Driving to a Fast IMS Feature Vector Computing

J. Fernández, R. Guerrero, N. Miranda, F. Piccoli
Universidad Nacional de San Luis
Ejército de los Andes 950
5700 - San Luis - Argentina
e-mail: {jmfer, rag, ncmiran, mpiccoli}@unsl.edu.ar

Abstract

Increasing amount of image data transmitted via Internet has triggered the development of general purposes Image Mining Systems (IMS).
An IMS performance relies on a good and fast feature vector specification that describes univocally an entire image. Vector size and the relationship between each evaluated feature and its computation time are critical, moreover when the image amount is big enough. Decreasing this IMS computational complexity by means of parallelism at the different involved tasks is one solution.
Nowadays clusters of computers are already widely used as a low cost and high utility option to special-purpose machines, and suited to solve image processing problems with a high degree of data locality and parallelism.
At this paper, we will focus on parallelism into the IMS processing stage trying to accelerate the feature vector calculus thru a cluster architecture attempting to give a better performance to the whole image mining system.

Keywords: Parallel computing, Embarrassingly parallel, Data-intensive and I/O demanding computing, Image processing, Signal processing.

Resumen

La creciente cantidad de imágenes transmitidas a través de Internet ha llevado al desarrollo de Sistemas de Minería de Imágenes de propósito general.
La performance de un SMI depende en gran medida de una rápida y buena especificación del vector característica que describe unívocamente a una imagen completa. El tamaño del vector y las relaciones existentes entre cada una de las características evaluadas y su tiempo de procesamiento son críticos, más aún cuando la cantidad de imágenes es lo suficientemente grande. Una posible solución consiste en el uso de paralelismo en las diferentes tareas involucradas en un SMI.
Hoy en día, los clusters de computadoras son una opción ampliamente utilizada, con un bajo costo y alto rendimiento, principalmente para máquinas de propósitos específicos y se adaptan a la resolución de problemas de procesamiento de imágenes con un alto grado de paralelismo y localidad de datos.
En este paper nos focalizaremos en el paralelismo de la etapa de procesamiento de un sistema SMI con la intención de acelerar el cálculo del vector característica por medio de una arquitectura cluster intentando brindar una mejor performance al sistema SMI en su totalidad.

Palabras claves: Computación paralela, Paralelismo embarazoso, Computación con uso intensivo de Datos y Entrada/salida, Procesamiento de Imágenes, Procesamiento de señales.

1 INTRODUCTION

With the advances in multimedia data acquisition and storage techniques, the need for automatic knowledge discovery from large image collections is becoming more and more relevant.

Automatic image categorization tries to simulate how the human visual system processes visual contents of images and categorizes them. The aim is to build a model using attributes extracted from and attached to stored images, then evaluating the effectiveness of the model by new images. Image mining deals with the study and development of new technologies that allow accomplishing this subject.

Figure 1 shows a general structure model for an Image Mining System (IMS). The system considers an specified amount of stored images, whose image features had been extracted to represent concisely their image content -*Transformation and feature extraction phase*-. Besides the relevance of this task, it is essential to consider the invariance problem to some geometric transformations and robustness with respect to noise and other distortions before designing a feature extraction operator -*Pre-processing phase*-. After the tranformation and feature extraction phase, a ***model description*** representing a correct semantic image interpretation is obtained. As a result, visual contents of images in the database will be characterized by descriptive patterns -usually a multidimensional feature vector-.

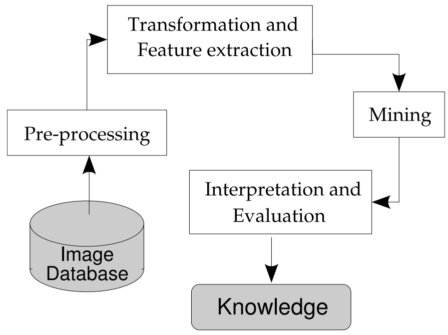


Figure 1: General Image Mining System.

Mining phase leads to describe new association rules for the stored complex information. It tries to define valid, novel, potentially useful, and ultimately understandable patterns, relations or rules from them, attempting to draw a *Predictive* or *Descriptive* model.

Interpretation and Evaluation phase tries to generate all significant patterns without any knowledge of the image content. It is tightly related with mining phase because it measures the quality from obtained relational patterns. Model preciseness can be secured by guarantying data independency between the training data set and testing data set [18, 23].

