientos [1, 10, 15, 19, 26]. La conservación de esas diferentes imágenes permite al usuario regresar a imágenes precedentes cuando una operación ha modificado mucho la imagen original o aplicar operaciones alternativas a partir de una imagen inicial o ya tratada. Ello permite igualmente seguir la evolución de una imagen o de una zona de una imagen después de diferentes tratamientos. En cualquiera de esos casos, esas imágenes pueden ser consideradas como similares. En efecto, ellas poseen generalmente un gran número de píxeles u objetos en común. Una imagen puede ocupar desde algunos KBytes hasta muchos Gigabytes. De allí la necesidad de crear un sistema eficiente de almacenamiento de imágenes que permita a la vez manipular, interrogar o modificar las imágenes. Existen diversos sistemas que tratan con sistemas de bases de datos de imágenes. En [6, 9, 15, 16, 17], se almacenan las imágenes y meta datos sobre las mismas para su recuperación a través de esa información. En esos enfoques no se interesan ni en la optimización del almacenamiento ni en las posibles modificaciones de las imágenes. Otros sistemas como los presentados en [5, 10, 13, 24, 28] se interesan principalmente en el almacenamiento y no toman en cuenta las operaciones sobre las imágenes.

Los árboles cuaternarios son una estructura de datos muy utilizadas en la representación de imágenes digitales. Un árbol cuaternario es una estructura que permite la partición del espacio en zonas homogéneas [20, 21, 22]. El almacenamiento de secuencias de imágenes similares representadas como árboles cuaternarios ha sido el objeto de diversos trabajos. En [12, 24, 28, 29] se adaptan o extienden los mecanismos de solapamiento de estructuras de datos [2, 3, 4, 14, 18] para representarlas. La idea es representar cada imagen por un árbol cuaternario y de compartir las partes comunes de esos árboles entre las diferentes imágenes similares. Esto permite minimizar el espacio de almacenamiento, con un tiempo de acceso igual al requerido si no existiera ningún solapamiento entre los diferentes árboles cuaternarios, pero para aplicar operaciones sobre cualquiera de esas imágenes es necesario recuperar cada una de las imágenes de la secuencia. Otro enfoque presentado en [8, 25] utiliza una estructura basada en un árbol cuaternario genérico (GQT **Generic Quad-Tree**) y un árbol de imágenes. Esta estructura trabaja sobre imágenes organizadas en árboles cuaternarios y permite realizar eficientemente operaciones sobre imágenes similares manteniendo a la vez un almacenamiento eficaz de las mismas.

El objetivo de este artículo es presentar MIS, un prototipo de un sistema **Manejador de Imágenes Similares**, a fin de validar nuestro enfoque de GQT presentado en [8]. MIS permite el almacenamiento y la aplicación de operaciones sobre imágenes similares. Trabajamos sobre imágenes organizadas en árboles cuaternarios. Consideramos que dos imágenes son similares en la medida de que sus árboles cuaternarios correspondientes sean similares. Esto es, dos imágenes son similares si los árboles cuaternarios que las representan tienen gran cantidad de nodos iguales. El documento está organizado de la manera siguiente. Una primera sección presenta el uso de árboles cuaternarios y el enfoque GQT para la representación de imágenes similares. La sección 3, define las funcionalidades de MIS. La sección 4, presenta el ambiente de ejecución y una evaluación, con respecto al uso de memoria, del enfoque GQT. Por último la sección 5 da nuestras conclusiones y perspectivas de investigación.

2.- Uso de Árboles Cuaternarios y el enfoque GQT para la representación de imágenes similares

