

“Paralelización del Análisis de Similitud de Imágenes”

Lic. Marcelo Naiouf¹, A.C.Diego Tarrío², Ing. A. De Giusti³, A.C. Laura De Giusti²

*Laboratorio de Investigación y Desarrollo en Informática⁴
Facultad de Informática
Universidad Nacional de La Plata*

Resumen

Se presenta la arquitectura y esquema algorítmico para la paralelización del análisis de similitud de imágenes, basado en la obtención de múltiples firmas digitales para cada imagen, donde cada “firma” está formada por los coeficientes más representativos de la transformada wavelet del área de la imagen correspondiente.

En el trabajo se analiza la representación de imágenes por los coeficientes de una transformada wavelet y la conveniencia/necesidad de utilizar múltiples coeficientes para el estudio de similitud de imágenes que pueden tener componentes trasladados, con cambios de tamaño, color o textura.

La complejidad del cálculo involucrado justifica la paralelización y la solución propuesta constituye una combinación de un “pipelining” de multiprocesadores, cada uno de los cuales es una arquitectura paralela homogénea que obtiene los coeficientes de una firma (wavelet). La reusabilidad parcial de los cálculos para las sucesivas firmas obliga al pipelining de estas arquitecturas.

Palabras Clave

Algoritmos Paralelos. Análisis de similitud de imágenes. Reconocimiento de Patrones. Transformada de Wavelet. Arquitecturas paralelas.

¹ Profesor Adjunto Ded. Excl., Facultad de Informática, UNLP. E-mail: mnaiouf@info.unlp.edu.ar

² Alumnos avanzados de la Licenciatura en Informática. Facultad de Informática, UNLP. Becarios L.I.D.I.
E-mail: {ldgiusti, dtarrio}@info.unlp.edu.ar

³ Inv. Principal CONICET. Profesor Tit. Ded. Excl., Facultad de Informática, UNLP.
E-mail: degiusti@info.unlp.edu.ar

⁴ Calle 50 y 115 Primer Piso, (1900) La Plata, Argentina, Teléfono 54-221-4227707
WEB: lidi.info.unlp.edu.ar

Introducción

El procesamiento de imágenes, en términos generales, se refiere a la manipulación y análisis de información gráfica. Cualquier operación que actúe para mejorar, corregir, codificar, analizar, o de alguna forma cambiar la representación obtenida de una imagen se denomina “procesamiento de imágenes”.

El procesamiento de imágenes se descompone en las siguientes etapas [Gon96]: adquisición de la imagen (obtención de los datos y digitalización), preprocesamiento de la imagen (mejorar contraste, eliminar ruido, aislar regiones, etc) segmentación (partir una imagen de entrada en sus partes constituyentes u objetos), descripción (extracción de rasgos o características que diferencien clases de imágenes) interpretación (asignar significado a un conjunto de objetos conocidos).

En el procesamiento de una imagen puede ser conveniente realizar una codificación (por ejemplo al comprimir la representación digital de la imagen o al pasar del plano espacial al plano de la frecuencia). Normalmente podemos codificar una imagen con pérdida o sin pérdida. [Gon96].

Existe un amplio espectro de aplicaciones que utilizan el procesamiento de imágenes, entre las cuales se pueden citar: diagnóstico médico, defensa e inteligencia militar, investigaciones biológicas.

Muchas de estas aplicaciones requieren tiempos de respuesta en tiempo real, por lo que resulta necesario incrementar la eficiencia del cómputo de las etapas descriptas. La paralelización sobre una arquitectura multiprocesador es la alternativa natural, debido a las posibilidades de reducir los tiempos de proceso de cada una de las etapas mencionadas.

Representación de una imagen:

Una imagen digital es una función $f(x,y)$ que ha sido discretizada en coordenadas espaciales y brillo. También puede ser representada como una matriz, donde los índices de fila y columna identifican un punto en la imagen, y el valor de elemento en la matriz identifica el nivel de gris (o color) en ese punto (pixel).

