IMPROVEMENT OF A PARALLEL SYSTEM FOR IMAGE PROCESSING

GUERRERO R., PICCOLI F., PRINTISTA M. *
Departamento de Informática
Universidad Nacional de San Luis
Ejército de los Andes 950
5700 - San Luis
Argentina
e-mail: {rag, mpiccoli, mprinti}@unsl.edu.ar

Abstract

Digital images are digital signals captured through different means. Sometimes these captured images contain variations combined with the original signal, this variations are called noise. Echo is a particular kind of noise with characteristics that turns it into a very interesting problem to solve. The echo detection process and its subsequent elimination from a digital image involves extensive mathematical calculations.

Differents parallel approaches taking advantage of new architectures can be implemented to solve this problem. Nevertheless these approaches have time depending characteristics, so the processing time is still the critical point. One improved version of a parallel one may be implemented by using a different algorithm and some other techniques that much more reduce the processing time.

In this work, the authors discuss their earlier work, the present approach and the future directions of this experimental application.

Finally the resulting values are sketched.

Keywords: Homomorphics Systems, Fourier Transform, Cepstrum, Convolution, Parallel models, System arquitecture.

*Group supported by the UNSL and ANPCYT (Agencia Nacional para la Promoción de la Ciencia y Tecnología)

1 INTRODUCTION

Toward the years digital images had turned into a natural communication media. The explotion in the use of digital images have triggered a corresponding explotion in image-processing developments. Image processing is intended on improve or enhance stored digital images captured from cameras, sensors, etc.[1][7]. Different image-processing techniques and algorithms have been proposed during the last years where the usefulness of one approach depends particularly of the problem [13].

Sometimes digital images contains "noise", unwanted variations added to the original signal. Echo is a particular kind of noise where the undesired signal is just the original signal that was *delayed* in the time of arrival and has suffered an *attenuation* in its intensity as a consequence of the delay. This is a very interesting problem because of the characteristic that the original signal is convoluted with the undesired signal. In the study about images with ghost (echo), the point is to get the two parameters previously stated and then restore the image by means of the use of those [12]. The problem can be treated and resolved mathematically through the use of an homomorphic system [3][12]. Computational approaches needs extensive repetitive calculations on large amounts of data to give valid results, then the execution time develops the critical point.

The parallel processing has become an important topic when the object is to increase the computational speed of a task. Images provide a natural source of parallelism; so image processing at low level have several characteristics that make it suitable for implementation with parallel computers[5][15]. In the last time, there is an incremental effort to develop platforms that make the parallel programming a more easy and practical job; this is the case of PVM. PVM enables an heterogeneous network of computers to be enrolled on a single problem by means of the use of message passing [6][16].

Since a parallel version of this problem on PVM is achieved readily with increases in the speed up, still some aspects of the whole process involves too much time in its processing depending not only on the programming language or the hardware but on the way they are treated.

When the processing implies to move data from the spatial domain trough the frecuency domain the use of the Fourier Transform becomes the more frequent way and, when the subject is to get speed up, a point to take under consideration. Because its characteristics the FT algorithms provides several aspects that can be refined by using differents techniques.

In this work, an improved version of a parallel algorithm for the detection and restoration of an image with ghost is presented. The results of the two parallel version and the sequential one are contrasted.

2 THE ECHO PROBLEM

Digital signal processing is concerned with the representation of the signals by sequences of numbers and its mathematical processing in a subsequent time. Although a signal can be represented mathematically like a function of one or more independent variables (system); the idea is to use a technique that transforms it into a new form which is in any sense more suitable for processing than the original. *Linear Shift-Invariant Systems* are an interesting class of systems that leads to elegant and powerful mathematical representations, allowing a variety of useful signal processing functions like the extraction of a signal parameters or the separation of signals that are combined by a particular superposition, the addition.