El árbol cuaternario es una estructura de almacenamiento de imágenes eficaz que permite representar las imágenes en dos dimensiones. Se trata de una estructura jerárquica construida por división recursiva del espacio en cuatro cuadrantes [20, 21, 27]. Existen árboles cuaternarios sobre diversos tipos de datos, el más conocido es el árbol cuaternario sobre las regiones [7], que permite dividir una imagen en zonas (o regiones) homogéneas, o zonas que satisfacen las mismas características: mismo color o textura, por ejemplo. La imagen es dividida recursivamente en cuatro cuadrantes tal que cada cuadrante es representado por un nodo del árbol cuaternario. La raíz representa la imagen completa. Esta posee cuatro nodos hijos que representan sus cuatro cuadrantes. Cada nodo hijo a su vez puede poseer cuatro hijos si el cuadrante correspondiente es dividido en cuatro y así sucesivamente. Los nodos hojas del árbol cuaternario no están todos en el mismo nivel. Ellos representan las zonas homogéneas de la imagen, de acuerdo a un cierto criterio que puede ser el color de la región, por ejemplo. Los diferentes cuadrantes de una imagen son enumerados de manera tal que existe una relación que permite a partir de un nodo de un árbol cuaternario localizar el cuadrante correspondiente en la imagen y viceversa.

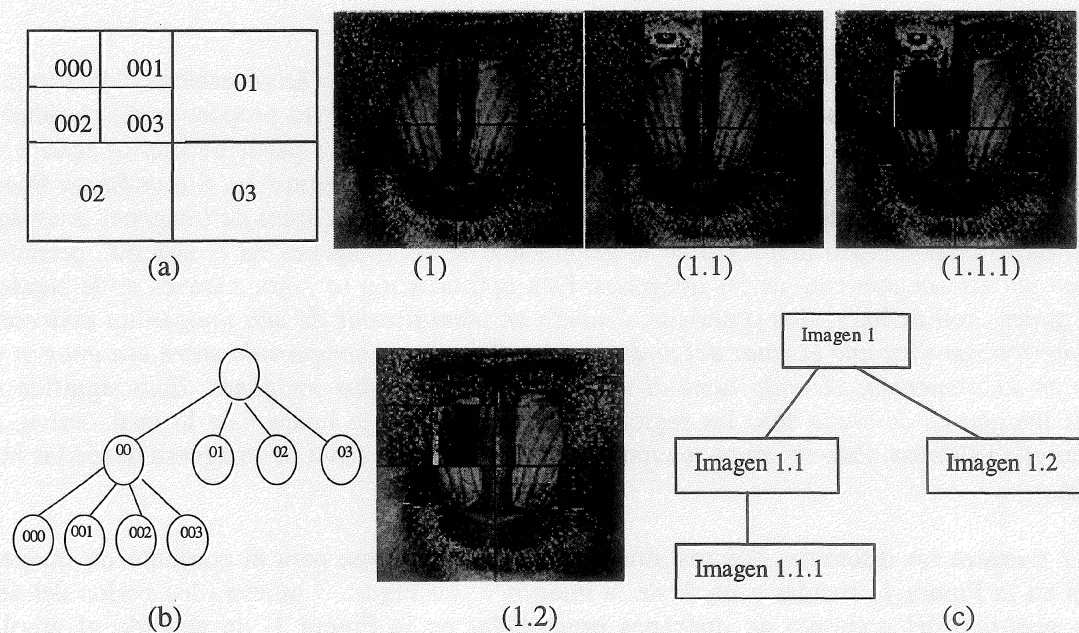


Figura 1: Conjunto de imágenes similares

En la Figura 1, se presenta un conjunto de imágenes similares. La imagen (1) es la imagen original donde se realizó una división de acuerdo a ciertos criterios del usuario. La imagen (1.1) presenta una modificación de la imagen original (1); el nodo 00 fue dividido y el nodo 001 fue modificado. Los valores de grises de este nodo fueron invertidos con respecto a la imagen (1). La imagen (1.1.1), es una modificación de la imagen (1.1) donde los valores del nodo 003 fueron invertidos. Por último la imagen (1.2) es una modificación de la imagen (1) donde se invirtieron los valores del nodo 003. La imagen de

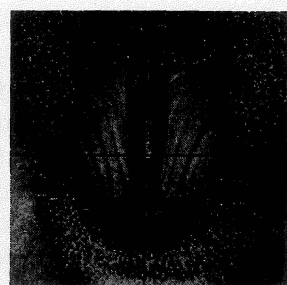
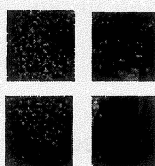
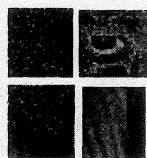
la izquierda (a), muestra la numeración de las diferentes regiones de una imagen cuando ésta es dividida para su representación en árbol cuaternario. Esta numeración es proyectada en la representación del árbol cuaternario correspondiente como lo muestra la Figura 1 (b).

