Estructuras Métricas Paralelas en la Recuperación de Imágenes en la Web

Eduardo Peña-Jaramillo[1] and Roberto Uribe-Paredes[1,2]

[1] Depto. de Ingeniería en Computación[**],
Universidad de Magallanes, Chile.
[2] Grupo de Bases de Datos - UART,
Universidad Nacional de la Patagonia Austral, Rio Turbio, Argentina.
E-mail: {edopena,ruribe}@ona.fi.umag.cl

Resumen La recuperación de imágenes es una necesidad latente debido a la explosión de tecnologías de la información. Actualmente existen muy pocas máquinas de búsqueda de imágenes con enfoques basados en contenidos. Este enfoque ha ganado popularidad en la comunidad científica debido a que sus resultados han sido satisfactorios y de mayor efectividad que los sistemas basados en enfoques tradicionales.
Una de las líneas de investigación en los recuperadores de imágenes está dirigida hacia el manejo eficiente de los espacios de almacenamiento, construcción de índices y manejo de estructuras de búsquedas.
El presente trabajo propone implementar uno de estos métodos basados en contenidos sobre diferentes estructuras métricas, desarrollar distintas alternativas de paralelización de los algoritmos, distribución de la base de datos y realizar un análisis comparativo entre las estructuras y las distintas alternativas propuestas. Las versiones paralelas fueron implementadas usando el modelo BSP.

Keywords: Recuperadores de imágenes basados en contenidos, Estructuras de datos y algoritmos, bases de datos, búsqueda por similaridad, paralelismo, BSP.

1. Introducción

1.1. Antecedentes

Con la rápida evolución de las tecnologías de la información han surgido nuevos depósitos no estructurados de datos tales como texto libre, imagen, sonido y video. Realizar búsquedas exactas sobre estos datos sería poco útil. Por ejemplo, si se consultase por un elemento sobre una base de datos de imágenes, la consulta sólo podría encontrar su copia digital exacta en la base de datos. El verdadero interés reside, por ejemplo, en consultar sobre una base de datos de fotografías una imagen que contiene un rostro, donde no necesariamente existe una copia exacta de la misma fotografía; identificación de individuos a través de dispositivos biométricos, donde el dato consulta (voz, retina, etc) podría verse afectado por factores externos; encontrar una especie más parecida a otra en una base de datos de cadenas de ADN, etc. Este tipo de búsqueda recibe el nombre de *búsqueda por similaridad* y consiste en recuperar todos los objetos mas relevantes o parecidos a una consulta dada.
Para manipular dichos datos, se deben generar estructuras que permitan almacenarlos y realizar búsquedas sobre ellos. Estructurar este tipo de datos es dificultoso ya sea manual o computacionalmente y restringe de antemano los tipos de búsqueda posibles.
Los sistemas recuperadores de imágenes permiten entregar como respuesta un conjunto de imágenes. En este sentido, en la actualidad, existen nuevos tipos de máquinas de búsqueda con enfoques basados en contenidos los cuales pueden ser implementados sobre estructuras métricas. Finalmente, la necesidad de procesar grandes volumenes de datos obligan a aumentar la capacidad de procesamiento y con ello la paralelización de algoritmos y distribución de la base de datos.
En este trabajo se muestra la posibilidad de implementar un sistema recuperador de imágenes basado en contenido sobre estructuras métricas y propone algunas estrategias de distribución de carga en procesadores paralelos para hacer más eficiente la búsqueda de elementos en la base de datos. El método de computación paralela utilizado es el denominado modelo BSP [10] qué proporciona independencia de la arquitectura de la computadora.

[**] Parcialmente financiado por programa de investigación PR-F1-002IC-06, Universidad de Magallanes, Chile.

1.2. Marco Teórico

Recuperación de Imágenes Basadas en Contenido: Un sistema *CBIR* (*Content Based Image Retrieval*), sinónimo de (*feature-based retrieval*), nace del estudio de la percepción humana y recupera información por su contenido. Tiene como base una serie de tecnologías que permiten al usuario almacenar, indexar, clasificar y posteriormente recuperar información de imágenes por su contenido. Usa un análisis y procesamiento digital para generar descriptores a partir de los datos. Los pasos que se deben tener en cuenta para su construcción y funcionamiento son: extraer características de imágenes, almacenar índices, construir solicitudes de búsqueda y retornar resultados del proceso de búsqueda. Estos pasos se efectúan independientemente de la arquitectura del sistema, ya sea centralizada, distribuida o Cliente/Servidor.

Espacios Métricos: La similaridad, en muchos casos, es modelada a través de un *espacio métrico* y la búsqueda de objetos más similares bajo una función conveniente de similaridad, a través de una búsqueda por rango o vecinos más cercanos.