Query decisions are based on a given symbolic description of a searched image content. The ***symbolic description*** might be just a feature or a set of features, a verbal description or phrase in order to identify a particular semantic. Image mining results are acquired after matching the symbolic description with its complementary model description and the corresponding discovered knowledge relationships.

The development of an image mining system is often a complex process since it implies joining different techniques ranging from data mining and pattern recognition up to image retrieval and indexing schemes. An IM system basically should assemble the following tasks: *image storage*, *image pre-processing*, *feature extraction*, *image indexing and retrieval* and, *pattern and knowledge discovery.*

There exist many investigations in the area, trying to discover useful image patterns for the understanding of existing interactions between image human perception at high level and image features at low level [18, 22, 25, 31]. The big challenge is extracting implicit knowledge, image data relationships, or other features not explicity stored in a pixel representation. Besides, it is expected that a good image mining system provides users with an effective access into the image repository at the same time it recognizes data patterns and generates knowledge underneath image representation.

As a result, an image mining system implies lots of tasks to be done in a regular time. The use of high performance computing in every or some mining tasks might be a good option to reduce the cost and overhead of the whole image mining process. Some approaches try to define a parallel model and analyse their theoretical performance cost [1, 10, 11].

The rest of this paper is organized as follows. Section 2 explains the significance of the processing phase into an IMS. Section 3 states some issues about how an IMS could be parallelized. Section 4 introduces the "what" and "where" systems of human visual system and how it could be related to the IMS processing stage. Last sections describe the high performance function implementations related to the introduced "what" system and some conclusions about their computations.

2 PROCESSING PHASE SIGNIFICANCE

Many issues of image mining can be optimized with different parallel techniques. Furthermore depending on tasks properties, different parallel paradigms could be applied in the same system.

Since real-life data is often incomplete, noisy and inconsistent, pre-processing becomes a necessity enablying data cleaning by removing noise or other aspects that could mislead the actual mining process. This image enhancement helps in qualitative improvement of the image and can be done either in the spatial domain or in the frequency domain. The most common techniques applied for data cleaning are tipical image processing techniques like smoothing and sharpening filters. All these techniques could be combined with respect to a specific application.

Following pre-processing, feature extraction task implies data transformation to get an image content descriptor based on its visual and semantic content. Visual content can be very *General* (color, texture, shape, spatial relationships, among others) or *Domain specific* (application dependent and may involve domain knowledge). A good visual content descriptor should be invariant to any accidental variance introduced by the imaging process. A visual content descriptor can be either *Global* or *Local*. Global descriptors use the whole image visual features, whereas Local descriptors use the visual features of *regions* or *objects* describing the image content. Usually, local descriptors imply dividing the image into tiles of equal size and shape. A simple partition does not generate perceptually meaningful regions but is a way of representing the global features of the image at a finer resolution. Some widely used techniques for extracting general content features are: Color Moments, Color Histograms, Color Coherence, Color Correlogram, Gabor Filter, Tamura features, Wavelet Transform, Moment Invariant, Turning Angles, among others [13, 24].

Semantic content could be obtained by textual annotation or by complex inference procedures based on visual content. We will not discuss in detail this topic, trying to focus on our subject.

Figure 2 shows an overview of a categorization process showing the different involved steps and their relationships.

Orthogonal to challenges of developing specific image mining algorithms and models that operate on idiosyncrassy of images, the major challenge for image mining is the pre-processing state previous to the extraction of relevant features. Pre-processing phase is necessary to improve the quality of the images and make the feature extraction phase more reliable. The pre-processing state is arguably the most complex phase of the knowledge discovery process when dealing with images. If the pre-processing is well done, it can be decisive whether patterns could be discovered, or whether the discovered patterns could be interpreted at all. This phase often requires related expertise to computer vision, image processing, image interpretation, graphics and signal processing, domain knowledge or domain applications, most of them constituting

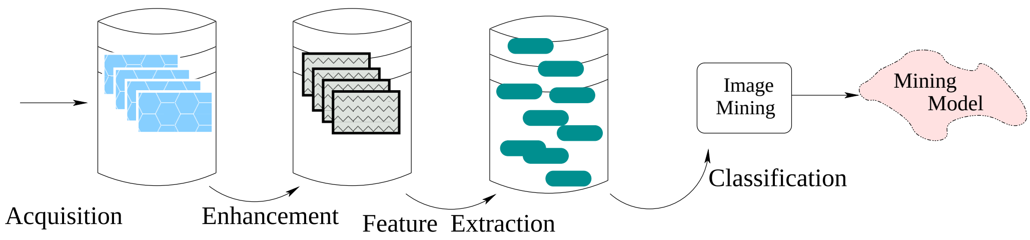


Figure 2: Image categorization process

natural sources of parallelism [28].