Para el almacenamiento y control del conjunto de imágenes similares utilizaremos el enfoque propuesto en [8], árbol cuaternario genérico (GQT). En este enfoque cada imagen del conjunto es identificada y descompuesta en un árbol cuaternario que representa un estado de un árbol cuaternario genérico.

El árbol cuaternario genérico de un conjunto de imágenes contiene tantos nodos como el número máximo de nodos de cada árbol cuaternario que represente una imagen del conjunto. Cada nodo del árbol cuaternario genérico, o nodo genérico, es identificado según la numeración presentada en la Figura 1 (b) y tiene un valor para cada imagen del conjunto. Para recuperar el valor de un mismo nodo en diferentes imágenes, cada nodo genérico del árbol cuaternario genérico asocia la imagen con el valor del nodo en esa imagen. Cuando un nodo no existe en el árbol cuaternario de una imagen particular, su valor en el nodo genérico correspondiente, es $\perp$. Cuando un nodo es hoja en un árbol cuaternario particular su valor puede ser negro, blanco o gris. Si éste es interno, es decir si fue dividido su valor es $\perp$.

El conjunto de imágenes es registrado en un árbol de imágenes. Este árbol establece la forma de derivación de las imágenes, esto es, el orden como fueron obtenidas. Para propósitos de claridad las imágenes son identificadas de manera tal que las imágenes obtenidas a partir de una imagen x son identificadas como $x.1$, $x.2$,..., etc., tal como lo muestra el árbol de la Figura 1c. Sin embargo pueden utilizarse otros enfoques de enumeración como el presentado en [11]. El árbol de imágenes además de ayudar al usuario manteniendo una traza de la generación de las imágenes en el sistema, permite la optimización del almacenamiento de las imágenes. Esta optimización se logra a través de la siguiente regla en la generación de los nodos genéricos: *Cuando un identificador de una imagen no aparece en un nodo genérico significa que el valor del nodo esta implícitamente compartido entre esa imagen y la precedente en la secuencia, el nodo tiene el mismo valor en las dos imágenes.* Esto significa que cuando una imagen es derivada sólo las regiones modificadas, de la imagen de la cual deriva, son almacenadas en el sistema. Esto se aplica recursivamente a través del árbol de imágenes desde las hojas hasta la raíz.

La Figura 2 muestra los diferentes valores almacenados en el sistema para el conjunto de imágenes presentadas en la Figura 1, creadas a partir de la imagen 1. La Figura 3 muestra los nodos del árbol cuaternario genérico del conjunto de imágenes presentadas en la Figura 1, de acuerdo al árbol de imágenes presentado en la Figura 1c. Los nodos genéricos 02 y 03 tienen la misma estructura que el nodo genérico 01. De la misma manera los nodos 001 y 002 tienen la misma estructura que el nodo genérico 000.

Valores de los nodos
para la imagen 1Valores de los nodos
para la imagen 1.2Valores de los
nodos para la
imagen 1.1Valores de los
nodos para la
imagen 1.1.1**Figura 2: Valores almacenados de cada región**

Valor nodo o Dir. del valor del nodo en la imagen	Identificador de imagen
Dir. valor del nodo en la Imagen 1	1
1	1.1, 1.2

NODO GENERICO 00

Valor nodo o Dir. al valor del Nodo en la imagen	Identificador de imagen
1	1
Dir. valor del nodo para la imagen 1.1	1.1
Dir. valor del nodo para la imagen 1.1.1	1.1.1
Dir. valor del nodo para la imagen 1.2	1.2

NODO GENERICO 003

Valor nodo o Dir. al valor del nodo en la imagen	Identificador de imagen
1	1
Dir. valor del nodo en la Imagen 1.1	1.1
Dir. valor del nodo en la Imagen 1.2	1.2

NODOS GENERICO 000,001 Y 002

Dir. valor del Nodo para Imagen	Identificador de imagen
Dir. valor del nodo para la imagen 1	1

