Asociación Argentina
de Mecánica Computacional

Mecánica Computacional Vol XXVIII, págs. 227-240 (artículo completo)
Cristian García Bauza, Pablo Lotito, Lisandro Parente, Marcelo Vénere (Eds.)
Tandil, Argentina, 3-6 Noviembre 2009

MULTI-LEVEL PARALELISM IN IMAGE IDENTIFICATION

J. Fernández[a], R. Guerrero[a], N. Miranda[a] and F. Piccoli[a]

[a] *Universidad Nacional de San Luis, Ejército de los Andes 950, 5700 - San Luis - Argentina, e-mail: {jmfer, rag, ncmiran, mpiccoli}@unsl.edu.ar*

Keywords: Image Processing, Feature Vector, Parallel Computation, Parallel Paradigm, Nested Parallelism.

Abstract. Images represent complex visual information. From a human point of view, image information extraction is merely a global process: starting with usually short, local and bad quality information, our brain can identify basic compositing elements and their relationship, elaborate a coherent sensorial experience and finally get the required information.

From a computational point of view, images clearly represent non structured information. Getting information from them involves to simulate human visual perception. At early stages, a human visual system is involved in identifying objects (the "*what*" system) and in locating them (the "*where*" system). The two systems can be simulated by looking for visual cues such as *color*, *shape*, *depth*, and *motion*. The combined cues lead to a unique image visual content representation. The most common image representation is a feature vector, where each vector component represents an image feature.

At this work we proposed a parallel feature vector construction following the precepts stated by the "*what*" and "*where*" system and by using high performance computing. The problem enables the application of multiple levels of parallelism and different paradigms combination. Finally some preliminary results are given.

1 INTRODUCTION

Visual perception is the process of acquiring knowledge about environmental objects and events by extracting information from the light they emit or reflect. Visual perception concerns the acquisition of knowledge - that is: vision is fundamentally a cognitive activity, distinct from purely optical processes such as photographic ones -. Image recognition area studies the different methods and techniques of capturing visual content of images for indexing and retrieval.

Marr (1982)'s theory supports that in the early stages of the vision process, there are cells that respond to stimulus of primitive shapes, such as corners, edges, bars, etc.. Young (1987) models these cells by using Gaussian derivative functions. Riesenhuber and Poggio (1999) propose a model for simulating the behavior of the Human Visual System (HVS), at the early stages of vision process.

At early stages, a human visual system is involved in identifying objects and in locating them. These two pathways are often called the "*what*" system and the "*where*" system, respectively. As was stated by Livingstone and Hubel (1988) information processing is done at separate neural pathways and related to different visual properties such as *color*, *shape*, *depth*, and *motion*. Color and shape properties may project to the "*what*" system for object identification and the depth and motion properties to the "*where*" system for object localization. Now, it is 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.

In image recognition, an image is firstly converted to a reduced format called a feature vector, then being recognized using some sort of classifiers. Therefore, extracting only the essential information contained in the original image and representing it as a feature vector is the most important processing in image recognition. In Adachi and Shibata (2001) an edge-based feature vector representation algorithm was proposed and has been successfully applied to medical radiograph analysis as well as to handwritten pattern recognition. Other representations also based on the edge information have been proven very robust in face detection Pan et al. (2003).

Other approaches use different local image features in the field of object categorization in images. Serre et al. (2005) use local features based on filter responses to describe objects, achieving a high performance in the problem of object categorization. On the other hand, different approaches using grey-scale image patches, extracted from regions of interest, to represent parts of objects has been suggested, Li et al. (2004), Agarwal et al. (2004), Leibe (2004). At the moment, all of the approaches are specific application oriented and there is not a clear advantage for general application problems.

The aim of this work is to carry out an experimental study in order to propose a set of different multi-scale filters bank that process local features of an image based on the *what* and *where* system. All the processing is oriented to the generation of a robust feature vector for visual image categorization by using a general, non-parametric and simple approach.

In section 2 of this paper, we review the feature vector construction and introduce the proposed filter functions associated with extracted image features. Section 3 analyses the system parallel characteristics. Section 4, describes the experiments and presents preliminary results. Finally, in section 5, a summary and some conclusions are sketched.