3 PARALLELISM INTO AN IMS

An Image Mining System (IMS) can be very computationally demanding due to the large amount of data to be processed, the requied response time or the involved image processing algorithms complexity. Any parallel system requires dividing up the work so that processors can make useful progress towards a solution as fast as possible. The main challenge is: *how to divide the labor*. There are three tightly components to the work: *computation*, *data set access*, and *communication among processors* [2, 14]. Dividing up the computation to make it faster creates more communication and often more data set accesses as well. Finding the best parallel algorithm requires carefully balance of the three named issues.

An IMS could be accomplished by assorted parallel strategies. In [10, 11] some models were proposed varying from a Global Parallel Model thru a combination of different Local Parallel Models focusing on the pre-processing stage. Here we will concentrate on the advantages got from the refered work on the local parallel model considering parallelism depending on image processing tasks. At this level any parallel model proposed not depends directly from the mining model itself, whereas it depends directly from any image processing task involved at the processing phase [13]. Any high performance techniques could be applied not only on every single image processing function but also on any conjuction of them. The best solution should be to build a standard parallel image processing library that enables to make parallel processing at different combinations. This gives the opportunity to apply different parallel methods and paradigms.

4 SELECTING PROCESSING FUNCTIONS

How to represent an image for knowledge discovery is an important issue in IMS. Images are not regular data, they represent complex information. It is necessary to find an element that describes univocally an entire image. Generally, an image is represented with a feature vector, where each component represents one key characteristic. The feature vector must constitute a whole image descriptor. The vector size and the relationship between each feature and its computing time are critical, moreover when the image amount is big enough.

An image mining systems claims acting like a human visual system. Visual perception concerns the acquisition of knowledge -vision is fundamentally a cognitive activity, distinct from purely optical processes such as photographic ones-. At earlier stages, a human visual system is involved in identifying objects and in locating objects. These two pathways are often called the “*what*” system and the “*where*” system, respectively. As was stated by Livingstone and Hubel [20] processing information is done at separate neural pathways and related to different visual properties such as *color*, *shape*, *depth*, and *motion*. Color and shape properties apply for the "*what*" system in object identification and the depth and motion properties to the "*where*" system in object localization. Now it is abundantly clear that a great deal of visual processing takes place in parallel across different subregions of visual cortical areas and finally they get together somewhere in the brain.

A sort of standard image processing tasks are commonly used at processing phase, like image smoothing, histogramming, 2-D FFT calculation, local area histogram equalization, contextual statistical classification, clustering feature enhancement, among others [8]. There exist many algorithmic implementions for those tasks that could be done thru parallel solutions [3, 5, 9, 12, 15, 16]. Moreover, different techniques at different grain scale could be applied depending on the particular task [4, 28, 30].

These image processing tasks must be classified into the four types of information to be obtained, be done in parallel, and at the end, be combined to enable the construction of a descriptive feature vector at the feature extraction stage. In this paper, we will focus on the "*what*" system, trying to organize the most commonly used image processing tasks for the extraction of color and shape information, running them in parallel, and finally, combining them into a unique vector feature.

4.1 Color Feature

Color is a psicological fenomenom, a subjective experience depending on the combined characteristics of three elements: the light emitting source, the observed object and the observer' s visual system.

Color space is the benchmark used to define the colors of an image. There exist different ways to define a color space [17]. Thinking about RGB color space containing n colors (RGB^n), the color features of an image I in this color space could be described as $RGB^n(I) = \bigcup_{i=1}^{n}(r_i, g_i, b_i)$, where r_i, g_i, b_i are the ith color value of the component Red, Green and Blue, respectively.

There exist at the bibliography many ways to operate color characterisitcs of pixels in order to define an image descriptor [6, 19, 29]. All of them use an specific color space. The color space in conjunction with the selected color operation will strongly influenciate the final image descriptor. The most commonly used color treatment methods are: color histogram, color moments, color coherence, and color correlograms. Here we will advocate to the first two cases: color histogram and color moments.