NODOS GENERICO 01, 02 Y 03

Figura 3: Nodos Genéricos del conjunto de imágenes

3.- Funcionalidad de MIS

MIS es un sistema de tratamiento de imágenes que permite el almacenamiento eficiente y la manipulación de imágenes o partes de una imagen. En MIS es posible cargar una imagen, aplicarle operaciones de tratamiento de imágenes (como detección de bordes, complemento, segmentación, etc.) y almacenar la imagen resultante como derivada de la primera. También es posible dividir una imagen, modificar determinadas regiones de esa imagen y guardar sólo esas modificaciones para luego recuperar tanto la imagen original como la imagen resultante de las modificaciones. MIS permite igualmente aplicar operaciones sobre dos imágenes como unión, intersección y comparación, y guardar la imagen resultante como derivación de alguna de las imágenes implicadas en la operación. MIS está basado en la representación de imágenes similares basada en árboles cuaternarios genéricos, descritos en la sección 2. Esto implica que cuando se modifican regiones de una imagen para la creación de una imagen derivada, sólo las partes modificadas son almacenadas. Para ello se utilizan instancias de cada nodo o región del árbol cuaternario genérico y el árbol de derivación de las imágenes. MIS dispone de las siguientes operaciones:

- **Búsqueda o recuperación de una imagen:** Esta operación permite recuperar cualquier imagen independientemente de su posición en el árbol de imágenes.
- **Integración y división de regiones:** Esta operación permite dividir o integrar la imagen en cuatro regiones de igual tamaño. De manera recursiva cada región puede ser dividida a su vez en cuatro regiones.
- **Operación de eliminación:** Esta operación permite eliminar del sistema una imagen
- **Operaciones de modificación de una imagen:** Estas operaciones permiten aplicar cualquier operación de tratamiento de imágenes sobre cualquier imagen. En el prototipo presentado, aunque es posible añadir cualquier otra operación, sólo se implementaron las operaciones:
 - **Complemento,** la cual produce el valor complemento de cada pixel de la imagen, por ejemplo si es negro lo cambia a blanco.
 - **Unión de dos imágenes:** cada pixel de la imagen resultante es la suma de los pixeles correspondientes de las imágenes de entrada.
 - **Intersección de dos imágenes:** el valor de cada pixel de la imagen resultante es igual al de los pixeles correspondiente de las imágenes de entrada si estos son iguales, o al color de fondo de la imagen si éstos son diferentes.
- **Operaciones de modificación de una región:** Estas operaciones permiten aplicar cualquier operación de tratamiento de imágenes sobre una región de cualquier imagen. Para ello es necesario que la imagen haya sido dividida, y seleccionar la región correspondiente. En el prototipo presentado sólo se implementó la operación complemento, pero es posible añadir cualquier otra operación sobre una región.

4.- Ambiente de Ejecución y Evaluación de MIS

Un ambiente para manipulación de imágenes fue elaborado para MIS. En este ambiente el usuario puede trabajar con dos imágenes simultáneamente, las cuales son visualizadas en todo momento. Para cada una de esas imágenes, a través de un menú, el usuario puede realizar las diferentes operaciones definidas en MIS. La Figura 4, muestra los diferentes menús que pueden ser utilizados en cada ventana conteniendo una imagen.

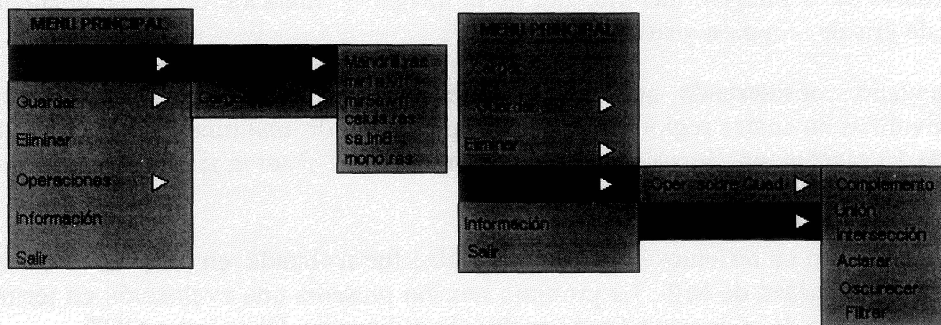


Figura 4: Diferentes Menús de MIS

La Figura 5, muestra una aplicación sobre MIS, donde se realizó la unión de dos imágenes dejando el resultado en la imagen de la derecha.

