"Parallelization of Image Similarity Analysis"

Lic. Marcelo Naiouf [1], A.C.Diego Tarrío[2], Ing. A. De Giusti[3], A.C. Laura De Giusti[2]

Laboratorio de Investigación y Desarrollo en Informática[4]
Facultad de Informática
Universidad Nacional de La Plata

Summary

The algorithmical architecture and structure is presented for the parallelization of image similarity analysis, based on obtaining multiple digital signatures for each image, in which each "signature" is composed by the most representative coefficients of the wavelet transform of the corresponding image area.

In the present paper, image representation by wavelet transform coefficients is analyzed, as well as the convenience/necessity of using multiple coefficients for the study of similarity of images which may have transferred components, with change of sizes, color or texture.

The complexity of the involved computation justifies parallelization, and the suggested solution constitutes a combination of a multiprocessors "pipelining", being each of them an homogeneous parallel architecture which obtains signature coefficients (wavelet). Partial reusability of computations for successive signatures makes these architectures pipelining compulsory.

Key words
Parallel Algorithms. Image similarity analysis. Pattern recognition. Wavelet Transform. Parallel architectures.

[1] Full-Time Co-Chair Professor , Computer Science Faculty, UNLP. E-mail: mnaiouf@info.unlp.edu.ar
[2] Computer Science Licentiates. Computer Science Faculty, UNLP. L.I.D.I.
E-mail: {ldgiusti, dtarrio}@info.unlp.edu.ar
[3] Principal Researcher of CONICET. Full-Time Chair Professor, Computer Science Faculty, UNLP.
E-mail: degiusti@info.unlp.edu.ar
[4] 50 st. y 115 st. First Floor, (1900) La Plata, Argentina, Telephone: 54-221-4227707
WEB: lidi.info.unlp.edu.ar

Introduction

Roughly speaking, image processing refers to the management and analysis of graphic information. Any operation used to improve, correct, codify, analyze, or, in any way, to change the representation obtained of an image is called "image processing".

Image processing is decomposed in the following stages [Gon96]:
image acquisition (obtaining of data and digitalization),
image preprocessing (improve contrast, eliminate noise, isolate regions, etc.),
segmentation (partition an input image in its constituent parts or objects),
description (extraction of attributes or characteristics which differentiate image classes),
interpretation (meaning assessment to a set of known objects).

In the processing of an image, it may be convenient to carry out a *codification* (for instance, when zipping the digital representation of the image or when passing from the spatial plane to that of frequency). We can normally codify an image with or without loss. [Gon96].

There exists a wide spectrum of applications which use image processing, such as: medical diagnosis, military defense and intelligence, biological research.

Most of these applications require answering times in real time, which altogether results in the need of increasing the computation efficiency of the descriptive stages. The natural alternative is parallelization on a multiprocessor architecture, due to the chances of reducing processing times of each of the mentioned stages.

Image representation

A digital image is a function f (x,y) which has been discretized in spatial coordinates and brightness. It can also be represented as a matrix, in which the rates of line and column identify a point in the image, and the content value in the matrix identifies the level of gray (or color) in that point (pixel).

The volume of the required data for the storage (and processing) of an image, makes it convenient to work on a *codification* of the image, trying to work on a minimal set of data which respects (and allows to reconstruct) the most important characteristics of the image. Besides, codification usually allows to delete redundant information and it is easy to work on the improvement and analysis of the image directly on the codified representation of the same [Gon96].

Some of the examples of codification are: Hufman techniques, histograms for gray or color level, Fourier transform, the representation by means of Wavelets, etc. Obviously, the reduction level of the image original data can be associated to a relative loss of information.

It is always convenient that the codification admits inversion (i.e., recovering the original image or an approximation of that original image with the slightest error). Also, despite modifications made to the image, such as transferrings, color, scale or texture changes, it would be important to maintain codification invariability.

This paper has as its main objective the employment of multiple Wavelet transforms in order to obtain a digital codification (signature) of a digital image [Cas95].

Image similarity:

Traditionally, the problem of image similarity analysis –i.e., the problem to find the subset of a image bank with similar characteristics to a given image- has been solved by computing a "signature" (codification) of each image to be compared, so then, correspondence between the signatures could be analyzed by means of a distance function that measures the degree of approximation between the two given signatures.