4.1.1 Histogram

Color histogram is the color's statistical information of an image at a given color space. In RGB^n color space, let I be an $N \times M$ image, its color histogram is a vector as following $H(I) = (h_{c_1}(I), h_{c_2}(I), \ldots, h_{c_i}(I), \ldots, h_{c_n}(I))$. According to this definition, the frequency of each color is relevant to the given image size. Normalizing each component $h_{c_f}(I)$ of $H(I)$ by image size will eliminate the image size dependency on the image feature specification and $h_{c_i} \in [0, 1]$. Thus, color histogram is a discrete representation of image appearances and similarity between images can be expediently computed.

In most of the known color spaces, image colors are represented in the form of a 3D vector. In RGB^n color space, axis $R, G,$ and B have 256 discrete color values. Hence, the color histogram of an image is a $256 \times 256 \times 256$ dimensional vector in RGB^{256} color space. As it is well known, if the number of colors of an image is too large, it will exceeds human's visual capacity.

Thus, without sacrifycing the representation precision and taking advantage of human's visual capacity, the number of dimension of each axis can be reduced, i.e.: the axis R and G into 8 bins, and the axis B into 4 bins [26]. In this way, the former $256 \times 256 \times 256$ feature vector is reduced into $8 \times 8 \times 4$ feature vector.

In this work, we present both implementation of color histogram calculation: full color histogram (CH) and bins color histogram or Mean Value Histogram (MVH). In section 5 we show implementation details and some results.

4.1.2 Color Moments

Color moments are statistical measures that can be used to differentiate images based on their features of color. The basis of color moments lays in the assumption that the distribution of color in an image can be interpreted as a probability distribution. Probability distributions are characterized by a number of unique moments. If color in an image follows a certain probability distribution, then the moments of the distribution can be used as features to identify that image based on its color.

In [27] Stricker and Orengo used three central moments of an image's color distribution: *Mean* (μ), *Standard deviation* (σ) and *Skewness* (s). For color representacion here we will restrict ourselves to the HSV color scheme (Hue, Saturation and Value) attempting to describe perceptual color relationships more accurately than RGB color space. Moments are calculated for each one of these three channels in an image, then 9 moments characterize an image. Defining the ith color channel at the jth image pixel as p_{ij}, the three color moments can then be defined as:

$$\mu_i = \frac{1}{N}\sum_{j=1}^{N} p_{ij}, \qquad \sigma_i = (\frac{1}{N}\sum_{j=1}^{N}(p_{ij}-\mu_i)^2)^{\frac{1}{2}}, \qquad s_i = (\frac{1}{N}\sum_{j=1}^{N}(p_{ij}-\mu_i)^3)^{\frac{1}{3}}$$

where N is the number of pixels in the image. In section 5 we show some implementation details.

4.2 Shape Feature

Edges characterize object boundaries and are visually important to human beings in recognizing or perceiving the shape of objects. Edge detection is a problem of fundamental importance in image analysis, therefore useful for segmentation, registration, and identification of objects in a scene. An edge is a jump in intensity and an ideal edge is a descontinuity. An edge detection strategy can be expressed in terms of the derivatives of the continous image $I(x, y)$. In practice, finite difference approximations of first and second order directional derivatives are used and represented by dicrete masks formally embodying linear-phase FIR filters that convolve with the image. Recovering of edges, is obtained by segmenting the convolved image with a global or local threshold operator. The perceived quality of the edge detector is determined by the selected threshold value. In selecting an appropriate threshold value, it is useful to consider the cumulative histogram of the convolved image.

Substantial research has been made on image processing methods on edge detection [13]. Here we will analize the most common first-order edge operators (Robert, Prewit, and Sobel) and second-order edge operator (Laplacian of a Gaussian). In section 5 we show implementation details and some results for the first-order edge operators.

4.2.1 Robert Operator

Roberts' Cross operator is one of the earliest algorithms. This can be accomplished by convolving the image with two 2×2 kernels (Fig. 3). Roberts' Cross is still in use due to the speed of computation, but performance compared to the alternatives is poor, with noise sensitivity being a significant problem.

(a)

0	1
-1	0

(b)

1	0
0	-1

Figure 3: Robert's masks.