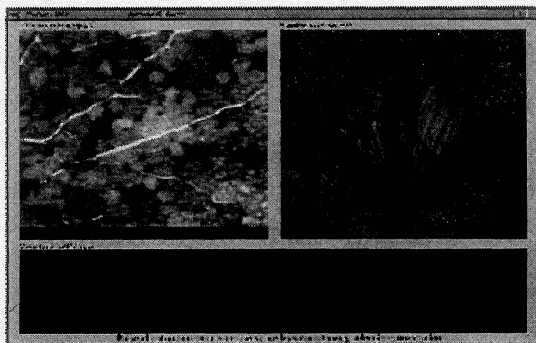


Figure 5: Una aplicación sobre MIS. Unión de dos imágenes

Implementación de MIS

MIS fue implementado en una Silicon Graphics, utilizamos el lenguaje C y las librerías OpenGL y GLUT para el desarrollo de la interfaz. Para cada una de las regiones almacenadas en el sistema se guardan sus diferentes valores de gris así como un encabezado que contiene la información del mapa de colores de la imagen. Cada nodo genérico contiene información sobre el tamaño de la región, las coordenadas (X,Y) del punto superior izquierdo de la región, nivel de profundidad donde se encuentra en el árbol cuaternario de la imagen, identificador de la imagen y dirección donde se encuentran los valores de niveles de gris de la región y su encabezado.

MIS fue implementado considerando que las imágenes son de tamaño $2^n \times 2^n$, las imágenes inicialmente son divididas en cuatro regiones, un usuario puede dividir una imagen hasta que la región tenga un tamaño de 16 pixeles, una imagen puede ser eliminada del sistema si ésta no tiene imágenes derivadas de ella.

Una evaluación del prototipo en términos de su funcionalidad fue realizada en [25]. Las Figuras 1 y 5 son ejemplos de la funcionalidad de MIS. La próxima sección presenta una evaluación en términos de su eficiencia en cuanto al uso de memoria a fin de evaluar la utilización del enfoque GQT.

Evaluación de MIS

Para evaluar la cantidad de memoria que utiliza MIS para almacenar un conjunto de imágenes debemos considerar lo siguiente: cada nodo genérico ocupa por lo menos 48 bytes, el encabezado de cada región ocupa aproximadamente 816 bytes.

Para evaluar el uso de memoria utilizado por MIS, se generó un conjunto de 20 imágenes de la siguiente manera: se tomó una imagen de dimensión 512×512 y se almacenó en el sistema, luego se dividió en cuatro y se modificaron las sub-regiones y se almacenó la nueva imagen. A esta nueva imagen se le dividió en cuatro su primer cuadrante, se modificaron y se almacenó la imagen resultante. Este proceso se repitió hasta obtener imágenes con 24 particiones. El proceso se realizó también de manera inversa, es decir se tomó una imagen con 24 particiones y se fue agrupando su información hasta obtener una imagen sin particiones. La Figura 6a muestra la cantidad de espacio utilizado por MIS para el almacenamiento de ese conjunto de imágenes y la cantidad requerida si se almacenaran todas las imágenes tratadas de manera independiente. En la Figura 6b, muestra la progresión de cantidad de memoria requerida por MIS para cada uno de los casos planteados. Podemos ver el incremento de memoria utilizado por MIS al ir insertando el conjunto de imágenes partiendo de imágenes con sólo cuatro divisiones y haciendo divisiones progresivas, y partiendo de imágenes completamente divididas y haciendo agrupaciones progresivas. Como puede verse en la Figura 6b, los dos procesos tienden a ocupar la misma cantidad de memoria para almacenar las imágenes. Esta cantidad de memoria, requerida para almacenar las imágenes, depende de la cantidad de cuadrantes modificados en cada imagen. A medida que se generan más particiones, no existentes en las imágenes almacenadas, es necesario crear nuevos nodos genéricos en el árbol cuaternario genérico.