Traditional methods to compute signatures are based on *some* attributes of the image (for example, color histogram, recognition of a fixed pattern, number of components of a given type, etc). This "linearity" of the signature makes it really difficult to obtain data about attributes which were not considered in the signature (and which could be relevant to the similarity or difference between two images): for instance, if we only take into account color histograms, we would not take into account image texture, nor we would be able to recognize similar objects painted in different colors.

The solution to this problem consists in defining independent signatures on each feature of the image (color, texture, form, etc), and then, combining them in order to obtain better results.

The alternative which is being investigated in this paper consists in using multiple Wavelet transforms such as image digital codification, obtaining as signature the set of more significant coefficients [Nat99]. In this way, a compacted signature is obtained, which, in turn, takes significant attributes from the image such as form, color and texture.

Analysis of representation of a Single/Multiple digital signature

Wavelet

Wavelet transform has outstanding characteristics for zipping and extracting image properties [Mas94][Cas95][Cod92], and, at the same time, it allows this image to be used to generate effectively a compacted representation of the same.

It uses an image decomposition in coefficients, in order to store the more significant coefficients. In this way, repairs of variable size can be generated with the coefficients obtained in the previous stage.

A wavelet transform can be obtained in a structure of one or more dimensions. In this paper, we will deal with Haar-Wavelet two-dimensional transform [Fou97][Nat99].

Next, a Wavelet transform description in one dimension is synthesized:

Given an I image represented as follows

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

The first stage consists in computing averages in pairs of the image components, obtaining the vector [3,9]. This vector represents the same image but with a consequent loss of detail, due to the average rate of adjacent pixels.

In order to recover the original image being based on the resulting vector of the first stage, the differences between the first pixel of each pair and the averages of detail coefficients are stored. Applying this stage to the example, the resulting <u>detail coefficients</u> will be [-1,1] (-1= 3-4, 1= 9-8).

In example [6], this stage is repeated until the vector is reduced to a single element, obtaining as detail coefficient [3] (3= 6-3). As a result of the algorithm, a vector I' is obtained, concatenating the last average vectors and detail vectors of the previous stages, thus, obtaining on the example:

I’ = [6,3,-1,1]

|Each input in I' is called Wavelet coefficient.

In a similar way, Wavelet coefficients can be obtained in two dimensions.

This transform can be carried out in two ways: standard and non standard decomposition. The first one consists in using a Wavelet transform of one dimension for each pixel line. This operation returns the Wavelet transform of one dimension for each line. In a second stage, the same one-dimensional transform is applied to each column considering as image the horizontal transforms resulting from the first stage. The values obtained are the Wavelet coefficients and a general average coefficient.

Non standard decomposition method computes horizontal averages and differences for the pixels of each line of the image. Then, vertical averages and differences are computed for each resulting column.

These stages are repeated recursively on a quadrant which contains the averages in both directions.

The resulting matrix obtained contains detail coefficients, in which the component of line 1, column 1 contains the principal coefficient.

Image similarity problems

Wavelet transform computing in one dimension on a complete image in the query by content problem, can fail when it does not find similar certain images, which, in turn, have similar objects but appear with a change of scale or transferring (Figure 1) since the signatures can differ in a value above the established level.

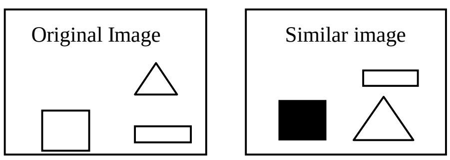

Figure 1: Similar Images

In order to increase the precision of the image similarity analysis, [Nat99] suggested Wavelet computing in two dimensions per regions by means of sliding windows of variable size, establishing as signature of an image a set of signatures corresponding to different regions of the same.

As a result, in order to establish similarity between two images, the percentage of similarities is analyzed between both sets of signatures; if the percentage of similar signatures surpasses a established level, both images are considered to be similar due to the percentage of the coincident area.

In order to reduce the computational cost, a previous clustering process [Zha96] is carried out, which, in turn, generates a single signature for a set of similar regions of an image, reducing the comparing cost between the signatures of two images.

Dynamic algorithm to obtain multiple digital signatures of an image.