4.2.2 Prewit Operator

Prewitt operator uses two 3×3 kernels (Fig. 4) which are convolved with the original image to calculate approximations of the derivatives - one for horizontal changes, and one for vertical. Although differential gradient edge detection needs a rather time-consuming calculation to estimate the orientation from the magnitudes in the x- and y-directions, the Prewitt edge detection obtains the orientation directly from the kernel with the maximum response.

(a)

-1	-1	-1
0	0	0
1	1	1

(b)

-1	0	1
-1	0	1
-1	0	1

Figure 4: Prewit's masks.

4.2.3 Sobel Operator

Sobel operator is based on convolving the image with a small, separable, and integer valued filters in horizontal and vertical (Fig. 5). Each Sobel edge mask is a combination of a digital differentiator in one of the spatial directions and a smoothing operator in the other. The result therefore shows how "abruptly" or "smoothly" the image changes at that point, and therefore how likely that part of the image represents an edge, as well as how that edge is likely to be oriented.

(a)

-1	-2	-1
0	0	0
1	2	1

(b)

-1	0	1
-2	0	2
-1	0	1

Figure 5: Sobel's masks.

4.2.4 Laplacian of a Gaussian Operator

A zero-crossing edge operator ($LoG(x, y)$) originally proposed in [21]. In order to effectively detect intensity changes (edges), the operator needs to have two characteristics. First, it must be a differential operator, taking either a first or second spatial derivative of the image. Second, it should be capable of being tuned to act at any desired scale so that large filters can be used to detect blurry shadow edges, and small ones can be used to detect sharply focused fine detail in the image. The so-called Laplacian-of-Gaussian edge operator is a compound operator that combines by convolution ($*$) a smoothing operation (Gaussian-shaped filter, G) with a differentation operation (discrete Laplacian, ∇^2). Edges are identified by the location of zero crossings. The resulting operator is defined for every pixel (x, y) of the image I as follows:

(a)

0	-1	0
-1	4	-1
0	-1	0

(b)

1/16	1/8	1/16
1/8	1/4	1/8
1/16	1/8	1/16

Figure 6: (a) Laplacian, (b) Gaussian masks.

$$LoG(x, y) = \nabla^2(G(x, y) * I(x, y)) \qquad where \;\; \nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \quad and \;\; G(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$$

5 HIGH PERFORMANCE COMPUTING DETAILS

When talking about cluster of computers as parallel architecture, it is necessary to keep in mind their characteristics: distributed memory and high communication costs. The working methodology of the BSP computation model fits adequately to this architecture type [7].

Common properties for each processing function were implemented following the BSP computation model. Every solution (excepted color moments) involves a unique superstep. The resulting computations follow the SPMD paradigm (SIMD generalization) with a *worker pool* scheme with *coordinator* [2]. At each implementation, the incoming image is partitioned into stripes, one for each worker. The workers calculate the corresponding processing function for the assigned image slices. Finally, one worker takes the coordinator rol and joins the processed image slices.

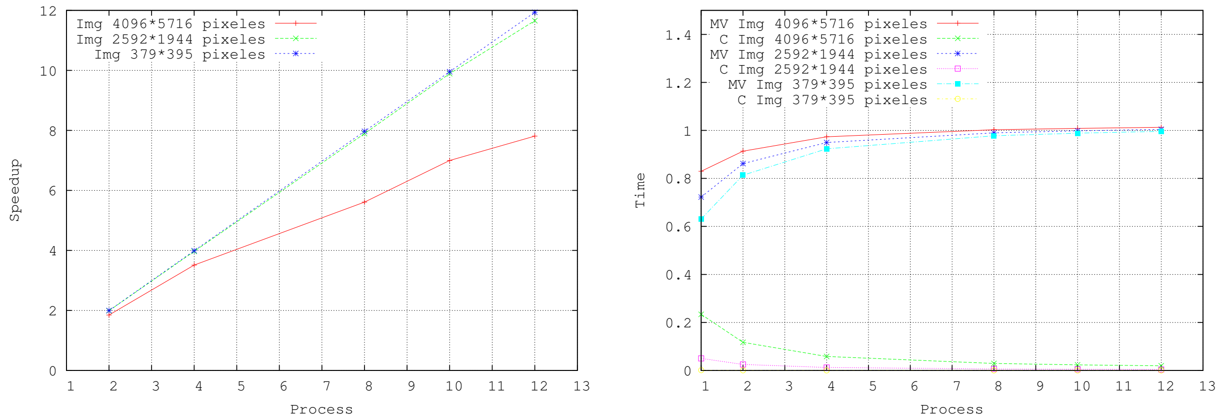
For color feature, the two presented versions of histogram calculus were parallelized: the whole color image space histogram ($256 \times 256 \times 256$) and the posterized image version ($8 \times 8 \times 4$). The second implementation could be thought as a combination of two processing steps, a posterization step and a histogram step.