A digital image with ghost is a digital signal that is convoluted with itself, a special kind of superposition that is not considered by the linear shift-invariant systems, so a new class of systems that obey a generalized principle of superposition is established, the class of nonlinear systems called *Homomorphic Systems*[11]. Then the two parameters associated with the echo, the **delay** and the **attenuation**; can be obtained through these systems.

In 1963 Bogert, Healey and Tukey [3] observed that the logarithm of the power spectrum [4][10] of a signal containing an echo has an additive periodic component due to the echo, this characteristic was named *Cepstrum*; and thus the Fourier Transform of the Cepstrum should exhibit a peak at the echo delay.

The remaining parameter, the attenuation, can be obtained through the values that arise from applying the Fourier Transform to the Cepstrum [11]. Since the power spectrum is the fourier transform of the autocovariance function the values obtained at the end of the process are correlated between themselves and the number identifying them represents the correlation level. Then the attenuation will be the existing relation between the total energy of the image, second value observed, and the value of the peak. To do this, first the correlation between the value of the peak of the delay and the previous ones must minimized by subtracting the average of the previous values to the one of the peak and the result is the ratio between this and the total energy of the image[8].

Finally the whole restoration process will consist of six stages: to compute the Fourier Transform of the image, to calculate the Cepstrum, to compute the Fourier Transform of the Cepstrum, to obtain the delay (τ) and attenuation (α), to apply the Filter and then to compute the Inverse Fourier Transform to the filtered image (See Fig. 1).

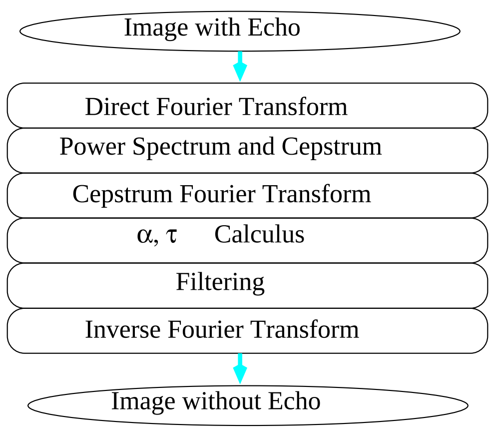

Figure 1: The Echo Restoration Stages

3 THE PARALLEL APPROACH

Splitting a sequential program into independents parallel units involves indentifying intermodule dependencies to determine where parallel implementation is safe. The original problem is a sequential process consisting of six stages (See Fig. 1) where the output of one stage is the input of the next one, and every level is in itself parallel. In addition, in each stage the problem is *embarrassingly parallel* [16] since the computation can be divided into independent parts (processes) that will be executed simultaneously requiring the same data and producing results from its inputs without any needs for results from the others parts.

Divide and Conquer technique applies very well to problems with this characteristics. However some interaction will be needed; at the end of every stage the results must be collected and combined in some way, which suggest that a single process will be operating alone to perform this operation. Furthermore, if at beginning of every stage this single process divides the task to be performed and iniciates the different processes which will execute the computations, the resulting structure will be a *Master/Slave organization* [14].

Since networked computers has become the more frequent choice for parallel processing, there is an increasing demand for tools that makes the programming much more easy and clear. The *Parallel Virtual Machine* (PVM) is one of these. This tool allows the developing of parallel programs independently of the hardware architecture enabling a collection of heterogeneous computer systems (Distributed Systems) to be viewed as a single parallel virtual machine.

4 THE ORIGINAL IMPLEMENTATION

The original implementation [9] was supported by a heterogeneous workstation network with NFS file system and where the processes are related in a *Crowd Computation model* on PVM [2].

The system architecture is showed in the Fig. 2.