Definición 1 (*Espacios Métricos*): Un espacio métrico es un conjunto X con una función de distancia $d : X^2 \to R$, tal que $\forall x, y, z \in X$,

1. $d(x, y) \geq 0$ *and* $d(x, y) = 0$ *ssi* $x = y$. *(positividad)*
2. $d(x, y) = d(y, x)$. *(Simetría)*
3. $d(x, y) + d(y, z) \geq (d(x, z)$. *(Desigualdad Triangular)*

Definición 2 (*Consulta por Rango*): Sea un espacio métrico *(X,d)*, un conjunto de datos finito $Y \subseteq X$, una consulta $x \in X$, y un rango $r \in R$. La consulta de rango alrededor de x con rango r es el conjunto de puntos $y \in Y$, tal que $d(x, y) \leq r$.

El objetivo de los algoritmos de búsqueda es minimizar la cantidad de evaluaciones de distancia realizadas al resolver la consulta. Los métodos para buscar en espacios métricos se basan principalmente en dividir el espacio empleando la distancia a uno o más objetos seleccionados.

Existen dos métodos para las construcción de estructuras métricas, los basados en Clustering y los basados en Pivotes. El primero divide el área en particiones de Voronoi, donde existe un centro por cada área y los demás objetos se almacenan en el centro más cercano.
En el caso de los *Algoritmos Basados en Pivotes*, un pivote es un objeto preseleccionado y que no necesariamente pertenece a la base de datos. Su objetivo es filtrar objetos en una consulta a través de la utilización de la desigualdad triangular, sin medir realmente la distancia entre el objeto consulta y los objetos descartados.

Existen distintas estructuras para buscar en espacios métricos, las cuales pueden ocupar funciones discretas o continuas de distancia. Algunos son BKTree [3], MetricTree [8], GNAT [2], VpTree [11], FQTree [1], MTree [4], SAT [5,7], EGNAT [9].

Sin embargo, son pocas las estructuras que se tienen buen desempeño en espacios de alta dimensión, que es el caso de estudio. Además, son mucho menos las estructuras que permiten dinamismo (eliminación y reinserción de datos).

Modelo de Programación Paralela *BSP*: El modelo BSP de computación paralela fue propuesto en 1990 con el objetivo de permitir el desarrollo de software sea portable y tenga desempeño eficiente y escalable [10]. BSP propone alcanzar este objetivo mediante la estructuración de la computación en una secuencia de pasos llamados supersteps y el empleo de técnicas aleatorias para el ruteo de mensajes entre procesadores. El computador paralelo, independiente de su arquitectura, es visto como un conjunto de pares procesadores-memoria, los cuales son conectados mediante una red de comunicación cuya topología es transparente al programador. Los supersteps son delimitados mediante la sincronización de procesadores. Los procesadores proceden al siguiente superstep una vez que todos ellos han alcanzado el final del superstep, los cuales son agrupados en bloques para optimizar la eficiencia de la comunicación. Durante un superstep, los procesadores trabajan asincrónicamente con datos almacenados en sus memorias locales. Cualquier mensaje enviado por un procesador está disponible para procesamiento en el procesador destino sólo al comienzo del siguiente superstep. Dada la estructura particular del modelo de computación, el costo de los programas BSP puede ser obtenido utilizando técnicas similares a las empleadas en el análisis de algoritmos secuenciales. En BSP, el costo de cada superstep esta dado por la suma de el costo en computación (el máximo entre los procesadores), el costo de sincronización entre procesadores, y el costo de comunicación entre procesadores (el máximo enviado/recibido

entre procesadores). El costo total del programa BSP es la suma del costo de cada superstep.

Para el presente artículo se seleccionaron las estructuras SAT y EGNAT, las cuales han demostrado buen desempeño en espacios de alta dimensión. Todas estas estructuras son basadas en clustering y son del tipo árbol. Sobre dichas estructuras se implementará el modelo de recuperación de imágenes basadas en contenido y se mostrarán distintas estrategias de distribución de carga sobre procesadores paralelos permitiendo una búsqueda más eficiente dada una consulta por rango.

2. Sistema Recuperador de Imágenes

Recuperar información desde una imagen basada en contenido corresponde a una metodología de recuperación con respecto al dominio de aplicación del proceso de recuperación en sí. Usa un análisis y procesamiento digital para generar descriptores a partir de los datos. Los méritos principales de sistemas basados en el contenido son: soporta el procesamiento de consultas visuales, la consulta es intuitiva y amistosa al usuario, la generación de los descriptores es automática, siendo objetiva y consistente.

2.1. Extracción de Características

El sistema recuperador de imágenes implementado es este trabajo, fue propuesto en [6]. En este recuperador, durante el proceso de extracción de características, se miden las propiedades de todas las imágenes que comprenden cada una de las clases de la base de datos. A partir de estas características medidas, se puede decidir si los objetos imagen corresponden a una u otra clase y así poder discriminar clasificar y recuperar desde el subconjunto adecuado. Se trabaja con este tipo de características para lograr una dimensión alta de los vectores representativos de cada imagen. Si las características extraídas representan un valor global y la imagen es compleja, (más de cuatro regiones localizadas), estas características no servirían para describir la imagen debido a que se necesita información adicional relacionada con los tipos de regiones como por ejemplo su descripción geométrica y su importancia dentro de la imagen en estudio. Una forma de disminuir esta imprecisión frente a este tipo de descriptores globales es aplicar la técnica *layout*. Esta técnica consiste en subdividir la matriz imagen en la mayor cantidad posible de subcuadros y extraer de cada uno de éstos la misma información que se pretendía extraer en un comienzo en forma global.