Considering the characteristics of the problem in study, the high performance solution for color moments needs two supersteps. The first one takes care of RGB to HSV color space convertion and a partial mean calculus for each image channel at the corresponding image portion. The second superstep involves a partial standard deviation and skewness calculus. This solution use two kinds of communications types: an all-to-all communication reduction and one-to-one communication among workers and coordinator. At this moment, the parallel version is ready but some adjustment are been made for a better speedup tunning.

Figure 7: Original Image.

For shape feature, every edge detection parallel implementation must consider pixel's correlation to stablish image strip sizes. The calculus of the sizes of stripes must take into account overlapping because the idiosincrasy of the calculus. Figure 7 shows an example image of the testing image set. Testings were applied at three different sizes: small (379×395), medium (2592×1944) and large (4096×5716). Results were obtained from a 12 networked nodes cluster, each node consisting on a 32 GHz Pentium IV with 1Gb Ram. Nodes are connected together by Ethernet segments and Switch Linksys SRW 2024 of 1Gb. Base software on cluster includes a Debian Etch SO and MIPCH 2.1.0.6. Following images show the different speedup got from the referred implementations. Under the specifications and the solution stated for the full color histogram problem, any improvements are obtained by the parallel version. No matter the image size used, to settle different slice sizes for full color histogram calculation do not reduce the combination cost of local histogram slices.

Figure 8(a) shows the speedup obtained by the mean value histogram applied to the three tested image sizes. It could be observed high profits and the influence of the communications and data combination in respect to the image size. Figure 8(b) presents the relationships between the two calculated histogram times. Here it is represented the high computational cost got from the full color histogram on account of the not required visual color information



(a) MV Hist. at three image sizes

(b) MV Histogram versus C Histogram times

Figure 8: Speedup for Color feature.

given to the whole image mining system.

Figures 9 shows perfomance obtained by the different edge operators. It can be pointed out that better results are gotten by those operators that require more computational calculus (See Fig. 9(c) and 9(d)), at figure 9(d) in spite of small image sizes good performance is obtained, here communications are hidden by the amount of calculus.