Wavelet coefficient computing for windows of WxW size employing sliding windows of 1 distant pixel on an NxN image, would imply a computing cost of $O(W^2(N-W)^2)$ and, considering that the sequential solution requires iterating this process for windows of different W sizes, it is necessary to reduce these computing costs; [Nat99] suggests a dynamic algorithm which computes coefficient of bigger windows reusing the computation made for smaller windows. Assuming that windows of W/2 x W/2 size are computed, we can obtain the forms for windows of WxW size using signatures of the corresponding subwindows of W/2 x W/2 size.

This algorithm obtains the signature of a I window of WxW size basing on the signatures of its four quadrants W1,W2,W3,W4 (figure 2.a) as follows: considering a new decomposition of each Wi in quadrants (figure 2.b), less significant coefficients corresponding to the 2nd, 3rd and 4th quadrants of the I window, would have the values of the 2nd, 3rd and 4th quadrants of the subwindows W1, W2, W3 and W4, as figure 2.c. shows.

Then, in order to compute the left superior quadrant of I, the process is repeated by computing the coefficients of that quadrant with the left superior quadrant coefficients of W/4 x W/4 size of W1,W2,W3 and W4.

The recursive process finishes when W1[1], W2[1], W3[1] and W4[1] have a single value. At this point, the four values of the 2x2 superior quadrant of W are obtained by carrying out the horizontal and vertical averaging of the 4 values W1[1], W2[1], W3[1] y W4[1].

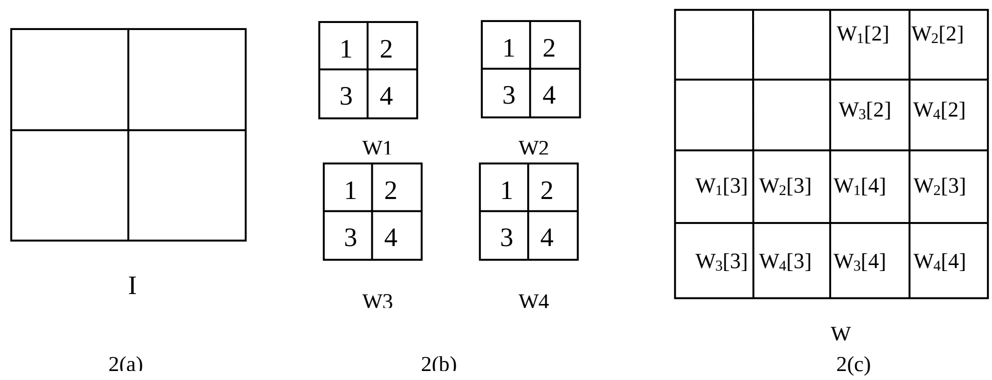

Figure 2:Dynamic process for Wavelet coefficient computation.

Similarity analysis, from multiple digital signatures.

Image Similarity Model.

Most of the algorithms which solve the problem of returning similar images, compute the signature of the whole image, each of them based on color histograms, texture information or more significant coefficients computed in the Wavelet transform. Then, similarity between the two images is defined in terms of the distance between their signatures.

The previously described systems fail when a pair of images match only partially, or when a parts of the objects of an image are located in different places, size, or color of the other image.

The basic problem of this type of algorithms is that they compute the signature on the whole image.

This paper presents a research of a solution based on computing an image signature, partitioning the image in a set of regions.

Before giving a formal definition of the similarity concept between two images, there will be certain issues to be taken into account, namely the fact that for each image, all regions are obtained by means of Wavelet computing on sliding windows applied to the same. Then, we will group these results by generating a structure which allows to store the results compactly, i.e., by graduating only the centroid (average between the signatures) of the previously computed signatures, when the same are very similar.

Now, region similarity concepts are given, as well as those of similarity among a set of regions, in order to define later similarity between a pair of images.

Given two images Q and T, which consist of a set of regions Q1...Qm and T1...Tm respectively, the following concepts are defined:

Region similarity: a pair of regions is considered similar if one of its signatures differs in a e distance from the other.

Set of region pairs Similarity: for Q and T images, and the set of pairs of regions {(Q1,T1),...,(Qk,Tk)}, let them be a set of similar pairs for Q and T if Qi is similar to Ti, taking into account i <> j, Qi<>Qj y Ti<>Tj (regions are not repeated for Q and T).