2 IMAGE FEATURE VECTOR

Images are an alternative form of communication and the most popular medium of information on the Internet. Feature extraction is the process by which image features are extracted and used to represent concisely the image visual content. The issue of choosing the features to be extracted should be guided by certain concerns like:

- The features should carry enough information about the image and should not require any domain-specific knowledge for their extraction.
- They should be easy to compute in order for the approach to be feasible for a large image collection and rapid retrieval.
- They should relate well with the human perceptual characteristics since users will finally determine the suitability of the retrieved images.

A sort of standard image processing tasks are commonly used at feature extraction stage, like image smoothing, histogramming, 2-D FFT calculation, local area histogram equalization, contextual statistical classification, scene segmentation, clustering feature enhancement, among others Choudhary and Ranka (1992). Because of perception subjectivity, there not exist a single best representation for a feature and consequently there not exist a unique representation for a feature vector. In addition, there not exist a unique way of feature extraction by processing and usually it is computationally expensive.

There exist many algorithmic implementions for those extraction tasks that could be done thru parallel solutions Ballard and Brown (1982); Giardina and Dougherty (1988); Jain (1989); Beucher and Meyer (1993); Crespo et al. (1995); Jahne (1997). Moreover, different techniques at different grain scale could be applied depending on the particular task Vorontsov (1999); Barbosa and Padilha (2000); T. Braunl (2001).

These image processing tasks could be classified according to the information to be obtained, they could be done in parallel, and at the end, be combined to enable the construction of a descriptive feature vector at the feature extraction stage. At the following sections, we will focus on trying to organize the most representive image processing tasks for the *what* and where *system*. In particular, we will focus on the extraction tasks and representation of color, shape, texture, and geometric projection information, running them in parallel, and at last, combining them into a unique system for the construction of the image feature vector.

2.1 The What System

The human ventral stream visual pathway, commonly referred as the "*What*" stream, is involved in the recognition, identification and categorization of visual stimuli. The ventral stream does not merely provide a description of the elements in the visual world, but it plays a crucial role in judging the significance of these elements. Visual cues for the *what* system are: color, shape and texture.

2.1.1 Color Feature

Color is a psicological fenomenom, a subjective experience depending on the combined characteristics of three element: 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 J.D.Foley et al. (1990). Thinking about RGB color space containing n colors (RGB^n), the color features of an image I in this color space could be describe 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 Burl (1999); Lee and Street (2001); Tan et al. (2001). 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, because they offer good processing times with parallel characteristics and complement each other giving a robust image color representation.

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

$$H(I) = (h_{c_1}(I), h_{c_2}(I), \ldots, h_{c_i}(I), \ldots, h_{c_n}(I))$$

where, $c_i = (r_i, g_i, b_i) \in RGB^n(I)$, and the component $h_{c_i}(I)$ is the number of pixels of a given color c_i in the image I and $\sum_{j=1}^{n} h_{c_j}(I) = N \times M$

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 sacrificing representation precision and taking advantage of human's visual capacity, the number of dimension of each axis can be reduced by using groups of colors. In a general context, groups of data are called *classes*, but in the context of a histogram they are known as *bins*, because one can think of them as containers that accumulate data and "fill up" at a rate equal to the frequency of that data class. As a result and according with the corresponding spectral response function for the R, G, B channels, axis R and G are grouped into 8 bins, and the axis B into 4 bins. In this way, the former $256 \times 256 \times 256$ feature vector is reduced into a $8 \times 8 \times 4$ feature vector. At this work, we present this last representation named bins color histogram or Mean Value Histogram (MVH).

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 an image color follows a certain probability distribution, then the distribution moments can be used as features to identify that image based on its color.

Stricker and Orengo (1995) used three central moments of an image's color distribution: *Mean*, *Standard deviation* and *Skewness*. 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.

2.1.2 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 $F(x, y)$. 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 for edge detection Gonzalez and Woods (2002). In Fernandez et al. (2008) we analized the most common first-order edge operators (Robert, Prewit, and Sobel) and here we will advocate to a second-order edge operator (Laplacian of a Gaussian) because it robustness in capturing important intensity changes filtering out information that may be regarded as less relevant, while preserving the important structural properties of an image.