6 CONCLUSIONS

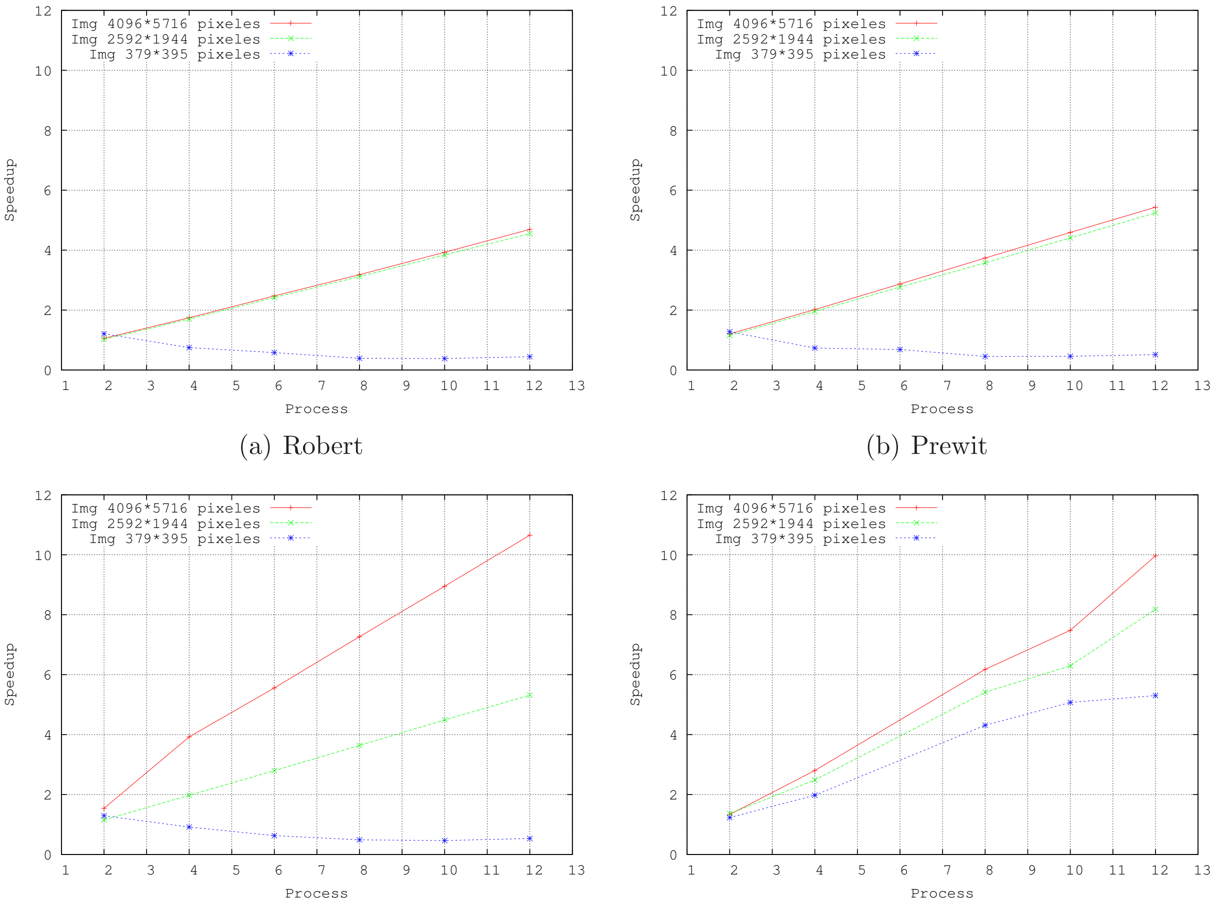
Here we have sketched some image mining system concepts enphasysing at the processing phase and showed how the human visual processing system is related to this processing phase.

A small parallel image processing function set is presented, the selected functions are among the most commonly used to simulate the human visual processing system in an isolate or combined form. At this stage, high performance solutions for the basic features of color and shape were developed, and some benefits were verified from these high performance implementations.

We had illustated that if distributed memory and high communication costs characteristics are allways kept in mind, high performance computing techniques can be applied into a cluster of computers. They will contribute to accelerate the automatic feature extraction and vector specification process, this last is a crucial one for a good IMS. According to this, any IMS could, in an automatic and faster way, extract features to build a vector representation that can be used then to compute similarities among images with an efficient resource use and in a lower cost and time.

Moreover, feature vector specification will involve a final cost equivalent to the maximun of the individual feature calculus cost, plus the communication cost among every coordinator process and the global coordinator. Some considerations about load balancing should be made at this final scheme.

Future studies will be oriented to extend the high performance image processing function set and to stablish a good vector represention with them. Besides, some studies could be oriented to apply high performance computation techniques to the remaining IMS stages.



(a) Robert (b) Prewit

(c) Sobel (d) Laplacian of a Gaussian

Figure 9: Speedup for Edge feature.

REFERENCES

[1] H. Krawczyk A. Mazurkiewicz. A parallel environment for image data mining. In *Proceedings of the International Conference on Parallel Computing in Electrical Engineering (PARELEC'02)*, 2002.

[2] A.Grama, A. Gupta, G. Karypis, and V. Kumar. *Introduction to Parallel Computing.* Addison Wesley, 2003.

[3] D. Ballard and C. Brown. *Computer Vission.* Prentice Hall, Englewood Cliffs, 1982.

[4] J. Barbosa and J. Tavares A. Padilha. Parallel image processing system on a cluster of personal computers. *Lecture Notes In Computer Science*, pages 439 – 452, 2000.

[5] S. Beucher and F. Meyer. The morphological approach to segmentation: the watershed transformation. *Mathematical morphology in image processing*, pages 433–481, 1993.

[6] M. Burl. Mining for image content, 1999.

[7] T. Cheatham, A. Fahmy, D.C. Stefanescu, and L.G. Valiant. Bulk synchronous parallel computing - a paradigm for transportable software. In *Proc. 28th Hawaii International Conference on System Science*, Jan. 1995.

[8] A. Choudhary and S. Ranka. Parallel processing for computer vision and image understanding. *IEEE Computer*, 25(2):7–9, 1992.