Como las imágenes utilizadas para el presente trabajo, son de dimensión 256x256 píxeles, al aplicar la técnica layout, la imagen original queda subdividida en 256 cuadros de 16x16 píxeles cada uno. Realizar esta técnica simula el tener 256 regiones segmentadas de una imagen original con el mismo peso y a las cuales se les extraen sus características. Las características extraídas de cada una de las imágenes de la base de datos, del presente trabajo, corresponden a características globales de intensidad en 9 espacios de color, histogramas en espacio RGB, histograma en niveles de gris, 20 niveles de energía bajo técnicas Wavelet Standard, textura de Segundo momento angular (E), Contraste (I), Entropía (H) y Homogeneidad (Z), obtenidas de la matriz de co-ocurrencia en niveles de grises con distancias entre píxeles de d=1, d=2, y d=3 y con orientaciones de 0, 45, 90 y 135. La unión de todas estas características extraídas nos da como resultado un vector de 27 características con dimensión de 14.431 elementos. Este vector será el representante de la imagen dentro de la base de datos. Cabe destacar que cada característica fue normalizada antes de incorporarla al vector. Esta normalización consiste en transformar linealmente cada característica de tal forma que se tenga un valor medio igual a cero y una varianza igual a uno. Una vez extraídas las características de las imágenes se procedió a preseleccionar cuál de éstas cumplen con estar por sobre el 20 % de la función discriminante maximizada, (Fisher), de tal manera de asegurar una buena separabilidad entre las clases, esto se realiza para cada una de los conjuntos de imágenes.

Luego de la preselección de características se procedió a seleccionar mediante el método *Sequential Forward Selection* (*SFS*) las características más relevantes de las preseleccionadas. Para esto se trabajó con el discriminante Fisher para evaluar el desempeño de la clasificación.

Esta clasificación de imágenes sobre un mismo espacio es la que permite finalmente poder agruparlas a todas bajo una misma estructura y no tener una para cada clase. Si apareciese una imagen cuyo patron difiere de los que lideran la estructura de índice , ése pasa a ser un nuevo patrón o clase de búsqueda e indexación.

2.2. Clasificación y Búsqueda de Patrones

Cuando ya se tiene las características que describen a cada una de las 12 clases de imágenes propuestas para este trabajo se procede a definir los patrones que caracterizan a cada clase. Para diferenciar los falsos positivos en la recuperación de la base de datos de imágenes se realizó una clasificación para cada grupo de

imágenes. La clasificación analiza los rasgos de imagen y las clasifica en una de las clases siguientes: imagen falsa que no corresponde al contexto del grupo seleccionado e imagen positiva perteneciente al grupo de búsqueda. La dimensión de los vectores extraídos de cada uno de las 12 clases de imágenes varían de una clase a otra dependiendo de la cantidad de características que la definan. El sistema de recuperación propuesto debe contener a todas las imágenes en su índice el cual apunta a la base de datos. El problema de comparar imágenes cuyos vectores de características no concuerdan en dimensión y posición, se solucionó creando un patrón por cada clase de imagen. Esto permite poder almacenar bajo un mismo índice a todas las clases de imágenes.

3. Estrategias de Paralelización de estructuras

3.1. Estructuras Métricas

Las estructuras seleccionadas para implementar el método de recuperación de imágenes fueron los árboles: SAT [5,7] y EGNAT [9]. Éstas fueron seleccionadas por su buen desempeño en espacios de alta dimensión, la posibilidad de dinamismo y por la documentación existente. Inicialmente el conjunto de estructuras incluía al GNAT [2], pero ésta resultó menos competitiva que las otras dos, por lo que fue eliminada en los restantes experimentos. Las dos estructuras son árboles métricos basados en clustering. Para discriminar áreas durante la búsqueda, el SAT usa el criterio de radio cobertor y el EGNAT una mezcla entre radio cobertor y el criterio de hiperplanos. El EGNAT almacena tablas de rangos que le permite determinar si las áreas de búsqueda se intersectan con las tablas de rango almacenadas, reduciendo así los subárboles donde buscar.