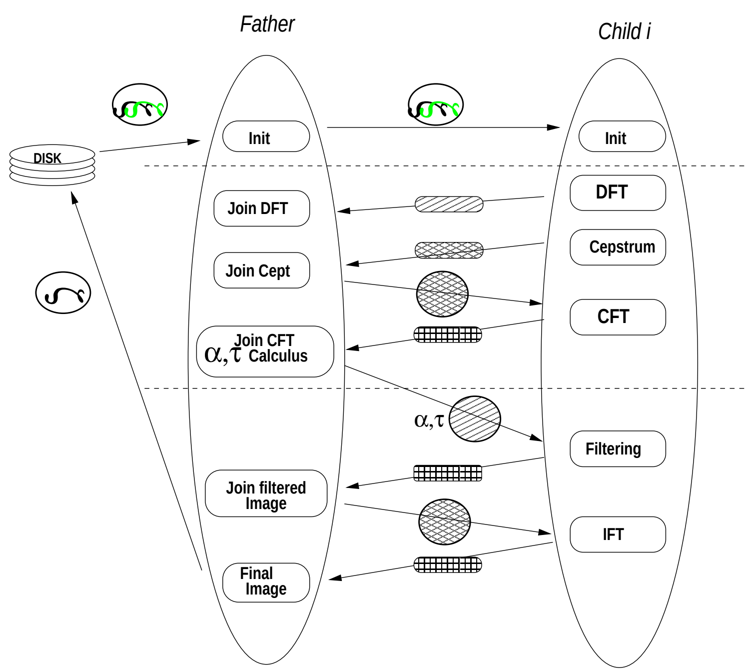

Figure 2: The Original Parallel Implementation

The master iniciate the slaves in a static way sending them the number of processes iniciated. The input data is a pixmap tipically held in a file and copied into an array in memory by each process. Every slave determines the region of the pixmap it must process by means of the calculation between its respective group identification and the number of processes iniciated. After each stage, every child process send their results to the father, which collects them and then iniciates the next stage.

There are two points (dash line in the Fig. 2) where is required that the independent processes be syncronized: at the beginning, because they must wait for every process get the original image from the father and, in the middle of the whole implementation, when the father must obtain the parameters that will be applied to restore the original image. These two steps can be implemented by a basic mechanism called *Barrier* [16]. All the processes, the master and the slaves will belong to a group in terms of that they can be syncronized.

The interaction among processes involved two communication types: from the father to childs by broadcasting and from the childs to the father by using a point to point communication.

Thus, while the process go on the original image with echo will be transformed into intermediate images as a consequence of applying the respective stage function.

5 PERFORMANCE EVALUATIONS

Many applications need the speedups that parallel processing offers. In applications with image processing this is often a critical bottleneck that affects system speed; the enormous size of images and their quantity make processing time a limiting factor in what can be done.

In the echo problem, the existing serial computational process is augmented with an additional processing to allow results to be computed in parallel and then combined. Despite the parallel implementation gets better computational times than the sequential one, an efficient use of this requires a thorough understanding of the application and the interactions between the involved subtasks. Particularly, the whole process involves the calculation of several FT. Although liner, it differs from common linear filter operations since every transformed output pixel depends on every pixel of the input image. The problem can be simplified somewhat.

There are several aspect about the FT calculus that not depends just only on the algorithm and programming language but on the characteristic of FT itself. The Discret Fourier Transform calculus implies many multiplications. Multiplications are the most time consuming operations on every computer, so a direct implementation of the DFT equations is costly in terms of the number of operations to be done. In general, because of the number of multiplications and substructions involved, may be expected the computational time to be roughly proportional to N^2, with N the number of input points.

Many of these time-consuming operations have an interesting characteristic: they are redundant. This is because the exponentials specified inside the equations are periodic, then the same products are calculated many times particularly with lengthy transforms.

As a solution, a number of highly efficient algorithms for computing the DFT have been developed[4][11]. These algorithms are collectively known as Fast Fourier Transforms (FFT) with a common general approach: the decomposition of the DFT into a number of succesively shorter and simpler DFTs.

Because the decomposition characteristic, the FFT algorithms operates on an N-point signal with N an integer power of 2; so the computational time is expected to be approximately N/log_2N.

Furthermore, every operation in the calculus requires trigonometric evaluations (sine or cosine) for each floating-point product and in turns time-consuming. To avoid this, an effective solution is to use a table lookup which is calculated and stored just once at the start of the program.