Laplacian of a Gaussian Operator

It is a zero-crossing edge operator ($LoG(x, y)$) originally proposed by Marr and E.Hildreth (1980). 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 details in the image. The so-called Laplacian-of-Gaussian edge operator is a compound operator that combines 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 stated by the following formulae and implemented by the composition of the two kernels of Figure 1.

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

(a)

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

(b)

Figure 1: Masks: (a) Laplacian, (b) Gaussian.

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

2.1.3 Texture Feature

A main drawback in texture image analysis is the lack of proper definition for texture. A number of definitions have been proposed and used by various researchers in their context of analysis. Few schemes for texture analysis have been suggested. They are statistical, structural and spectral approaches. Using statistical approaches, several schemes have been suggested right from *co-occurence matrix*, *run length matrix based*, *auto correlation*, *auto regression*, *MRF based*, *moments based*, etc. as found in literature Chen (1988); Wang et al. (2004). The most common is *radius* and *angular histogram* features obtained by computing Discrete Fourier Transform of the texture image signal for content based image retrieval.

In structural approach, the texture region is defined to have a constant texture if a set of local statistics or other local properties of the image are constant, slowly varying or approximately periodic. An image texture is described by the number and types of its (tonal) primitives and the spatial organization or layout of the primitives. Textures could be rated as coarse, micro, macro, regular, periodic, aperiodic, directional, random, or stochastic Tuceryan and Jain (1998). Texture images are analyzed by identifying the local and global properties of the images under consideration.

Very few schemes have been suggested which attempt to propose for both local and global descriptors. One such method is texture spectrum scheme Stefanou and Kerekes (2009). Gabor filters have been recently used for the texture analysis at different tuning parameters Ilonen et al. (2008); Kong (2009). Wavelet based features have been successfully employed for the texture classification Fischer et al. (2007). A new statistical model based wavelet domain single value decomposition for image texture classification has been described Huang and Aviyente (2008). Other recent approaches for texture analysis and synthesis may be referred from Dong et al. (2008); Li et al. (2009).

Each method discussed above can be chosen based on the application and the problem domain. At this paper two methods corresponding to two different types of the named methods will be used into a general feature vector specification: Radius and Angular Histogram by Direct Fourier Transform calculus (statistical) and Gabor filters (spectrum scheme).

Discrete Fourier Transform

This operator calculates texture features based on image power spectral in the frequency domain. Let $F(u, v)$ the Fourier frequency spectrum of an image signal and the corresponding associated characteristics of Magnitude ($|F(u, v)|$), Phase ($|\phi(u, v)|$) and Power Spectrum ($|F(u, v)|^2$). Texture features will be the Power Spectral Radius ρ and the Power Spectral Angle θ of the image's power spectral representation of the Fourier transform obtained after converting $F(u, v)$ to polar coordinates.

Gabor Filter Operator

A filter design-based approach has been of interest in modeling the processing by the visual cortex. The Gabor model is one of the most widely used in human visual system modeling at the visual cortex level Rosin (2006). This operator is based on the premise that texture is an image pattern containing a repetitive structure that can be effectively characterized in a frequency domain, such as the Fourier domain. However, one of the challenges of such an approach is dealing with the tradeoff between the joint uncertainty in the space and frequency domains. Meaningful frequency based analysis cannot be localized without bound. An attractive mathematical property of Gabor functions is that they minimize the joint uncertainty in space and frequency. In Kong (2009) they achieve the optimal tradeoff between localizing the analysis in the spatial and frequency domains. Texture analysis is accomplished by applying a bank of scale and orientation selective Gabor filters to an image. These filters are constructed by using a two-dimensional Gabor function $g(x, y)$ and its Fourier transform $G(u, v)$ written as:

$$g(x,y) = \left(\frac{1}{2\pi\sigma_x\sigma_y}\right) e^{-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}+2\pi i W x\right)}$$

$$G(x,y) = e^{-\frac{1}{2}\left(\frac{(u-W)^2}{\sigma_u^2}+\frac{v^2}{\sigma_v^2}\right)}$$

2.2 The *Where* System

The "*Where*" system is the mental process at work as a person sees with two normal eyes. It is related with depth perception. In monocular images depth estimation, if no particular prior knowledge of the scene is given, can be retrieved in many real images by the monocular cues. Several strong monocular cues allow relative distance and depth to be judged. These monocular cues include: *Relative size*, *Occlusion* (also referred to as *Interposition*), *Linear Perspective*, *Aerial perspective*, *Accommodation* (also referred as *Focus*), *Motion parallax*, *Depth from motion*, *Familiar size*, *Peripheral vision* and *Texture gradient*.