3.2. Estrategias

Pese a la efectividad y robustez de las estructuras evaluadas, existen problemas de búsquedas para conjuntos de imágenes sobre 100.000 elementos, para lo cual se proponen estrategias de paralelización basadas sobre modelos BSP (*Bulk Synchronous Parallel*). En esta sección se describen estrategias de distribución de las estructuras SAT y EGNAT en múltiples procesadores y la paralelización de sus algoritmos de búsqueda. La paralelización se realizó utilizando modelo BSP, (*Bulk Synchronous Parallel*), para ver y evaluar el comportamiento y rendimiento de los algoritmos bajo las estrategias de distribución. Se midió cálculo observado por el número de evaluaciones de distancias entre objetos. El número de evaluaciones de distancia (ED), es la suma de todos los $maxED_i$ (número máximo de ED de un determinado procesador en el superstep i).

$$ED = \sum_{i=1}^{superstep} \text{máx } ED_i$$

La principal métrica que se utiliza para cómputos efectuados por cada procesador es el balance de carga. Ésta es medida por el radio promedio hacia el máximo número de cómputos observados en cada procesador. Esto es llamado eficiencia Ef y un valor 1 indica el óptimo.

$$Ef = \frac{\frac{ED = \sum_{i=1}^{N.\,Proc} \text{máx } ED_i}{Nro.\,Procesadores}}{\text{máx } ED \textit{ entre los procesadores}}$$

Los tiempos de la aplicación paralela (Tp), es la suma de todos los máxTp_i (Tiempo de CPU del procesador que más se demoró en el superstep i).

$$Tp = \sum_{i=1}^{superstep} \text{máx } Tp_i$$

Para todas las estrategias que se proponen, existe un broker que distribuye las consultas de forma circular entre todos los procesadores. Por cada superstep cada uno de los procesadores reciben Q consultas, (desde el broker), y realiza el proceso de búsqueda con esa información. Posteriormente se procede a distribuir las Q consultas, (procesadas anteriormente), a los demás procesadores y a la vez se envían resultados de las consultas a los procesadores que corresponda.

Estrategia 1: Estructura duplicada en cada procesador: Un primer acercamiento a la paralelización es asumir que los procesadores tienen suficiente memoria para mantener en cada uno una copia completa de la estructura de datos. Las consultas Q son distribuidas uniformemente entre los procesadores y su procesamiento es tal cual como si se aplicara localmente el algoritmo secuencial. No es requerido comunicación entre procesadores y cada consulta puede resolverse en un superstep. La Figura 1 muestra los resultados para esta estrategia con 3 procesadores, con grupos de 3 nuevas consultas por superstep. Se puede apreciar en esta estrategia que los resultados dependen directamente de la distribución de consultas. El promedio de eficiencia Ef es de 0.53 para el caso de estructura SAT y 0.55 para EGNAT, ambos con consultas de radio 0.1 % y 1 %. Sin considerar el problema de sobre consumo de memoria, esta estrategia no es conveniente debido a que no puede lograr un buen rendimiento.

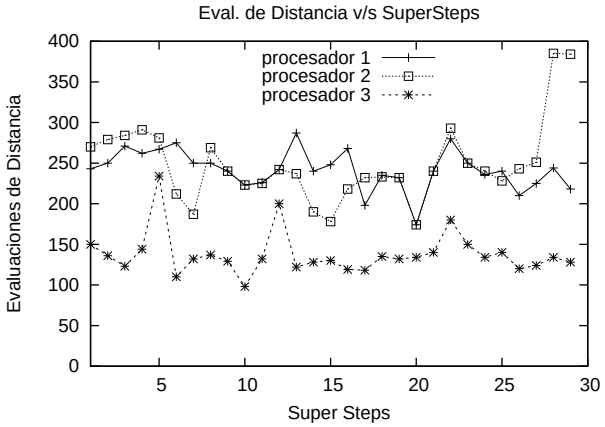

Figura 1. Estrategia 1: Estructura replicada en cada procesador. Comparaciones de cálculos de distancia por cada procesador versus superstep. Distribución circular de consultas, con radios de recuperación de 0.1 y 1 %.

Estrategia 2 (Estructura multiplexada en cada procesador): La siguiente estrategia que se propone consiste en distribuir en forma multiplexada entre los procesadores disponibles los subárboles enlazados a los nodos hijos de la estructura. No se requiere replicación de datos de la estructura, salvo la raíz y sus hijos. Al recuperar el 0.01 % y 0.1 % de los datos se puede observar que esta estrategia resultó tener una mejor eficiencia en la recuperación llegando a un promedio de $Ef = 0{,}61$ para estructura SAT y $Ef = 0{,}68$ para estructura EGNAT. A pesar de haber aumentado la eficiencia en la recuperación paralela la estrategia presentada presenta desbalances significativos de carga entre los procesadores, la estructura SAT en mayor proporción que EGNAT. La siguiente estrategia busca solucionar este problema equilibrando la cantidad de evaluaciones realizadas en cada superstep por cada procesador. Por esta razón es necesario distribuir los nodos del árbol entre los procesadores considerando el número de evaluaciones de distancias que pueden ser potencialmente realizados en cada subárbol enlazado a los hijos de la raíz de la estructura.