Thus, one time-improved approach should consider the speed in computing the FT.

6 THE IMPROVED VERSION

The improve in the parallel version implies to replace all the FT's (direct or inverse, particularly three), for the FFT algorithm.

The implemented FFT algorithm is the *Radix-2, Decimation in Time, in Place with Input Data-shuffling and Natural Order Output*[11], where the shuffling of the input is made by means of the *bit-reversal technique* and the time consuming in the trigonometric evaluations avoids by using a *table lookup*. The same applies to the Inverse FT because the FFT algorithm can be used to compute this with only minor modifications.

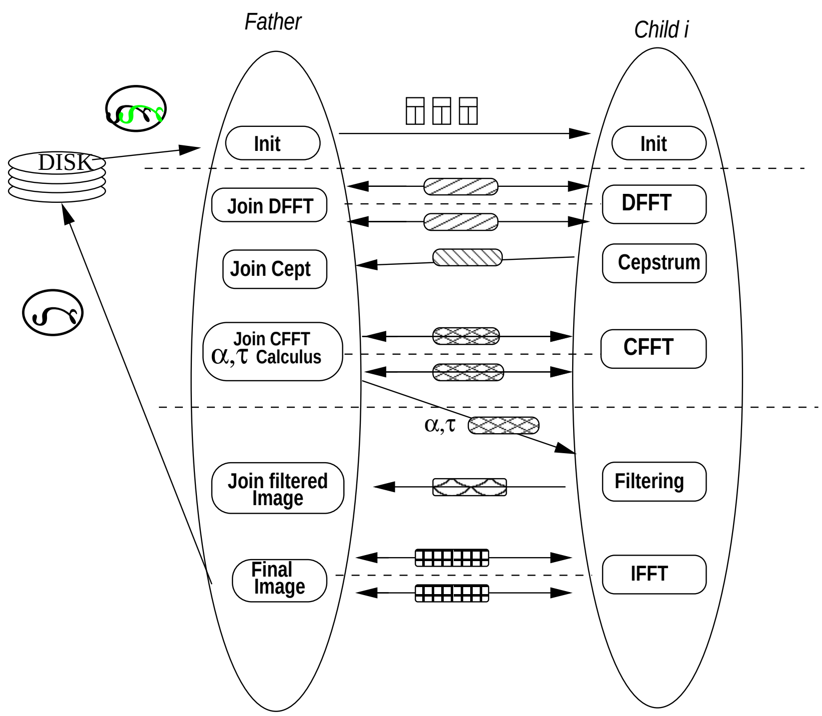

Figure 3: New Parallel version

At the start of the program the father generates the precalculated tables (bit-reversal, table lookup) and send them to the childs. Ver Fig. 3.

At difference with the first implementation, the childs never will need the whole original or intermediate images to make their work, so at every stage, they just keep waiting for the father send them the corresponding data blocks. Thus the resulting communication involved in one direction that the father made a personal broadcast and in the other the childs made a point to point communication.

The two-dimensional FFT treatment can be divided in two sequential steps, one operating on elements of the columns and one operating on (transformed) elements of the rows so only a one-dimensional FFT algorithm need be implemented. At first the father distribute the column blocks between the childs, collects the results and then distribute the row blocks. Every child process the data blocks in a FFT way using the same algorithm for the row and columns. A syncronization point is needed between these two steps because the childs must wait the father complete the pixmap before the row processing.

7 RESULTS

The experiments were based on square images of N-point with N an integer power of 2, more specifically images where its width and length were: 64, 128, 256, 512 and 1024. In these, just only the size of the images but not the echo existing inside them was considered, because the calculation time of those parameters (τ & α) is the same independently of their values.

The resulting processing times were obtained by running the echo elimination process in the sequential and the two parallel versions. For the parallel ones, different parallel virtual machines were built, involving a variable number of real processors.