At this work we started incorporating the "*where*" system to the feature vector by the addition of the linear pespective feature to its specification.

2.2.1 Linear Perspective

When objects of known distance subtend a smaller and smaller angle, it is interpreted as being further away. Among others, vanishing points can also refer to perspective in a perspective drawing. A vanishing point is a point in the distance to which parallel lines of objects appear to converge (such as roads and railway lines).

There exist different techniques that try to estimate vanishing point from still images Criminisi et al. (2000); Seo et al. (2006); Marquis-Bolduc et al. (2008).

The technique implemented in Cantoni et al. (2001) estimates the position of vanishing points in an image, based on a deterministic analysis. It works directly in the image plane, which is used as an accumulation space for a particular version of the Hough Transform.

The algorithm is composed of the following steps. First, an edge detection operation is performed on the original image through an isotropic operator, which is composed of two 3×3 masks (one for the horizontal and one for the vertical component - Sobel as an example). The result image is then normalized and thresholded against a very high value, to eliminate redundant information. This way, a binary image is obtained. For each point P within it, the tangent is calculated, by means of the following expresion:

$$m = \tan\theta = \frac{D_y}{D_x} \tag{1}$$

where D_x and D_y are the x and y components resulting from the application of the isotropic operator in the point. Now that θ is known, it can be used to draw a straight line, with slope m, passing through P. Such line is then accumulated in the (x, y) parameter space. At the end of the process, those points having the greatest numbers of votes will be marked as candidates.

2.3 Feature Vector Composition

Color, shape, texture and geometric properties associated values will set the feature vector represention of visual content of images for indexing and retrieval. This feature vector structure must lead to an unique image identification. The following table shows the resulting feature vector structure with its vector components associated values:

Feature Descriptions	Num. of data
Color	
. RGB values: mean	3
. HSV values: conversion from RGB values	3
. Hue histogram (4 bins) and entropy	5
Shape	
. Normalized x and y pixels: mean	2
. Number of detected edge pixels	1
Texture	
. Power spectral radius and angle values	2
. Gabor transform value: mean	1
3D Geometry	
. Detected vanishing point: x and y values	2

3 OPTIMIZING FEATURE VECTOR CALCULUS

Representing the image visual content through a feature vector implies not only the issue of choosing the appropiate image features but to extract them. Considering the processing time and resources involved, feature extraction is an expensive process that should be expected to be done in a fast way. The use of different optimal performing computational techniques could help to reduce processing costs.

Most selected image feature extraction methods could be performed in an independent way and do not require an specific processing order. Moreover, each feature calculus could enable the use of different optimal computing techniques going from traditional parallel ones to high performance like cloud, grid and GPU computing Foster and Kesselman (2004); Buck (2000); Chen and Hang (2008); Beard (2008).

A posible solution to the feature extraction problem is the application of two different levels of parallelism: a macro level, corresponding to the calculus of each vector component visual feature, and a micro level, stablished by the singularities of each calculated feature, that could be resolved in multiple levels of parallelism. Figure 2 shows graphically the global system structure for the feature vector calculus.