Estrategia 3 (Estructura multiplexada balanceada en cada procesador): Esta estrategia consiste en crear varios árboles a partir de la estructura original, un árbol por procesador, con la característica que cada árbol generado sólo tendrá algunos subárboles de la estructura original. Al momento de asignar un subárbol se consulta sobre quién es el procesador con menos nodos para que a éste se le asigne dicho subárbol. Los nodos raiz e hijos de la raíz están duplicados en cada procesador. Esta estrategia mejora la media $Ef = 0{,}74$ para estructura SAT y $Ef = 0{,}8$ para estructura EGNAT para las consultas exigidas en el caso de de la Estrategia 2. La mejora es significante con un pequeño aumento en el costo de la comunicación, (menos del 1 % con respecto al número de evaluaciones de distancias). La comunicación consiste en el envío de consultas entre procesadores durante el proceso de búsqueda, esto se hace necesario debido a que los subárboles se localizan en diferentes procesadores sin duplicaciones.

Estrategia 4 (Estructura multiplexada y balanceada recursivamente en cada procesador): Consiste en controlar la distribución de subárboles o nodos, según sea el caso, en los procesadores que se tengan

disponibles. Esta estrategia toma la estructura original y cuenta la cantidad total de nodos que la conforman. Se obtiene un promedio de los nodos que deberían ir distribuidos en total por cada procesador y ese valor será el patrón de control para ir incorporando nodos a los procesadores. Para incorporar un subárbol completo en un procesador se debe tener en cuenta que la cantidad de nodos que posee el subárbol debe ser menor igual al promedio aceptado por el procesador. En caso contrario se copia el nodo que encabeza ese subárbol en cada uno de los procesadores y se procede recursivamente a avanzar dentro del subárbol. Una vez avanzado al siguiente nodo del subárbol se consulta nuevamente por la cantidad de nodos en él.

Este proceso se repite tantas veces sea necesario hasta que el promedio de nodos sea menor igual al aceptado por los procesadores y que se haya recorrido todos los subárboles del nodo por donde se comenzó.

A medida que se avanza en los nodos y subárboles se van copiando en el procesador que tenga menos nodos de carga, (distribución balanceada). Los nodos raíz e hijos de la raíz están duplicados en cada procesador.

Esta estrategia mejora la media $Ef = 0{,}82$ para estructura SAT y $Ef = 0{,}9$ para estructura EGNAT para las mismas consultas exigidas en los casos anteriores. La mejora es significante pero aumenta nuevamente el costo de la comunicación pero a pesar de ello sigue siendo menos del 1 % con respecto al número de evaluaciones de distancias. Los costos de comunicación aumentan en los casos en que existen más nodos que ramas en un subárbol, (caso estructura SAT), y/o más elementos en el split que el promedio aceptado por cada procesador, (caso estructura EGNAT). La Figura 2 muestra los resultados de las tres últimas estrategias, las mas relevantes.

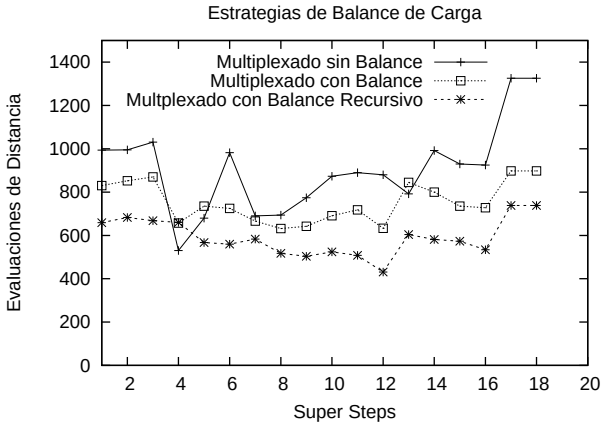

Figura 2. Comparación de estrategias de balance de carga. Distribución circular de consultas, con radios de recuperación de 0.1 % sobre 4 procesadores.

La figura 3 muestra los resultados en términos de eficiencia para las tres más relevantes estrategias, comparando los resultados para las dos estructuras. Los experimentos graficados en esta figura corresponde a un set de imágenes de 1.000 datos con dimensión 14.431, la ejecución fue realizada sobre 4 procesadores. Como se puede observar, la estructura EGNAT entrega mejores resultados en términos de eficiencia que la estructura SAT en espacios de alta dimensión. Este comportamiento ocurrio en las tres estrategias.

Las siguientes pruebas fueron realizadas sobre un segundo conjunto de datos correspondiente a 10.000 vectores de dimensión 14.431 que representan 27 características correspondientes a intensidades de color y textura de regiones extraidas de 10.000 imágenes. La estructura SAT de descartó debido a que el tiempo de construcción es exponencial a medida que aumenta la cantidad de elementos de la base de datos. Otro problema de la estructura SAT, es que al no estar optimizada para memoria secundaria requería de muchos recursos al trabajar directamente en memoria principal. EGNAT, en cambio, está diseñado para trabajar en memoria secundaria.

Se llevó a cabo la evaluación de la estructura formada con la estrategia de balance de carga multiplexado y recursivo con pruebas en 2 y hasta 10 procesadores para evaluar mediciones de distancias, tiempos, comunicación y eficiencia de la paralelización. Los resultados son mostrados en la figura 4.