El volumen de datos requeridos para el almacenamiento (y procesamiento) de una imagen, hace conveniente trabajar sobre una codificación de la imagen, tratando de trabajar sobre un conjunto mínimo de datos que respete (y permita reconstruir) las características más importantes de la imagen. Asimismo en ocasiones la codificación permite eliminar información redundante y se puede trabajar en el mejoramiento y análisis de la imagen, directamente sobre la representación codificada de la misma. [Gon96].

Ejemplos de codificación son las técnicas de Huffman, los histogramas por nivel de gris o color, la transformada de Fourier, la representación mediante Wavelets, etc. Naturalmente el nivel de reducción de los datos originales de la imagen puede estar asociada con una pérdida relativa de información.

Siempre interesa que la codificación admita inversa (es decir recuperar la imagen original o una aproximación con el menor error posible a la imagen original). En ocasiones también es importante la invariancia de la codificación a modificaciones de la imagen tales como traslaciones, cambios de color, cambios de escala o cambios de textura.

Este trabajo se aboca a la utilización de múltiples transformadas de Wavelet para obtener una codificación (firma) digital de una imagen [Cas95].

Similitud de imágenes:

Tradicionalmente, el problema de análisis de similitud de imágenes, es decir, el problema de encontrar el subconjunto de un banco de imágenes con características similares a una imagen dada, se ha solucionado calculando una “firma” (codificación) de cada imagen a comparar, para posteriormente analizar la correspondencia entre las firmas mediante una función de distancia que mida el grado de aproximación entre dos firmas dadas.

Los métodos tradicionales para calcular firmas, se basan en *algún* atributo de la imagen (por ejemplo el histograma de colores, el reconocimiento de un patrón fijo, el número de componentes de un tipo dado, etc). Esta “linealidad” de la firma hace muy difícil obtener datos sobre atributos no tenidos en cuenta en la firma (y que pueden hacer a la similitud o diferencia entre dos imágenes): por ejemplo si nos basamos sólo en el histograma de colores, no consideraremos la textura de la imagen, ni podríamos reconocer objetos similares pintados de diferente color.

Una solución a este problema consiste en definir firmas independientes sobre cada atributo de la imagen (color, textura, forma, etc) y luego combinarlas para obtener mejores resultados.

La alternativa que se investiga en este trabajo consiste en utilizar múltiples transformadas Wavelet como codificación digital de la imagen, obteniendo como firma el conjunto de los coeficientes de mayor peso [Nat99]. De esta manera se obtiene una firma compacta, que absorbe características importantes de la imagen como forma, color y textura.

Análisis de la representación de una imagen con una/múltiples firmas digitales

Wavelet

La transformada de Wavelet tiene excelentes características de compactación y extracción de propiedades de una imagen [Mas94][Cas95][Cod92], permitiendo que la misma pueda ser utilizada para generar efectivamente una representación compacta de la misma.

Utiliza una descomposición de la imagen en coeficientes, para luego almacenar sólo los coeficientes de mayor peso. De esta manera, se pueden generar arreglos de tamaño variable con los coeficientes obtenidos en el paso anterior.

Se puede obtener la transformada de Wavelet en una estructura de una o más dimensiones. En este artículo se tratará la transformada bidimensional Haar-Wavelet [Fou97][Nat99].

A continuación se sintetiza la descripción de la transformada de Wavelet en una dimensión:

Dada una imagen I representada de la siguiente manera:

$$I = [4,2,8,10]$$

El primer paso consiste en calcular los promedios de a pares de los componentes de la imagen, obteniendo el vector $[3,9]$. Este vector representa la misma imagen pero con la consecuente pérdida de detalle debido al promediado de pixels adyacentes.

Para restaurar la imagen original basándose en el vector resultante del primer paso, se guardan las diferencias entre el primer pixel de cada par y los promedios (coeficientes de detalle). Aplicando este paso al ejemplo, los coeficientes detalle resultantes serán $[-1,1]$ ($-1 = 3 - 4$, $1 = 9 - 8$).