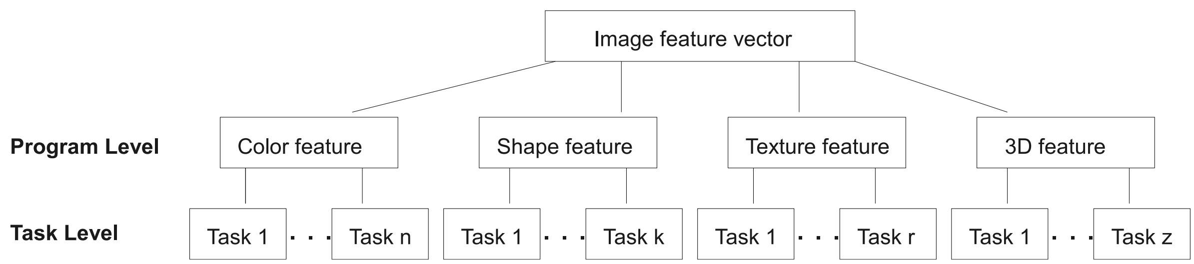


Figure 2: Parallel processing Levels

This macro level can be considered a program level parallelism, it is characterized by parallel execution of different programs that are totally independent units without common data (every one works over image) and with little or no communication and synchronization needs. By other side, the micro level can be considered task level parallelism, the program can be partitioned into a number of parallel executable tasks. Tasks have to be synchronized, and they usually communicate with each other. Tasks can be the same, but working over different data.

The first level is an MIMD parallelism, all features are calculated at the same time in a parallel way, keeping independence and not sharing data. Each process has an image copy, with no communication with others, and data are collected at the end of the labor by an specific role process. This structure applies directly to a *farm* parallel programming structure.

The second level depends on the specific feature properties to be calculated. At present work, solutions could be implemented in an SIMD parallelism or, in a more general way, in SPMD parallelism while no consideration are taking over feature particularities.

Process organization for feature vector calculus is associated to the concept of group. Processes are divided into different groups depending on features to be extracted. Each group extracts an specific feature and every process into the group works in parallel. Every group has a group-with-coordinator structure without considering the specific feature.

In order to get architecture independence, the working methodology of the BSP computational model fits adequately to any architecture type, including cluster among them. In consequence, every feature calculus parallel version was implemented following the BSP computational model.

Image, as the input data, is partitioned into stripes, one for each worker. Workers calculate the corresponding processing function for the assigned image slices. Finally, one worker takes the coordinator rol and joins the processed image slices.

3.1 Feature Calculus Analysis

In color feature calculus, the implemented histogram version (posterized image - $8\times8\times4$) could be thought as a combination of two processing steps, a posterization step and a histogram step. Because of their particularities, color moments calculus 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 communication types: an all-to-all communication reduction and a one-to-one communication among workers and coordinator. At this moment, the parallel version is ready but some adjustments have to be made for a better speedup tunning.

In shape feature calculus, the Laplacian-of-a-Gaussian filter is simple combined mask filter implementation. Thus it could done in an embarrasly parallel way, every edge detection parallel implementation must consider pixel's correlation to stablish image strip sizes. Calculus of strips sizes must takes into account overlapping because calculus singularity.

In texture feature calculus, the statistical implementation (Fourier Transform and Power Spectrum) enables de use of two parallel levels. At the first one there exist three task in parallel, every one concerning an image plane (R, G, B) leading to a three group organization for the processes. At each group, processes split the FT calculus into a row FT calculus, follow by the column FT calculus and finally the Power Spectrum calculus. At the resulting system, texture feature calculus will imply three levels of parallelism, one at the feature level, other at the image level, and the last one coming from FT calculus. Communication follows an all-to-one model for the second level, while level an all-to-all and all-to-one model for the third level.

On the other side, Gabor transform implemention involves two supersteps, one for the calculus of the Gabor function and the other for the calculus of its Fourier transform. Communication among processes and coordinator are all-to-all in the first superstep and all-to one in the second superstep.

For this work, linear perspective calculus had been implemented but results must be adjusted before valid results could be shown.

4 EXPERIMENTS

Experiences were applied on three different image sizes: small (128×128), medium (1024×1024) and large (4096×4096), and organized into two stages: at first, the calculus of optimal features in an independent way and then the calculus of the final feature vector composition. Results were obtained from a 18 networked nodes cluster, each node consisting on a 64 bits Intel E6750 de 2,66GHz processor with 2GB DDR2 667 Mhz Ram. Nodes are connected together by Ethernet segments and Switch Linksys SLM2048 of 1Gb. Base software on cluster includes a 64 bits Debian 5 Lenny Operating System and MIPCH 2.1.0.6.

Taking into account every vector component parallel calculus and cluster particularities the 18 processors were divided into 5 groups (4 groups with 3 processors and the remaining group with 6 processors) While the biggest group was working on FFT spectral calculus (2 levels of parallelism) the others were working on remaining features.