Es importante notar que la eficiencia disminuye a medida que se trabaja con más procesadores, esto debido a la cantidad de evaluaciones realizadas por los mismos y por la alta comunicación existente entre ellos. Esta eficiencia se puede mejorar utilizando una variable de máximas evaluaciones por superstep lo cual postergaría el proceso de cálculo de distancias en el procesador hasta el siguiente superstep. Sin embargo, esto corresponde a

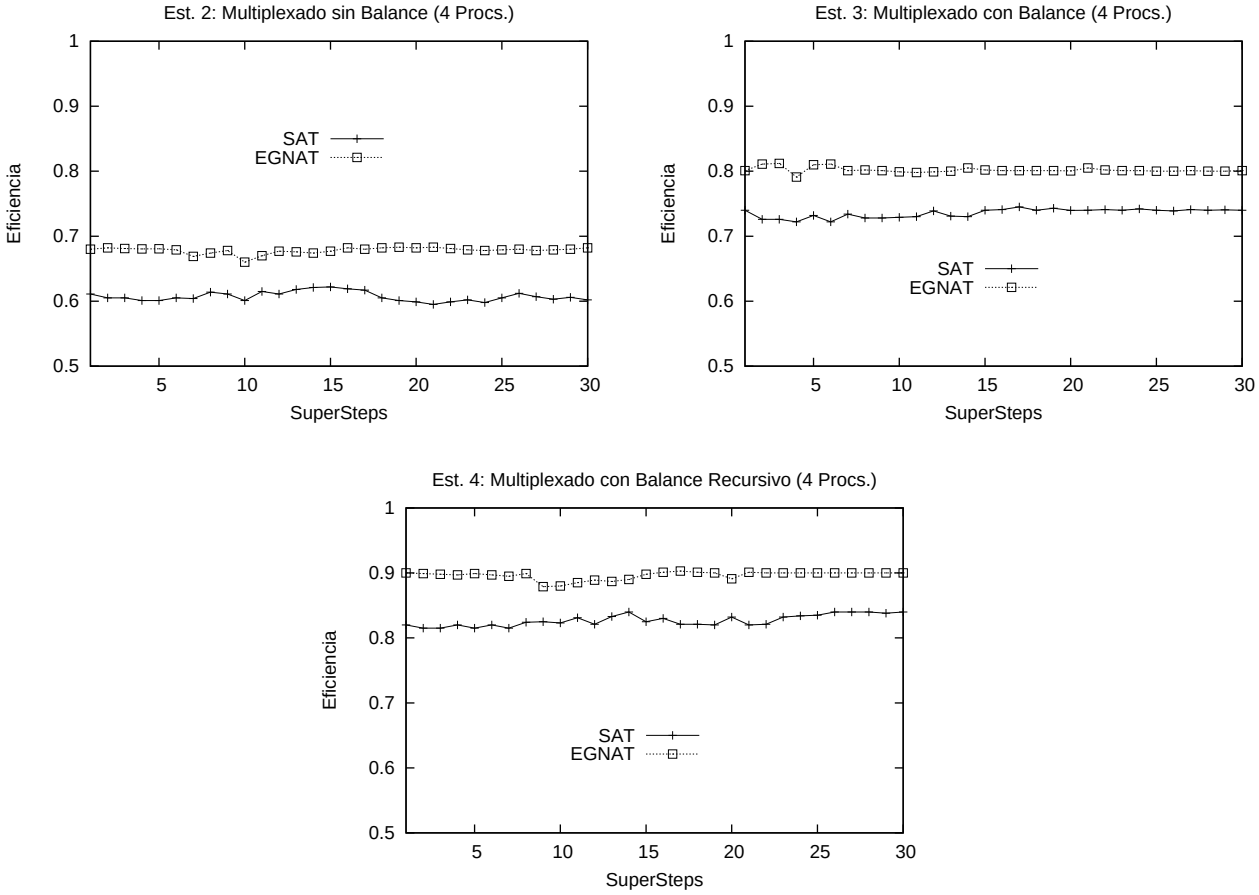

Figura 3. Gráficos de Eficiencia para cada Estrategia con ambas estructuras.

futuros trabajos. Respecto a los resultados en la cantidad de comunicación, cada punto indica la relación A/B dónde A es el número total de mensajes enviados entre los procesadores y B es el número total de tiempo que la función de búsqueda fue llamada para completar el proceso de todas las consultas.

4. Conclusiones

4.1. Aspectos Relevantes y Aportes

Aunque desde los años 90's que se estudian métodos que permitan recuperar imágenes almacenadas en algún sistema de base de datos, las soluciones obtenidas hasta ahora no son del todo satisfactorias. Los sistemas sencillos de Bases de Datos de Imágenes o bien no tienen capacidades de consulta o éstas son muy limitadas. Esto ha ocasionado que el trabajo presentado se centre en un CBIR que permite recuperar imágenes a partir de una descripción de los objetos que en ellas aparecen.