Image similarity: let images Q and T be similar if there exists a set of region pairs similar to Q y T {(Q1,T1),...,(Qk,Tk)}, so that:

$$\text{Area}(\cup Qi) + \text{Area}(\cup Ti) / \text{Area}(Q) + \text{Area}(T) \Rightarrow \alpha$$

From the previous definition, it can be deduced that two images are considered to be similar if the fraction of the area which matches, compared with the total of areas of both images, is higher than a α parameter (let α be the allowed error between two images). It is worth mentioning the fact that, allowing α parameter variations, it can be affected the result in order to know whether two images are similar or not.

Considering the previously explained algorithms, it follows another stage in which an algorithm is defined for the image similarity analysis. This algorithm is, in turn, divided in four fundamental stages:

1. Generating signatures for each of the sliding windows.
2. Grouping (clustering) signatures obtained in some structure.
3. Using region matching.
4. Using image matching.

Generating signatures for each of the sliding windows.

Each image is partitioned in sliding windows (which can be overlapped). Each of theses subwindows will be of different sizes ranging from Wmin*Wmin to Wmax*Wmax. In order to compute the signature of each of the windows, we will only employ the last *s* coefficients obtained in the Wavelet transform computing. In this stage, when we compute the Wavelet transform for a window of W*W size, we will employ the results obtained when we computed the Wavelet transform for a window of W/2*W/2 size.

Grouping signatures obtained in some structure

After carrying out the previous stage, the number of generated sliding windows can be really high. Evidently, storing process of all the signatures for all the generated windows tends to represent an elevated cost in terms of space and processing. A way to reduce this overhead is grouping similar windows of an image in a cluster and storing only one representative signature of all the windows grouped in that cluster (centroide).

Besides, in order to decide whether a signature can be included in a cluster, an Euclidean distance is used between both signatures. Then, if this distance is lower than a β parameter (previously established), this window is absorbed within the cluster, and if this is not the case, a new cluster is generated. When the window is absorbed, the signature of the cluster is computed once again.

Each cluster has a set of windows in which altogether form a region, hence, the query image is decomposed in set of regions

In order to compare whether a window is absorbed or not, a relevant issue to be considered is the value of a given β parameter; the lower the value, the higher the number of clusters generated. And, on the contrary, the higher the value, the lower the number of cluster obtained.

Region Matching

We would like to verify whether Q is similar to T, so given two images Q and T, a rate is used in which for each region of Q image, all matching regions of T image are found. For this stage, the definition of region matching is used –which was previously described. As a result, we will obtain all sets of regions which match between both images.

Image matching

As previously explained, after executing the previous stage, all sets of regions which match between both images are obtained. At this moment, the definition of region matching previously described applied in order to verify whether the images are similar or not.

Parallelization architecture and algorithm.

The structure of parallel architecture to be used consists in a processor tree (Transputers T805) (figure 3). This type of processors is suitable for the presented architecture model due its capacity of physical connection by means of 4 two-directional links and local memory of 4 MB.

Each processor is in charge of executing the dynamic algorithm in order to compute a Wavelet coefficient window of W size, basing on the four windows of W/2 size received. This result is sent to the associated processor of the next level, so then it computes the Wavelet coefficient window of W size basing on the results of the 4 associated processors of the previous level.

As server of Wavelet coefficient windows already computed, and as collector of Wavelet coefficient windows returned by the processor tree, a S Server processor is used, which, in turn, broadcast the windows and receives a set of coefficient windows as a result of the tree.

The results collected by S Server will be used to feed the processor tree once again in order to compute bigger windows. In this way, computing process of multiple sliding windows of a variable size will consist of a "Pipeline" which will feed the processor tree until it calculates the sliding windows of specified maximum size.