Figure 3(a) shows the speedup obtained on the Mean Value Histogram calculus applied to the three tested image sizes. It could be observed high profits and the influence of communications and data combination in respect to the image size. The same applies to figure 3(b).

Figure 3(c) shows performance obtained by Laplacian of a Gaussian operador. The size of images carried some weight on the obtained results. Besides the speedup is not close to lineal speedup, it is not bad.

Figures 3(d) and 3(e) show the speedup obtained through the application of texture filters. Besides both performances are similar, Gabor filter works better.

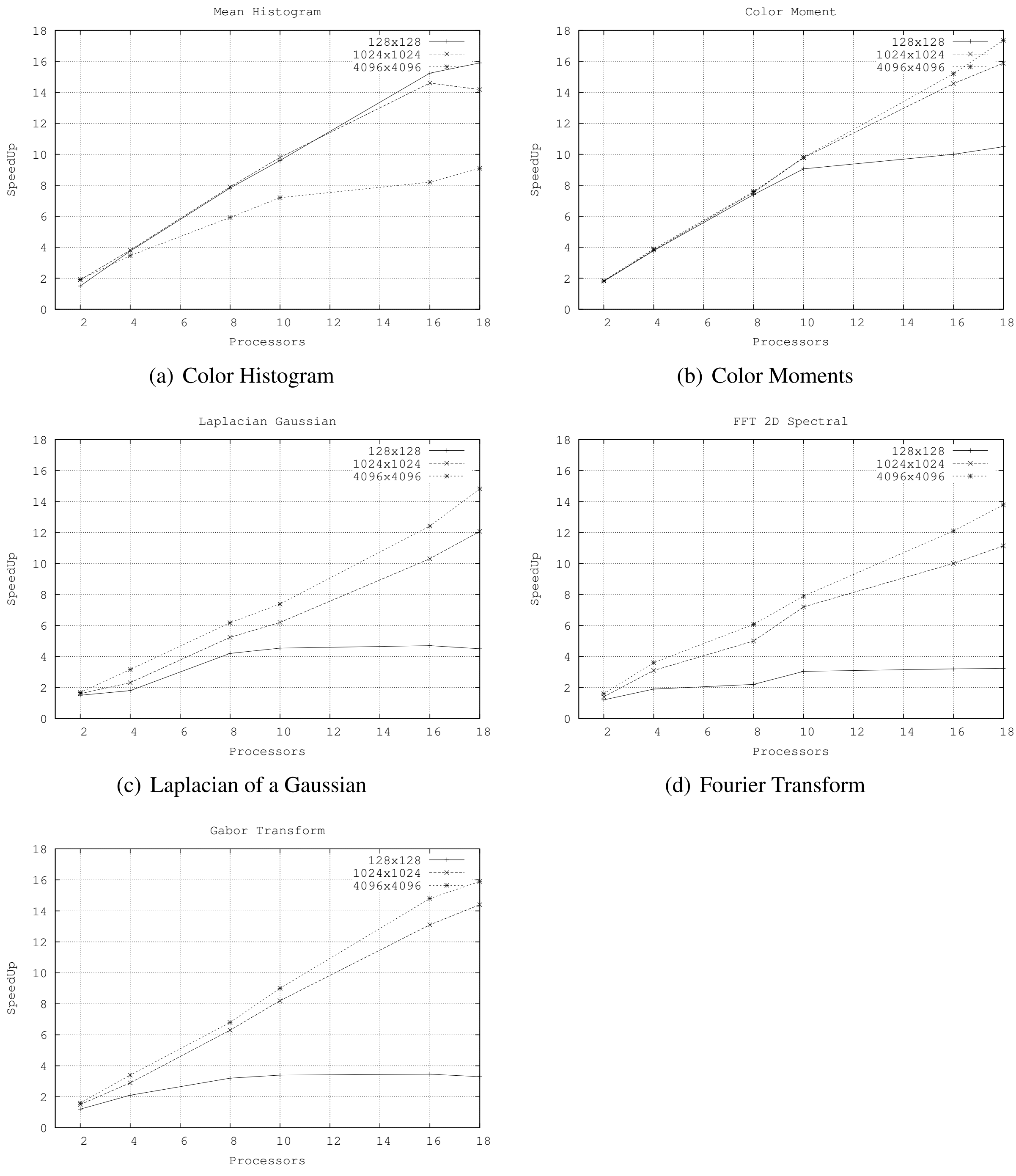
In general, results show some improvement only on medium and large image sizes (speedup: 3,21 and 3,36 respectively). This behavior comes from feature calculus cost disparity, where greater costs hide some acquired cost optimizations. Some improvements could be got through load balancing according with assigned work. For example, 2 groups with 6 processors working on FFT calculus and color moments, and 3 groups with 2 processors working on the remaining features. This arrangement could give some profits also on small images. Another arrangement is combining in group those less working cost features and processing them in parallel.