La implementación fue realizada sobre dos estructuras que han demostrado buen desempeño en búsquedas por similaridad. La eficiencia de las estructuras fueron dependientes de las dimensiones, de la cantidad de vectores en la base de datos y de si la estructura era manejada en memoria principal o secundaria. En este sentido los experimentos determinaron que a baja dimensionalidad del espacio, la estructura SAT se comportaba mejor que el EGNAT, pero en el caso contrario el mejor rendimiento lo ofrece las estructuras EGNAT.

Se considera que el principal aporte del presente trabajo es mostrar el desarrollo de una versión paralela eficiente de un nuevo CBIR y su implantación sobre estructuras métricas, permitiendo de esta manera acercar más estas estructuras a problemas verdaderamente reales, como es el caso de grandes volumenes de imágenes representados en espacios de muy alta dimensión.

Se considera, también, como parte de los aportes, el hecho de realizar análisis comparativos entre las distintas estructuras y entre distintas estrategias de paralelización de éstas.

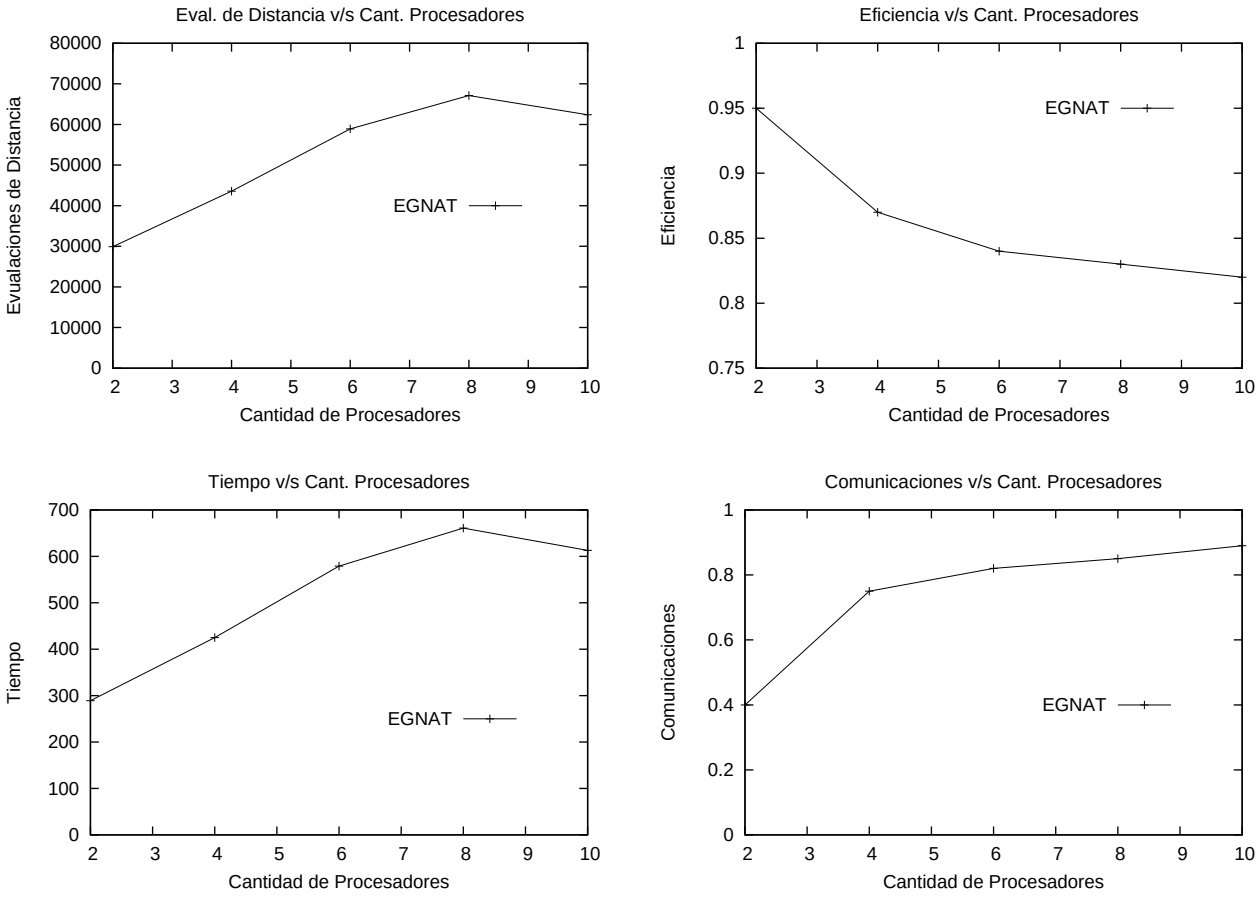

Figura 4. .