Este paso se repite hasta reducir el vector a un único elemento, en el ejemplo: $[6]$, obteniendo como coeficiente de detalle $[3]$ ($3 = 6 - 3$). Como resultado del algoritmo se obtiene un vector I' concatenando el último vector de promedios y los vectores de detalle de los pasos anteriores, sobre el ejemplo se obtiene:

$$I' = [6,3,-1,1]$$

Cada entrada en I' es llamada coeficiente de Wavelet.

De manera análoga se puede obtener los coeficientes de Wavelet en dos dimensiones.

Esta transformada puede realizarse de dos maneras: descomposición estándar y no estándar. La primera consiste en utilizar la transformada de Wavelet de una dimensión para cada fila de pixels. Esta operación retorna la transformada de Wavelet de una dimensión para cada fila. En un segundo paso se aplica la misma transformada unidimensional a cada columna considerando como imagen las transformadas horizontales resultantes de la primer etapa. Los valores obtenidos serán los coeficientes de Wavelets mas un coeficiente de promedio general.

El método de descomposición no estándar calcula los promedios y diferencias horizontales para los pixels de cada fila de la imagen. Luego se realizan los promedios y diferencias verticales para cada columna resultante.

Estos pasos se repiten recursivamente sobre el cuadrante que contenga los promedios en ambas direcciones.

La matriz obtenida como resultado contiene los coeficientes de detalle donde el componente de la fila 1, columna 1 contiene el coeficiente principal.

Problemas de la similitud de imágenes

El cálculo de la transformada de Wavelet en una dimensión sobre una imagen completa en el problema de consultas por contenido, puede fallar al no encontrar similares ciertas imágenes que tienen objetos similares pero aparecen con cambios de escala o traslaciones (Figura 1) ya que las firmas pueden diferir en un valor por encima de la cota establecida.

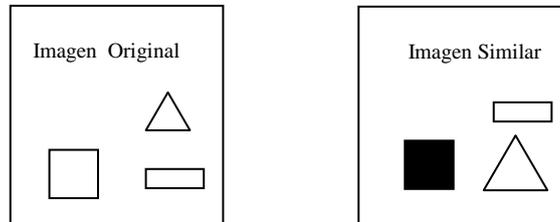


Figura 1: Imágenes similares

Para incrementar la precisión en el análisis de similitud de imágenes, [Nat99] propuso el cálculo de Wavelets en dos dimensiones por regiones a través de ventanas deslizantes y de tamaño variable, estableciendo como firma de una imagen un conjunto de firmas correspondientes a las distintas regiones de la misma.

Como resultado, para establecer la similitud entre dos imágenes se analiza el porcentaje de similitudes entre ambos conjuntos de firmas, si el porcentaje de firmas similares supera una cota establecida, se considera que ambas imágenes son similares debido al porcentaje del área coincidente.

Para minimizar el costo computacional, se realiza un proceso previo de clustering [Zha96] que genera una única firma para un conjunto de regiones similares de una imagen, reduciendo el costo de comparación entre las firmas de dos imágenes.