The variable of interest was the total running time of the algorithm **EET** *(Echo Elimination Time)*, where:

$\mathbf{EET}_s$, $\mathbf{EET}_{op}$ and $\mathbf{EET}_{ip}$ are the sequential and parallel (original and improved version) times respectively.

From comparing the techniques a **Speed Up(Sp)** and **Parallel Profit(PP)** were get as representative measures.

$\mathbf{Sp} = \mathbf{EET}_s/\mathbf{EET}_{op}$ is the ratio between the sequential and original parallel processing times and

$\mathbf{PP} = \mathbf{EET}_{op} \ / \ \mathbf{EET}_{ip}$ is the ratio between the original and improved parallel processing times in the corresponding group of processes.

The figure 4 shows the mean values for the **Sp** with 2, 4 and 8 processes in each image size.

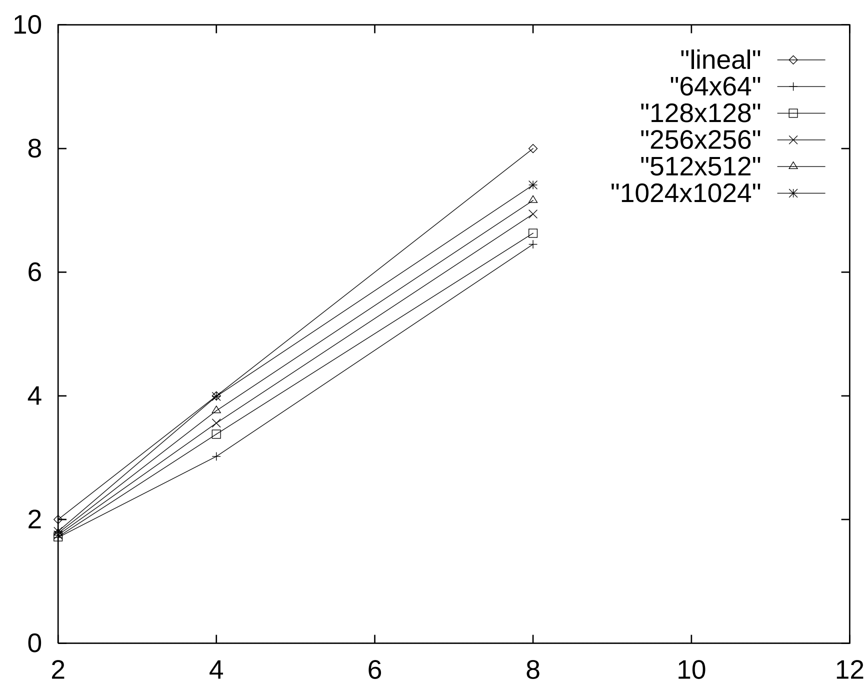

Figure 4: Speed Up values for each Experiments

As it can be deduced, the parallel processing times maintain an inverse relationship with the number of processes involved in the whole treatment. In general, the Speed Up increase as the size of image increases showing an effective improvement over the sequential processing time and getting a near-linear Speed Up.

The following figure shows the **PP** obtained with parallel processing of the different images considered.

Images	P = 2	P = 4	P = 8
64 x 64	221,90	140,61	63,91
128 x 128	972,66	535,81	280,10
256 x 256	2286,60	1184,85	624,47
512 x 512	9106,77	4576,19	2468,06
1024 x 1024	35447,13	17183,51	9563,53

Figure 5: Parallel Profit values for each Experiments

All cases show a good profit with the improved parallel system; in special when the size of the images increases. At difference, when the number of processes increase the profit decrease. This last is due to two reasons: the decomposition characteristic of the FFT algorithm since greater numbers of processes implies greater syncronization and communication between them and, the change of the communication pattern.

8 CONCLUSIONS

All image-adquisition processes are subject to noise of some type. Noise cannot be predicted accurately because of its random nature, and cannot even be measured accurately from a noisy image, since the contribution to the gray levels of the noise can not be distinguished from the pixel data. However, noise can sometimes be characterized by its effect on the image. This is the case of images with ghost where through an homomorphic system is possible to obtain an estimate of this characteristic [11].