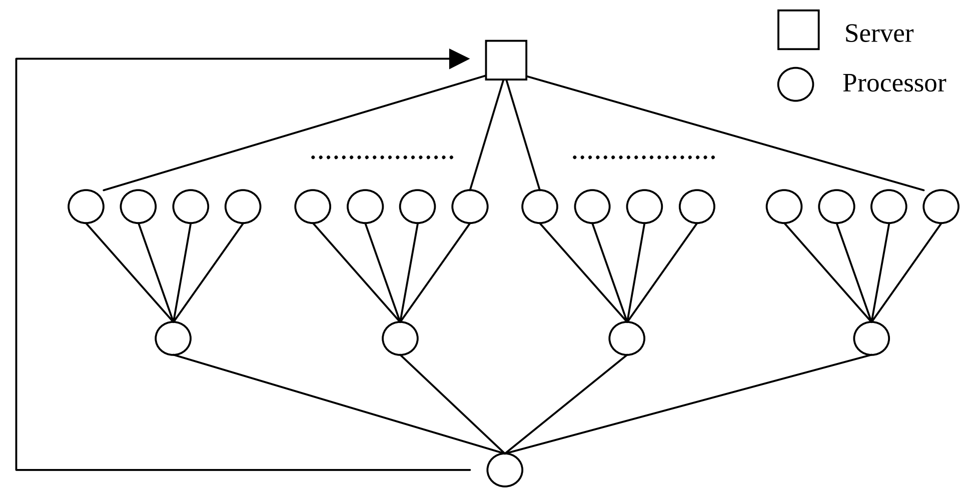

Figura 3: Parallel Architecture to be used (Transputer Trees)

Parallel algorithm:

In order to obtain a digital signature of an image as a balanced region tree corresponding to the digital signatures of an image window of variable size, the algorithm starts a process by making a broadcast of individual pixel values as coefficient of windows of 1x1 size. In this way the processors of the first level of the tree compute the coefficients of windows of 2x2 size, the processors of the second level obtain the coefficients of windows of 4x4 size, etc. The node of the last level of the tree generates its coefficient of windows of N size basing on four windows of N/2 size received, and it sends the set of windows obtained in S node, which is used to feed the tree once again.

The structure of this algorithm is shown in figures 4 and 5 in which the Server node is responsible for serving the already calculated Wavelet coefficients, receiving coefficient windows resulting from the tree, and generating the cluster tree. The tree nodes receive 4 coefficients of windows of N/2 size, they execute the dynamic algorithm to generate the coefficient window corresponding to N size and return the set of windows of N/2 size and the new window of N size to the node of the next level. Finally, the node of the last level sends the server the set of resulting windows of this stage.

```
Procedure DigitalSignautrecomputing (Signature: BalancedTree)
Begin
   w:= 1;
  While (w <= Maximum size) do
    While (there are windows of w size) do
          Carry out broadcast of M windows of W seize to the first level of processors;
          If (there is a result of the last node of the tree processors) then
               Receive list of coefficient windows of the last level node;
               Carry out  clustering process of the list of windows (Signature);
       End If
    End While
    w:= w*2;    /* moves forward to the next size of sliding windows */
  End While
End;
```

Figure 4: Server Process
(Broadcast,collection y clustering of coefficient windows)

```
Procedure CoefficientWindowDynamicComputing
Begin
  While True do
          Receive results lists and 4 windows of the previous level nodes (L1,L2,L3,L4,W1,W2,W3,W4);
          CoefficientWindowDynamicComputing (W1,W2,W3,W4, W);
   Send list and list to inferior level node ([L1,L2,L3,L4,W1,W2,W3,W4], W);
  End;
End;
```

Figure 5: Process of the processor tree nodes
(Dynamic Algorithm for coefficient window computing)

Conclusions and current Lines of Work.

An image similarity analysis algorithm has been presented based on the obtaining of multiple digital signatures from the images to be compared, and a similarity metrics has been established.

Then, a parallel solution was schematized, starting form an homogeneous multiprocessor architecture based on transputers.

Recognition of the degree of similarity between images has been proved on the basis of the suggested scheme and the effectiveness of the method is being studied for images with transferred forms, with color and size changes.

Problems of effectiveness and scalability of the parallel algorithm, as well as its possible migration to another physical architecture, are future lines for research, which, in turn, represent part of the Postgraduate Research of the Licentiate Degree of all of the authors.

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”, Addisson 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 parallelos 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: Parallelism, 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. Desing 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 parallelo del costo mínimo de caminos en grafos”. En los anales del III Workshop Chileno en Sistemas Distribuidos y Parallelismo (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. Transpurer 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