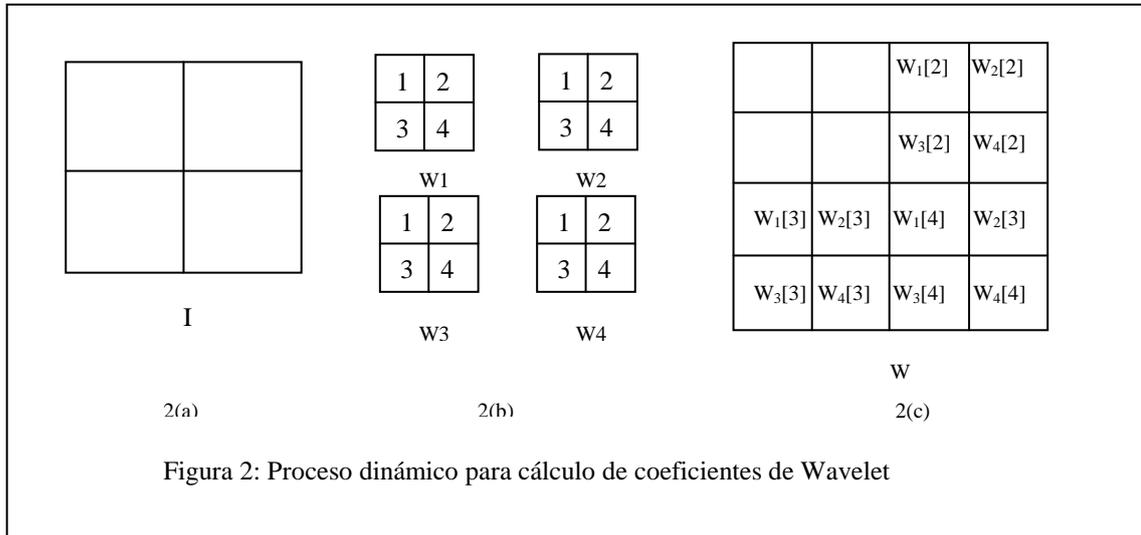
Algoritmo dinámico para obtener las múltiples firmas digitales de una imagen.

El cálculo de los coeficientes de wavelet para ventanas de tamaño $W \times W$ utilizando ventanas deslizantes de 1 pixel de distancia en una imagen de $N \times N$ implicaría un costo computacional $O(W^2(N-W)^2)$ y considerando que la solución secuencial requiere iterar este proceso para los distintos tamaños W de ventanas, resulta necesario reducir estos costos de cómputo, [Nat99] propone un algoritmo dinámico que calcula los coeficientes de ventanas de tamaño superior reutilizando el cálculo realizado para las ventanas menores. Asumiendo que las ventanas de tamaño $W/2 \times W/2$ están calculadas, podemos obtener las formas para ventanas de tamaño $W \times W$ usando las firmas de las correspondientes subventanas de tamaño $W/2 \times W/2$.

Este algoritmo obtiene la firma de una ventana I de tamaño $W \times W$ en base a las firmas de sus 4 cuadrantes W_1, W_2, W_3, W_4 (figura 2.a) de la siguiente manera: considerando una nueva descomposición de cada W_i en cuadrantes (figura 2.b), los coeficientes de menor peso correspondientes al 2do, 3er y 4to cuadrante de la ventana I tendrán los valores de los 2dos, 3eros y 4tos cuadrantes de las subventanas W_1, W_2, W_3 y W_4 como se observa en la figura 2.c.

Luego, para calcular el cuadrante superior izquierdo de I, se repite el proceso calculando los coeficientes de ese cuadrante con los coeficientes de los cuadrantes superiores izquierdos de tamaño $W/4 \times W/4$ de W_1, W_2, W_3 y W_4 .

El proceso recursivo termina cuando $W_1[1]$, $W_2[1]$, $W_3[1]$ y $W_4[1]$ contienen un único valor. En este punto los cuatro valores del cuadrante superior de 2×2 de W se obtienen realizando el promediado horizontal y vertical de los 4 valores $W_1[1]$, $W_2[1]$, $W_3[1]$ y $W_4[1]$.



Análisis de similitud, a partir de múltiples firmas digitales.

Modelo de similaridad de Imágenes

La mayoría de los algoritmos que solucionan el problema de retornar imágenes similares lo hacen calculando una firma de toda la imagen, cada una de las mismas basadas en histogramas de color, información de textura o los coeficientes topes calculados en la transformada de Wavelet. Luego la similaridad entre dos imágenes es definida en términos de la distancia entre sus firmas.

Los sistemas descriptos anteriormente fallan cuando un par de imágenes hacen "matching" sólo parcialmente, o cuando parte de los objetos de una imagen están en la otra en diferentes lugares, tamaño, o color.

El problema básico de este tipo de algoritmos es que calculan la firma sobre toda la imagen.

En este paper se investiga una solución basada en calcular la firma de una imagen partiendo la imagen en un conjunto de regiones.