5 CONCLUSIONS

In this report, we have summarized the most relevant ideas for extracting essential visual information contained into an image and how to used them to conform a unique image visual content representation by means of a feature vector specification. This feature vector could be used later for indexing and retrieval. The selected features tried to follow the precepts stablished in section 2 (information relevance, computational simplicity and perceptual meaning).

We have given an overview of the used parallel filter applications and their computational speed. In all cases they got good profits inherited from feature properties in spite of communication and processing relationship.

Small changes in processors work distribution policy and task assignment had caused better results. A more detailed analysis would enable to get better results avoiding processors



(a) Color Histogram

(b) Color Moments

(c) Laplacian of a Gaussian

(d) Fourier Transform

(e) Gabor Transform

Figure 3: Speedup on Color, Shape and Texture calculus.

subutilization.

Our design and analysis show that image feature vector lends itself well to parallelization, extending the posibilities to multi-level parallelism. However, there are several issues that could benefit from further work on the parallel algorithm front. The parallel implementation demonstrated good speedup and performance at the feature vector calculus and gives some hints on communication and synchronization improvements. In addition, some advances will be to adjust the existing parallel implementation to dual and quad core processors and the corresponding architectures,

Finally, future studies will be oriented to enable portability to other architectures and high performance computation.

REFERENCES

Adachi M. and Shibata T. Image representation algorithm featuring human perception of similarity for hardware recognition systems. In H.R. Arabnia, editor, *Proc. of the Int. Conf. on Artificial Intelligence (IC-AI'2001)*, volume 1, pages 229–234. CSREA Press, Las Vegas, Nevada, USA, 2001. ISSN 1-892512-78-5.

Agarwal S., Awan A., and Roth D. Learning to detect objects in images via a sparse, part-based representation. In *IEEE PAMI*, volume 26, page 1475?1490. 2004.

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

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

Beard H. Cloud computing best practices for managing and measuring processes for on-demand computing. In E. Lulu.com., editor, *Applications and Data Centers in the Cloud with Slas*. 2008. ISSN 9781921523199.

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

Buck I. Gpu computing with nvidia cuda. In *ACM SIGGRAPH 2007 courses ACM*. New York, USA, 2000.

Burl M. Mining for image content. 1999.

Cantoni V., Cantoni V., Lombardi L., Porta M., and Sicard N. Vanishing point detection: Representation analysis and new approaches. In *Proceedings of the 11 th International Conference on Image Analysis & Processing*, pages 26–28. 2001.

Chen C.C. Markov random fields in image analysis,ph.d thesis. In M.S. University, editor, *Computer Science department*. East Lansing, 1988.

Chen W. and Hang H. H.264/avc motion estimation implementation on compute unified device architecture (cuda). In *IEEE International Conference on Multimedia and Expo 2008*, pages 697–700. 2008.

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

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

Criminisi A., Reid I., and Zisserman A. Single view metrology. *Int. J. Comput. Vision*, 40(2):123–148, 2000. ISSN 0920-5691. doi:http://dx.doi.org/10.1023/A:1026598000963.

Dong W., Zhou N., and Paul J.C. Perspective-aware texture analysis and synthesis. *Vis. Comput.*, 24(7):515–523, 2008. ISSN 0178-2789. doi:http://dx.doi.org/10.1007/s00371-008-0232-1.

Fernandez J., Miranda N., Guerrero R., and Piccoli F. A distributed computing for an image processing function set. In *XVII Congreso sobre Métodos Numéricos y sus Aplicaciones, ENIEF 2008*, pages 2895–2906. Univ. Nac. de San Luis. San Luis, 2008. ISSN 1666-6070.