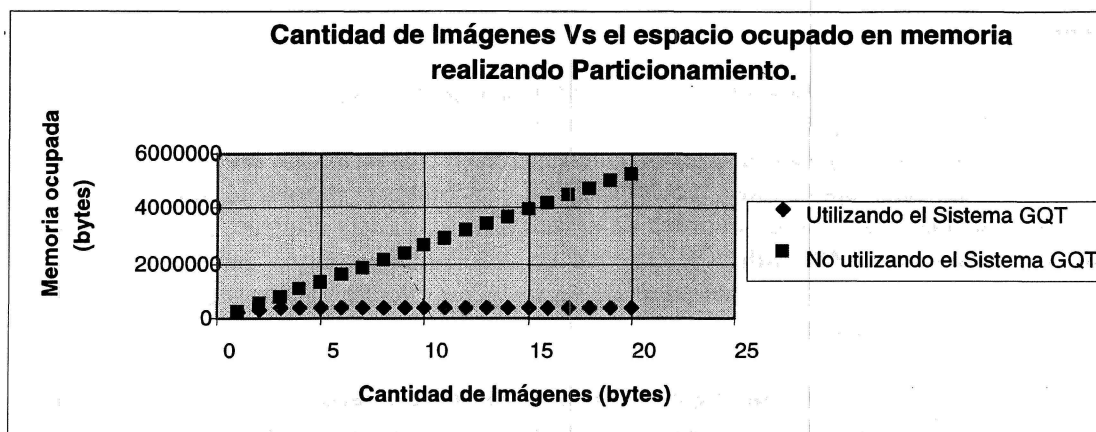
5.- Conclusiones

Hemos presentado un sistema de tratamiento de imágenes que permite aplicar operaciones sobre una imagen o sobre partes de esa imagen. Para ello la imagen es dividida en forma de árbol cuaternario. Los criterios de división de las imágenes dependen enteramente del usuario. El usuario puede decidir dividir una imagen y aplicarle operaciones a cualquiera de esas partes y guardar la imagen resultante. MIS almacena las imágenes, de acuerdo al orden de derivación, de manera eficiente al sólo guardar las partes modificadas. Para ello, MIS utiliza el enfoque de árboles cuaternarios genéricos. El usuario puede recuperar de manera transparente cualquiera de las imágenes producidas durante su tratamiento y conocer la derivación de esas imágenes.

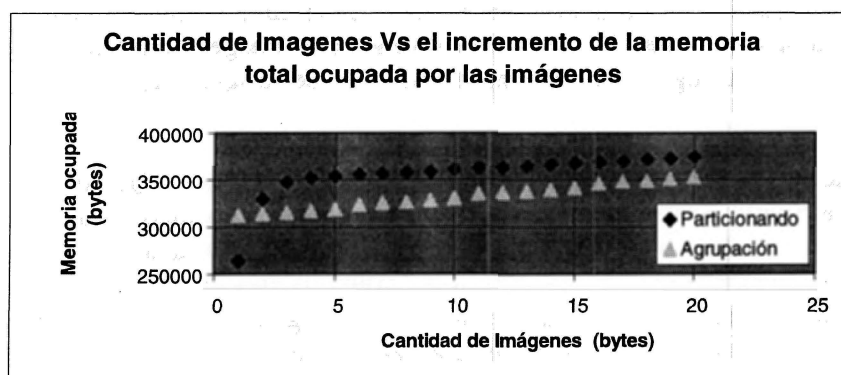
El prototipo presentado sólo dispone de algunas operaciones sobre imágenes pero es posible añadir nuevas operaciones. Se están desarrollando operaciones sobre las imágenes y sobre las regiones de una imagen, tales como: detección de bordes, segmentación, operadores morfológicos, etc. Así como también operaciones de manipulación de imágenes, que faciliten su control, como por ejemplo, saber cuales operaciones fueron aplicadas a una imagen hasta obtener una imagen resultante; mostrar la evolución de una imagen o de una región de una imagen después de la aplicación de una serie de operaciones.

Es evidente que la eficiencia de un sistema como MIS, que utiliza el enfoque GQT, depende de la similitud de los árboles cuaternarios correspondientes a la imagen inicial y a la imagen derivada por la aplicación de una operación. Cuando se aplican operaciones que modifican una gran cantidad de píxeles distribuidos uniformemente sobre la imagen, tales como filtros, los árboles cuaternarios correspondientes serán completamente diferentes aún cuando esas imágenes pueden lucir visualmente muy similares. En esos casos, el sistema es capaz de detectar tal situación, ya que la imagen resultante no comparte ningún nodo con su imagen original, y permitir al usuario decidir si es necesario almacenar toda la imagen resultante o sólo partes de ella.