Antes de dar una definición formal del concepto de similaridad entre dos imágenes se tendrá en cuenta que para cada imagen se obtienen todas las regiones por medio del cálculo de Wavelet sobre ventanas deslizantes aplicadas a la misma. Luego agruparemos estos resultados generando una estructura que permita almacenar los resultados en una forma compacta, esto es, guardando solo el centroide (promedio entre las firmas) de las firmas calculadas anteriormente cuando las mismas sean muy similares.

Ahora se definen los conceptos de similaridad de regiones y similaridad entre un conjunto de regiones para luego dar la definición de similaridad entre un par de imágenes.

Dadas dos imágenes Q y T las cuales consisten en un conjunto de regiones $Q_1 \dots Q_m$ y $T_1 \dots T_m$ respectivamente, se definen los siguientes conceptos:

Similaridad de regiones: un par de regiones se considera similar si una de sus firmas cae en una distancia ϵ de la otra.

Similaridad de conjunto de pares de regiones: para imágenes Q y T, y el conjunto de pares de regiones $\{(Q_1, T_1), \dots, (Q_k, T_k)\}$ se dice que es un conjunto de pares similares para Q y T, si Q_i es similar a T_i , teniendo en cuenta $i < j$, $Q_i < Q_j$ y $T_i < T_j$ (no se repiten las regiones para Q y T).

Similaridad de imágenes: se dice que dos imágenes Q y T son similares si existe un conjunto de pares de regiones similares para Q y T $\{(Q_1, T_1), \dots, (Q_k, T_k)\}$ tal que:

$$\text{Area}(\text{unión de todas la } Q_i) + \text{Area}(\text{unión de todas la } T_i) / (\text{área}(Q) + \text{área}(T)) \Rightarrow \alpha$$

De la definición anterior, se puede deducir que dos imágenes son consideradas similares si la fracción del área que hace "matching" comparada con el total de áreas de las dos imágenes es mayor a un parámetro α (α es el error permitido entre dos imágenes). Hay que tener en cuenta que permitiendo variaciones del parámetro α se puede afectar en el resultado que se obtiene para informar si dos imágenes son similares.

Como paso siguiente se define un algoritmo para el análisis de similitud de imágenes considerando los algoritmos explicados anteriormente, el cual se divide en cuatro pasos fundamentales:

1. Generar las firmas para cada una de las ventanas deslizantes.
2. Agrupar (cluster) las firmas obtenidas en alguna estructura.
3. Utilizar el matcheo de regiones.
4. Utilizar el matcheo de imágenes.

Generar las firmas para cada una de las ventanas deslizantes

Cada imagen es particionada en ventanas deslizantes (que se pueden solapear). Cada una de estas subventanas serán de un tamaño que varía desde $W_{min} * W_{min}$ hasta $W_{max} * W_{max}$. Para calcular la firma de cada una de las ventanas solo vamos a utilizar los últimos s coeficientes obtenidos en el cálculo de la transformada de Wavelet. En este paso se tiene en cuenta que cuando calculamos la transformada de Wavelet para una ventana de tamaño $W * W$ utilizaremos los resultados obtenidos cuando calculamos la transformada de Wavelet para la ventana de $W/2 * W/2$.

Agrupar las firmas obtenidas en alguna estructura

Después de realizar el paso anterior, el número de ventanas deslizantes generadas puede llegar a ser muy elevado. Obviamente almacenar todas las firmas para todas las ventanas generadas tiende a ser muy elevado en términos de espacio y costo de procesamiento. Una forma de reducir este overhead es agrupar las ventanas similares de una imagen en un cluster y almacenar sólo una firma representativa de todas las ventanas agrupadas en ese cluster (centroide).

Además para poder decidir si una firma puede ser incluida dentro de un cluster se utiliza la distancia de euclidean entre las dos firmas, luego si esta distancia es menor a un parámetro β (establecido previamente), esta ventana es absorbida dentro del cluster, caso contrario se genera un nuevo cluster. Cuando la ventana es absorbida la firma del cluster es recalculada.