A sequential implementation of this homomorphic system involves enormous processing time, so this has been a main issue of the present research.

Since this algoritm is fully determined and act identically on all input data (and the initial and final data structures are filled arrays) the algorithm is ideally suited to Single Process, Multiple Data Model. Data parallelism is achieved readily with near-linear increases in processing speed. The obtained gain through parallelism is pure, since by the characteristics of the problem there is no pay in time for the interprocess communication between the slaves (embarrassingly computation) into the parallel version. Only the master process increases the system processing time by the slave synchronization after every stage.

Although the advantage of using parallelism is clearly appreciated, the application remains computationally expensive and the system can be refined in some points by using different techniques. In particular, this parallel model is expensive in the use of resources for the computational time involved in the calculus of the FT, that can be avoided by using the FFT (Fast Fourier Transform) since the decomposition characteristic and its independency on the image dimension.

The improved parallel system provides a high performance for large image dimensions and this increases as image dimensions increases. Nevertheless, a loss of efficiency in the parallel computation is appreciable when a greater number of processes are involved in the processing. Specially, there is an overhead due to the granularity refining in the parallelism.

As the distributed approach to parallelise the echo elimination process showed its effectiveness, at the present time, experiment with larger number of processes and greater image dimensions are being performed.

In the future this model will be suited by attempting to minimize the overhead in communications without loss in parallel degree.

References

[1] Andrews, H. C. and Hunt B. R., *Digital Image Restoration*, (Prentice-Hall,1977).

[2] Benguelin, Dongarra, J., A. Geist, A. Jiang, W., Manchek, R. ,Sunderam, V. *PVM: Parallel Virtual Machine A Users' Guide and Tutorial for Networked Parallel Computing*, (The MIT Press).

[3] Bogert B., Healey M. & Tukey J., The Frequency Analysis of Time Series for Echoes: Cepstrum, Pseudo-Autocovariance, Cross-Cepstrum, and Saphe Cracking, *Proc. Symp. Time Series Analysis*, (M. Rosenblatt Ed., J. Wiley & Sons, 1963, pp. 209-243).

[4] Bracewell, R., *The Fourier Transform and Its Applications*,(Mc Graw-Hill,1965).

[5] Choudhary A. and Ranka S., Parallel Processing for Computer Vision and Image Understanding, *IEEE Computer*, (Vol 25. Nro. 2, 1992, pp. 7-9).

[6] Foster I., *Designing and Building Parallel Programs*, (Addison Wesley, 1995).

[7] Gonzalez R. and Woods R., *Digital Image Processing*, (Addisson Wesley, 1992).

[8] Guerrero R., Gonzalez A., Zavala E. & Colavita A., Aplicación de un sistema Homomórfico a la Detección de Ecos en Señales de Video B/N, *Proc. III CACIC*, (Argentina, 1997, pp 157-164)

[9] Guerrero R., Piccoli F., Printista M., Colavita A., A Parallel Approach of Image Processing System on PVM, *Proc. The IASTED International Conference on Computer Systems an Applications*, (Jordan, 1998, in press)

[10] Hsu H., *Fourier Analysis*, (Simon & Schusters, 1973).

[11] Lynn P. & Fuerst W., *Digital Signal Processing with Computer Applications*, (J. Wiley & Sons, 1994).

[12] Oppenheim A. & Schafer R., *Digital Signal Processing*, (Prentice Hall, 1975).

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

[14] Smith J., *The Design and Analysis of Parallel Algorithms*, (Oxford University Press, 1993).

[15] Webb J., Steps Toward Architecture-Independent Image Processing, *IEEE Computer*, (Vol. 25, Nro. 2, 1992, pp. 21-31).

[16] Wilkinson B. & Allen M., *Parallel programming: Techniques and Application using Networked Workstations*, (Prentice-Hall, 1997).