La evaluación de uso de memoria presentada fue realizada a través de particionamientos o reagrupamientos regulares de partes de las imágenes. Una evaluación más exhaustiva de MIS se está realizando, en la cual se generan modificaciones aleatorias en las imágenes tratadas. De la misma manera estamos realizando un análisis comparativo del enfoque GQT utilizado en MIS con otros propuestos en la literatura para el almacenamiento eficiente de imágenes, basados en árboles cuaternarios solapados [12, 24, 28, 29]. Este análisis se está realizando tomando en cuenta no sólo el factor de almacenamiento sino también el uso esos enfoques en la aplicación de operaciones sobre imágenes.



(a)



(b)

Figura 6: Evaluación de MIS

6.- Referencias Bibliográficas

- [1] M. Aritsugi, M. Tabata, H. Fukatsu, Y. Kanamori and Y. Funyu. *Manipulation of Image Objects and Their Versions under CORBA Environment*. In 8th International Workshop on Database and Expert Systems Applications (DEXA '97), pag. 86-91, Toulouse, France, 1997. IEEE-CS Press.
- [2] F.W. Burton, M.M. Huntbach, and J.Y.G. Kollias. *Multiple Generation Text Files Using Overlapping Tree Structures*. The Computer Journal 28(4): 414-416, Aug. 1985.
- [3] F.W. Burton, J.G. Kollias, D.G. Matsakis and V.G. Kollias. *Implementation of overlapping B-trees for Time and Space Efficient Representation of Collections of Similar Files*. The Computer Journal, 33(3): 279-280, June 1990. Short Note.
- [4] M.J. Carey, D.J. DeWitt, J.E. Richardson and E.J. Shekita. *Object-Oriented Concepts, Databases, and Applications – Chap. 14: Storage Management for Objects in EXODUS*. Pag.

- 341-369. W. Kim, F.H. Lochovsky (Eds.), Addison Wesley - ACM Press. 1989. ISBN: 0-201-14410-7.
- [5] G. M. Davis. *A wavelet-Based Analysis of Fractal Image Compression*. IEEE Transactions on Image Processing, 7(2): 141-153, Feb. 1998.
- [6] C. Djeraba, I. Savory, M. Barere and S. Marchand. *Content Based Image Retrieval Model in an Object Oriented Database*. Image Databases and Multi-Media Search, series on Software Engineering and Knowledge Engineering, 8: 263-275, 1997. A.W.M. Smeulders and R. Jain (Eds) World Scientific. ISBN:981-02-3327-2.
- [7] C.R. Dyer, A. Rosenfeld, and H. Samet. *Region representation: Boundary codes from quadrees*. Communications of the ACM, 23(3): 171-179, 1980.
- [8] G. Jomier, M. Manouvrier and M. Rukoz. *Stockage et Gestion d'Images par un Arbre Quaternaire Générique. Por aparecer en 15 èmes. Journées en Bases de Données Avancées (BDA'99). Bordeaux-France, 1999.*
- [9] S. Khoshafian and H.B. Baker. *Multimedia and Imaging Databases*. Morgan Kaufmann, 1996. ISBN: 0018-9162.
- [10] S. Kawashima, M. Tabata, Y. Kanamori and Y. Masunaga. *Versioning Model of Image Objects for Easy Development of Image Database Applications*. In 7th International Workshop on Database and Expert Systems Applications (DEXA '96), pag. 194-200, Zurich, Switzerland, Sept. 9-10 1996. IEEE-CS Press.
- [11] A. Keller and J. Ullman. *A version numbering scheme with a useful lexicographic ordering*. Proc. of 11th Int. Conf. on Data Engineering (ICDE'95), pag. 240-248, Taipei (Taiwan), 1995.
- [12] Tsong-Wuu Lin. *Compressed quadtree representations for storing similar images*. Image and Vision Computing, 15: 833-843, 1997.
- [13] M. K. Mandal, S. Panchanathan and T. Aboulnasr. *Choice of Wavelets for Image Compression et Representation des Signaux Audiovisuels*. (CORESA'98) Lanion- Francia, 1998. Journées CNET. <http://www-rocq.inria.fr/imedia/>.
- [14] Y. Manolopoulos and G. Kapetanakis. *Overlapping B+-Trees for Temporal Data*. In Proc. of Jerusalem Conference on Information Technology (JCIT 90) - IEEE Computer Science Press, pag. 491-498, Jerusalem (Israel), Oct. 1990.
- [15] F. Mokhtarian, S. Abbasi and J. Kittler. *Efficient and Robust Retrieval by Shape Content through Curvature Scale Space*. Image Databases and Multi-Media Search, series on Software Engineering and Knowledge Engineering, 8: 51-58, 1997. A.W.M. Smeulders and R. Jain (Eds) World Scientific. ISBN:981-02-3327-2.
- [16] A. Nagasaka and Y. Tanaka. *Automatic video indexing and full video search for objects appearances*. IFIP Trans., Visual Database Systems II, pag. 113-127, Knuth, Wegner (Eds.), 1992. Elsevier.
- [17] N. Nappi, G. Polese and G. Tortora. *IFS Based Indexing in Image Databases*. Image Databases and Multi-Media Search, series on Software Engineering and Knowledge Engineering, 8: 233-240, 1997. A.W.M. Smeulders and R. Jain (Eds) World Scientific. ISBN:981-02-3327-2.
- [18] M.A. Nascimento and J.R.O. Silva. *Toward Overlapping R-trees*. In Proc. of ACM Symposium on Applied Computing (SAC'98), pag. 235-240, Feb. 1998.