Cada cluster contiene un conjunto de ventanas donde todas juntas forman una región, por lo tanto la imagen consultada se descompone en un conjunto de regiones.

Un punto a tener en cuenta es el valor dado al parámetro β para comparar si una ventana es absorbida o no, cuanto más chico sea este valor más clusters serán generados, por el contrario cuanto más grande sea el valor tendremos menos número de clusters generados.

Matcheo de regiones

Dadas dos imágenes Q y T y queremos ver si Q es similar a T, se utiliza un índice donde para cada región de la imagen Q, se encuentran todas las regiones de la imagen T que matchean. Para este paso se utiliza la definición de matcheo de regiones descripto previamente. En este paso obtendremos como resultado todos los conjuntos de regiones que matchean entre las dos imágenes.

Matcheo de imágenes

Como se explicó anteriormente después de ejecutar el paso anterior se obtienen todos los conjuntos de regiones que matchean entre las dos imágenes. En este momento se aplica la definición de matcheo de regiones descripta previamente para verificar si las imágenes son o no similares.

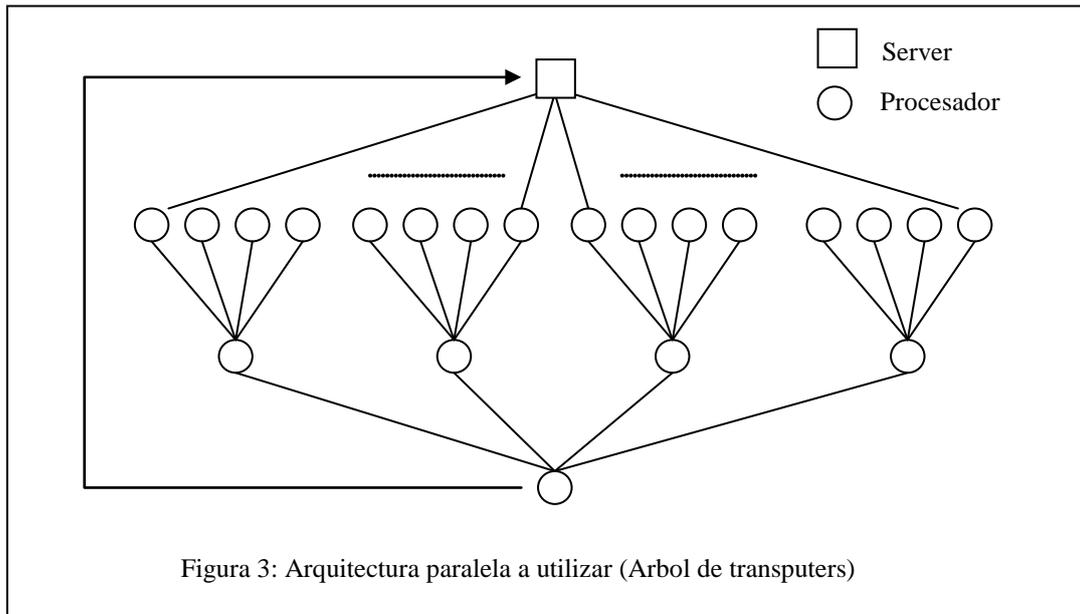
Arquitectura y algoritmo de paralelización.

El esquema de la arquitectura paralela a utilizar consiste en un árbol de procesadores (Transputers T805) (figura 3). Este tipo de procesadores resulta adecuado para el modelo de arquitectura planteado debido a su capacidad de conexión física a través de 4 links bidireccionales y memoria local de 4 MB .

Cada procesador se encarga de ejecutar el algoritmo dinámico para calcular una ventana de coeficientes de Wavelet de tamaño W en base a las 4 ventanas recibidas de tamaño W/2. Este resultado se envía al procesador asociado del siguiente nivel para que realice el cálculo de la ventana de coeficientes de tamaño 2W en base a los resultados de los 4 procesadores asociados del nivel anterior.