En este sentido, las estrategias presentadas fueron:

A. Duplicar la estructura original en todos los procesadores de los cuales se disponga.
B. Distribuir los nodos en forma multiplexada por cada uno de los procesadores que se tenga disponibles sin balancear.
C. Distribuir los nodos en forma multiplexada pero balanceada por los procesadores disponibles.
D. Distribuir los nodos en forma multiplexada pero balanceada recursiva (subdividiendo cada uno de las ramas del árbol a distribuir en caso de superar el subárbol, el promedio de nodos permitido por los procesadores).

La estrategia que mejor resultados arrojó fue la estrategia (D) seguida de la (C) con eficiencias de 0.9 y 0.8 respectivamente y utilización de base de datos de 10000 elementos. Por otro lado, se ha observado que la cantidad de comunicación y la sincronización es muy pequeña con respecto al costo de cálculos de distancia. El número de envíos de mensajes está por debajo del 1 % con respecto al número de distancia No se ha considerado el costo de enviar los objetos de la solución. Este costo tiene que ser pagado por cualquier estrategia. Por oto lado los experimentos llegaron a término en menos de 500 supersteps, lo cual es una cantidad muy modesta de sincronización para procesar 1.000 consultas. En las estrategias propuestas de multiplexión o distribución de subárboles de la estructura se debió considerar los subárboles pertenecientes a los hijos de la raíz. Puede darse el caso que se dispongan más procesadores que hijos. En este caso no podría paralelizarse la estrategia considerando la distribución de subárboles con uno o más niveles hacia abajo. Esos subárboles generan muy pocas comparaciones de distancia y en tal caso simplemente deben tratarse usando menos procesadores que los disponibles, (puede suceder en este caso que la estructura no admita más paralelismo), o acudiendo a la duplicación de algunos subárboles en los procesadores. El desempeño del recuperador de imágenes para estas dos estrategias propuestas se mide de acuerdo a los valores de sensibilidad y especificidad y para ambos casos los valores son 0.78 y 0.993 respectivamente.

Finalmente, este trabajo constituye una ayuda al desarrollo y comprensión del problema de búsqueda y recuperación eficiente de imágenes en una base de datos, dada una petición de usuario. Ésta, estaría determinada por la unión de nuevas técnicas de información que son estructuras basadas en similitud espacial (SAT y EGNAT), clasificación basada en contenidos y modelos de paralelización bajo BSP (Bulk Synchronous Parallel). La eficiencia del sistema de recuperación de datos propuesto bajo una estructura EGNAT y con estrategia de paralelización de distribución de los nodos en forma multiplexada balanceada y recursiva nos presenta un 90 %.

4.2. Trabajos Futuros

- Naturalmente, el paso siguiente es la implementación de prototipos reales del sistema de recuperación de imágenes, permitiendo además, la introducción de consultas online por parte del cliente.
- Realizar los experimentos sobre conjuntos mayores de imágenes e implementar las versiones paralelas sobre un cluster de grandes prestaciones.
- Desde el punto de vista de las estructuras métricas, experimentar el desempeño tanto de las estructuras secuenciales, como de sus versiones paralelas, la inserción y eliminación masiva de objetos.

Referencias

1. R. Baeza-Yates, W. Cunto, U. Manber, and S. Wu. Proximity matching using fixedqueries trees. In *5th Combinatorial Pattern Matching (CPM'94)*, LNCS 807, pages 198–212, 1994.
2. Sergei Brin. Near neighbor search in large metric spaces. In *the 21st VLDB Conference*, pages 574–584. Morgan Kaufmann Publishers, 1995.
3. W. Burkhard and R. Keller. Some approaches to best-match file searching. *Communication of ACM*, 16(4):230–236, 1973.
4. P. Ciaccia, M. Patella, and P. Zezula. M-tree : An efficient access method for similarity search in metric spaces. In *the 23st International Conference on VLDB*, pages 426–435, 1997.
5. Gonzalo Navarro. Searching in metric spaces by spatial approximation. *The Very Large Databases Journal (VLDBJ)*, 11(1):28–46, 2002.
6. Eduardo Peña-Jaramillo. Estructuras métricas paralelas en la recuperación de imágenes. Master's thesis, Escuela de Ingeniería, Departamento de Ciencias de la Computación, Pontificia Católica de Chile, Santiago, Chile, Nov. 2006.
7. Nora Reyes. Índices dinámicos para espacios métricos de alta dimensionalidad. Master's thesis, Universidad Nacional de San Luis, Argentina, 2002.
8. J. Uhlmann. Satisfying general proximity/similarity queries with metric trees. In *Information Processing Letters*, pages 40:175–179, 1991.
9. Roberto Uribe-Paredes. Manipulación de estructuras métricas en memoria secundaria. Master's thesis, Facultad de Ciencias Físicas y Matemáticas, Universidad de Chile, Santiago, Chile, Abril 2005.
10. L.G. Valiant. A bridging model for parallel computation. *Comm. ACM*, 33:103–111, Aug. 1990.
11. P. Yianilos. Data structures and algoritms for nearest neighbor search in general metric spaces. In *4th ACM-SIAM Symposium on Discrete Algorithms (SODA'93)*, pages 311–321, 1993.