- [19] N. Nes, C. van den Berg and M. Kersten. *Database Support for Image Retrieval Using Spatial-Color Features*. Image Databases and Multi-Media Search, series on Software Engineering and Knowledge Engineering, 8: 293-300, 1997. A.W.M. Smeulders and R. Jain (Eds) World Scientific. ISBN:981-02-3327-2.
- [20] H. Samet. *The Quadtree and Related Hierarchical Data Structures*. Computing Surveys, 16(2):187-260, June 1984.
- [21] H. Samet and R.E. Webber. *Hierarchical data structures and algorithms for computer graphics. Part I: Fundamentals*. IEEE Computer Graphics and Applications, 8(5): 48-68, 1988.
- [22] I.P. Stewart. *Quadtrees: Storage and Scan Conversion*. The Computer Journal 29(1): 60-75, Feb. 1986.
- [23] Du-Ming Tsai. *An improved generalized Hough transform for the recognition of overlapping objects*. Image and Vision Computing, 15: 877-888, 1997.
- [24] T. Tzouramanis, M. Vassilakopoulos and Y. Manolopoulos. *Overlapping Linear Quadtrees: a Spatio-temporal Access Method*. In 6th ACM Symposium on Geographic Information Systems (ACM GIS'98), Washington D.C., Nov. 1998.
- [25] Y.Valero, J.B. Ramirez Vivas. *QTV: Una estructura de árboles para el versionamiento de imágenes. Trabajo Especial de Grado*. Licenciatura en Computación. Universidad Central de Venezuela. 1998.
- [26] C. van den Berg, R. van den Boomgaard, M. Wooring, D. Koelma and A.W.M. Smeulders. *Horus: Integration of Image Processing and Databases Paradigms*. Image Databases and Multi-Media Search, series on Software Engineering and Knowledge Engineering, 8: 309-317, 1997. A.W.M. Smeulders and R. Jain (Eds) World Scientific. ISBN:981-02-3327-2.
- [27] P. van Oosterom. *Reactive Data Structures for Geographic Information Systems*. Oxford University Press, 1993. ISBN: 0-19-823320-5.
- [28] M. Vassilakopoulos,, Y. Manolopoulos and K. Economou. *Overlapping Quadtrees for the Representation of Similar Images*. Image and Vision Computing, 11(5): 257-262, June 1993.
- [29] M. Vassilakopoulos, Y. Manolopoulos, and B. Kroell. *Efficiency Analysis of Overlapped Quadtrees*. Nordic Journal on Computing, 2(1): 70-84, 1995.