Como servidor de las ventanas de coeficientes de Wavelet ya calculadas y recolector de las ventanas de coeficientes de Wavelet retornadas por el árbol de procesadores, se dispone de un procesador Server S que realiza un broadcast de las ventanas y recibe un conjunto de ventanas de coeficientes como resultado del árbol.

Los resultados recolectados por el Server S servirán para realimentar el árbol de procesadores para poder calcular ventanas de tamaño superior. De esta manera el proceso de cálculo de las múltiples ventanas deslizante de tamaño variable consistirá en un "Pipeline" que alimentará al árbol de procesadores hasta calcular las ventanas deslizantes del tamaño máximo especificado.



Algoritmo Paralelo:

El algoritmo para obtener la firma digital de una imagen como un árbol balanceado de regiones correspondientes a las firmas digitales de las ventanas de tamaño variable de la imagen, inicia el proceso realizando un broadcast de los valores de pixels individuales como ventanas de coeficientes de tamaño 1x1, de esta manera los procesadores del primer nivel del árbol realizan el cálculo de las ventanas de coeficientes de tamaño 2x2, los procesadores del segundo nivel obtienen las ventanas de coeficientes de tamaño 4x4, etc. El nodo del último nivel del árbol genera su ventana de coeficientes de tamaño N basándose en las cuatro ventanas de tamaño N/2 recibidas y enviará el conjunto de ventanas obtenidas al nodo S, que se utiliza para realimentar el árbol.

Un esquema de este algoritmo se observa en las figuras 4 y 5 donde el nodo Server tiene la responsabilidad de servir los coeficientes de Wavelets ya calculados, recibir las ventanas de coeficientes resultantes del árbol y generar el árbol de clusters. Los nodos del árbol reciben 4 ventanas de coeficientes de tamaño N/2, ejecutan el algoritmo dinámico para generar la ventana de coeficientes correspondiente de tamaño N y retornan el conjunto de ventanas de tamaño N/2 y la nueva ventana de tamaño N al nodo del siguiente nivel. Finalmente el nodo del último nivel envía al server el conjunto de ventanas resultantes de este paso.

Procedure cálculoFirmaDigital (Firma: ArbolBalanceado)

Begin

w:= 1;

While (w <= tamaño Máximo) **do**

While (hay ventanas de tamaño w) **do**

Hacer broadcast de M ventanas de tamaño W al primer nivel de procesadores;

If (hay resultado del último nodo del árbol de procesadores) **then**

Recibir lista de ventanas de coeficientes del nodo del último nivel;

Realizar proceso de clustering de la lista de ventanas (Firma);

End If

End While

w:= w*2; /* avanza al siguiente tamaño de ventanas deslizantes */

End While

End;

Figura 4: Proceso del Server

(Broadcast, recolección y clustering de ventanas de coeficientes)

Procedure cálculoDinámicoVentanaDeCoeficientes

Begin

While True **do**

Recibir listas de resultados y 4 ventanas de los nodos del nivel anterior (L1,L2,L3,L4,W1,W2,W3,W4) ;

CálculoDinámicoVentanaDeCoeficientes (W1,W2,W3,W4, W);

Enviar a nodo del nivel inferior lista y resultado ([L1,L2,L3,L4,W1,W2,W3,W4], W);

End;

End;

Figura 5: Proceso de los nodos del árbol de procesadores

(Algoritmo dinámico para el cálculo de una ventana de coeficientes)

Conclusiones y Líneas de Trabajo actuales.

Se ha presentado un algoritmo de análisis de similitud de imágenes, basado en la obtención de múltiples firmas digitales de las imágenes a comparar y se ha establecido una métrica de similitud.

Posteriormente se esquematizó la solución paralela, partiendo de una arquitectura multiprocesador homogénea basada en transputers.

Se ha probado el reconocimiento del grado de similitud entre imágenes a partir del esquema propuesto y se está investigando la efectividad del método para imágenes con formas trasladadas, cambiadas de tamaño y de color.