[9] J. Crespo, J. Serra, and R. Schafer. Theoretical aspects of morphological filters by reconstruction. *Signal Processing*, 2(47):201–225, 1995.

[10] J. Fernandez, N. Miranda, R. Guerrero, and F. Piccoli. Analyzing parallelism into an image mining system. In *6th International Information and Telecommunication Technologies Symposium (I2TS2007) IEEE*, pages 95–102, Braslia, DF - Brazil, December 2007.

[11] J. Fernandez, N. Miranda, R. Guerrero, and F. Piccoli. Towards a parallel image mining system. In *XIII Congreso Argentino de Ciencias de la Computacion (CACIC 2007)*, pages 787–797, Corrientes y Chaco, October 2007.

[12] C. Giardina and E. Dougherty. *Morphological Methods in Image and Signal Processing.* Prentice Hall, 1988.

[13] R. Gonzalez and R. Woods. *Digital Image Processing, 2nd Edition.* Prentice Hall, 2002.

[14] M. Gou and L. Yang. *High-Performance Computing: Paradigms and Infraestructure.* Wiley-InterScience, 2006.

[15] B. Jahne. *Digital Image Processing: Concepts, Algorithms, and Scientific Applications.* Springer Verlag, 1997.

[16] A. Jain. *Fundamentals of Digital Image Processing.* Prentice Hall, 1989.

[17] J.D.Foley, Van A.Dam, and J.F. Hughes. *Computer Graphics Principles and Practice.* Adisson Wesley Publishing Company, 1990.

[18] Y. Keiji. Managing images: Generic image classification using visual knowledge on the web. In *Proceedings of the eleventh ACM international conference on Multimedia*, pages 167–176, November 2003.

[19] K. Lee and W.N. Street. Automatic feature mining for personalized digital image retrieval. In ACM, editor, *Proceedings of the International Workshop on Multimedia Data Mining (MDM/KDD 2001)*, pages 38–43, San Francisco, USA, August 2001. ACM.

[20] M. S. Livingstone and D. Hubel. Segregation of form, color, movement, and depth: Anatomy, physiology, and perception. *Science*, 240(4853):740–749, 1988.

[21] D. Marr and E.Hildreth. Theory of edge detection. In *Proceedings of the Royal Society London*, number 207, pages 187–217, 1980.

[22] R. Missaoui and R. Palenichka. Effective image and video mining: an overview of model-based approaches. In *MDM '05: Proceedings of the 6th international workshop on Multimedia data mining*, pages 43–52, New York, NY, USA, 2005. ACM Press.

[23] H. Orallo, R. Quintana, and F. Ramirez. *Introduccion a la Mineria de Datos.* Prentice Hall, 2004.

[24] J. Parker. *Algorithms for Image Processing and Computer Vision.* J. Wiley & Sons, 1997.

[25] A. Selim, K. Krzysztof, T. Carsten, and M. Giovanni. Interactive training of advanced classifiers for mining remote sensing image archives. In *Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 773–782, 2004. (Industry/government track posters).

[26] Qin-Bao Song, Xiang-Rong Yang, Jun-Yi Shen, and Li-Ming Chen. Integrating color and spatial information for content-based image retrieval in large image database. *Machine Learning and Cybernetics, 2002. Proceedings. 2002 International Conference on*, 4:2082–2086 vol.4, Nov. 2002.

[27] M. Stricker and M. Orengo. Similarity of color images. In *Image and Video Databases III*, volume 2420, pages 381–392. SPIE Conference on Storage and Retrieval, Feb. 1995.

[28] W. Rapf M. ReinhardtL T. Braunl, S. Feyrer. *Parallel Image Processing.* Prentice Hall, Englewood Cliffs, Berlin Heidelberg, 2001.

[29] K.L. Tan, B.C. Ooi, and C.Y. Yee. An evaluation of color-spatial retrieval techniques for large image databases. *Multimedia Tools and Applications*, 14(1):55–78, 2001.

[30] M. A. Vorontsov. Parallel image processing based on an evolution equation with anisotropic gain: integrated optoelectronic architectures. *Optical Society of America*, (16):1623–1637, 1999.

[31] Y. Wang, F. Makedon, J. Ford, L. Shen, and D. Goldin. Image and video digital libraries: Generating fuzzy semantic metadata describing spatial relations from images using the r-histogram. In *Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries*, pages 202–211, 2004.