Fischer S., Sroubek F., Perrinet L., Redondo R., and Cristobal G. Self-invertible 2d log-gabor wavelets. 75(2):231–246, 2007.

Foster I. and Kesselman C. The grid: blueprint for a new computing infrastructure. In E.M. Kaufmann, editor, *Elsevier series in grid computing*. I. Foster and C. Kesselman, 2004.

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

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

Huang K. and Aviyente S. Wavelet feature selection for image classification. 17(9):1709–1720, 2008.

Ilonen J., Kamarainen J., Paalanen P., Hamouz M., Kittler J., and Kalviainen H. Image feature localization by multiple hypothesis testing of gabor features. 17(3):311–325, 2008.

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

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

J.D.Foley, A.Dam V., and Hughes J. *Computer Graphics Principles and Practice*. Adisson Wesley Publishing Company, 1990.

Kong A. An analysis of gabor detection. pages 64–72. 2009.

Lee K. and Street W. 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. ACM, San Francisco, USA, 2001.

Leibe B. *Interleaved Object Categorization and Segmentation, PhD thesis*. ETH Zurich, 2004.

Li C., Wang J., Ye L., and Wang H. A novel method of dynamic textures analysis and synthesis. In *CSO '09: Proceedings of the 2009 International Joint Conference on Computational Sciences and Optimization (cso 2009)*, pages 328–332. IEEE Computer Society, Washington, DC, USA, 2009. ISBN 978-0-7695-3605-7. doi:http://dx.doi.org/10.1109/CSO.2009.381.

Li F., Fergus R., and Perona P. *Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories*. Merriam Webster, 2004.

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

Marquis-Bolduc M., Deschênes F., and Pan W. Combining apparent motion and perspective as visual cues for content-based camera motion indexing. *Pattern Recogn.*, 41(2):445–457, 2008. ISSN 0031-3203. doi:http://dx.doi.org/10.1016/j.patcog.2007.06.021.

Marr D. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. W. H. Freeman, 1982. ISBN-10: 0716715678.

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

Pan Q., Min-Gui Z., De-Long Z., Yong-Mei C., and Hong-Cai Z. Face recognition based on singular-value feature vectors. In *Optical engineering*, volume 42, pages 2368–2374. Society of Photo-Optical Instrumentation Engineers, Bellingham, 2003. ISSN 0091-3286.

Riesenhuber M. and Poggio T. Hierarchical models of object recognition in cortex. In *Nature Neuroscience*, volume 20, page 1019?1025. 1999.

Rosin P.L. Training cellular automata for imageprocessing. In *IEEE Transactions on Image Processing*, volume 15, pages 2076–2087. 2006.

Seo K.S., Lee J.H., and Choi H.M. An efficient detection of vanishing points using inverted coordinates image space. *Pattern Recogn. Lett.*, 27(2):102–108, 2006. ISSN 0167-8655. doi:http://dx.doi.org/10.1016/j.patrec.2005.07.011.

Serre T., Wolf L., and Poggio T. Object recognition with features inspired by visual cortex. In *IEEE CSC on CVPR*. 2005.

Stefanou M. and Kerekes J. A method for assessing spectral image utility. 47(6):1698–1706, 2009.

Stricker M. and Orengo M. Similarity of color images. pages 381–392. 1995.

T. Braunl S. Feyrer W.R.M.R. *Parallel Image Processing*. Prentice Hall, Englewood Cliffs, Berlin Heidelberg, 2001.

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

Tuceryan M. and Jain A.K. Texture analysis. In P.S.P.W.e. by C. H. Chen L. F. Pau, editor, *The Handbook of Pattern Recognition and Computer Vision (2nd Edition)*, World Scientific Publishing Co., pages 207–248. World Scientific Publishing Co., 1998.

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

Wang C., Yu-bin Y., Wu-jun L., and Shi-fu C. Image texture representation and retrieval based on power spectral histograms. In *Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2004)*. 2004.

Young R.A. The gaussian derivative model for spatial vision: I. retinal mechanisms. In *Spatial Vision*, volume 2, page 273?293. 1987.