Los problemas de eficiencia y escalabilidad del algoritmo paralelo, así como su posible migración a otra arquitectura física son líneas de investigación futuras, que forman parte del Trabajo de Tesina de Licenciatura de dos de los autores.

Bibliografía

- [Akl89] Akl S, "The Design and Analysis of Parallel Algorithms", Prentice-Hall, Inc., 1989.
- [Akl97] Akl S, "Parallel Computation. Models and Methods", Prentice-Hall, Inc., 1997.
- [Bec90] Beckmann N., Kriegel H., Schneider R., Seeger B., "The R*-tree: An Efficient and Robust Access Method for Points and Rectangles", Proceedings of the ACM SIGMOD, Atlantic City 1990. Pags. 322-331.
- [Bri95] Brinch Hansen, P., "Studies in computational science: Parallel Programming Paradigms", Prentice-Hall, Inc., 1995.
- [Cas95] Castro L., Castro S., "Wavelets y sus Aplicaciones", En los proceedings del 1er. Congreso Argentino de Ciencias de la Computación, Argentina 1995. Pags. 195-204.
- [Cha88] Chandi K. M., Misra J., "Parallel Program Design. A Foundation", Addison Wesley, 1988.
- [Cod92] Cody Mac A., "The Fast Wavelet Transform, Beyond Fourier Transform", Dr. Dobb's Journal. April 1992. Pags. 16-28.
- [Deg99] De Giusti L., Tarrío D., Naiouf M., De Giusti A., "Eficiencia y escalabilidad en algoritmos paralelos de cálculo del costo mínimo de caminos en grafos". En los anales del Congreso de las Nuevas Tecnologías Informática 2000. Cuba 2000.
- [Fou97] Fournier N. Castro G., "Algoritmo de Compresión de Imágenes Fijas Utilizando la Transformada de Wavelet", Tesina de Grado Lic. en informática U.N.L.P. 1997.
- [Gon96] González R., Woods R., "Tratamiento digital de imágenes", Addison-Wesley/Diaz de Santos, 1996.
- [Hwa93] Hwang K., "Advanced Computer Architecture: Paralelism, Scalability, Programability", McGraw-Hill, 1993.
- [IEEE] Colección de "IEEE Transactions on Parallel and Distributed Systems", IEEE.
- [Kum94] Kumar V., Grama A., Gupta A., Karypis G., "Introduction to Parallel Computing. Design and Analysis of Algorithms", Benjamin/Cummings, 1994.
- [Lei92] F. T. Leighton, "Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes", Morgan Kaufmann Publishers, 1992.
- [Mas94] Massopust P., "Fractal Functions, Fractal Surfaces, and Wavelets", Academic Press 1994.
- [Mil98] Miller R., Stout Q. F., "Algorithmic Techniques for Networks of Processors", CRC Handbook of Algorithms and Theory of Computation, M. J. Atallah, ed, 1998.
- [Nai99] Naiouf M., Tarrío D., De Giusti L., De Giusti A., "Análisis de tiempo y eficiencia en cálculo paralelo del costo mínimo de caminos en grafos". En los anales del III Workshop Chileno en Sistemas Distribuidos y Paralelismo (WCSDP). Talca (Chile) 1999.
- [Nat99] Natsev A., Rastogi R., Shim K., "WALRUS: A Similarity Retrieval Algorithm for Image Databases", Proceedings of the ACM SIGMOD 1999 Philadelphia. Pags. 395-405.
- [Nig95] Nigam M., Sahni S., "Sorting n^2 Numbers on $n \times n$ Meshes", IEEE Transactions on Parallel and Distributed Systems, Vol. 6, No. 12: Diciembre 1995, Pags. 1221-1225
- [Tra90] "Transputer Architecture and Overview. Transputer Technical Specifications.", Computer System Architects, 1990.
- [Zha96] Zhang T., Ramakrishnan R., Livny M., "BIRCH: An Efficient Data Clustering Method for Very Large Databases", Proceedings of the ACM SIGMOD Conference on Management of Data. Pags. 103-114, Montreal, Canada, Junio 1996