

EXHIBIT 1020

A PARALLEL IMPLEMENTATION OF A NOVEL CUBIC INTERPOLATION TECHNIQUE FOR VISUAL ANALYSIS

A. Bouridane, D. Demir, M. Erat, M. Sezgin, M. Bardakci, and S. Birecik

Department of Electronics,
Marmara Research Center,
P.O. Box 21, Gebze-Kocaeli,
41470 TURKEY.

Abstract- This paper describes the development of a transputer based system for interpolating 2-D images using a modified cubic interpolation technique. The technique devised decomposes a standard cubic interpolation into a set of two more cheap bilinear interpolations using the two windows surrounding any pixel in the interpolated image. The two bilinear interpolation results are then used to compute a local contrast by measuring the decrease or increase of the luminance in a given neighbourhood. A local contrast enhancement is then used to compute and enhance the value of the interpolated pixel. Implementation on a network of transputers using "image parallelism" has been carried out and a significant speedup factor obtained.

I. INTRODUCTION

Techniques for image enhancement are playing an increasingly important role in the area image analysis, visual inspection and related applications. In many cases, images to be analysed are of a low spatial resolution nature due to reasons of transmission or storage specifications. This is particularly the case for radar and medical images where very often low resolution images are used. Therefore, resampling the original images is very much desired in order to improve the resolution for further analysis such as inspection and/or recognition. Typically, there are two different processes of converting one sampling rate to another depending on whether the sampling rate conversion obtained results in a higher sampling rate than the original sampling rate, in which case the process is generally called *interpolation*, or lower than the original sampling rate and referred to as *decimation*. These techniques are becoming common place tools for many applications ranging across many diverse areas such as communication, speech processing, medical imaging, image processing, sonar, radar and antenna applications and many others. As the tasks encountered are so diverse, then many different techniques and algorithms have been developed over recent years. These have considered the data in the most appropriate form for the required

processing, such as spatial or spectral and many algorithms developed [1-3].

In the field of digital image analysis and inspection, many situations require the magnification of a particular section of the image data for further analysis and/or inspection/classification. Thus, the use of a convolution interpolation algorithm, which can be seen as the process of estimating the intermediate values of a continuous event from discrete samples, is essential. The majority of the techniques developed either do not provide an acceptable accuracy or are computationally expensive, as in the case of bilinear interpolation, cubic convolution interpolation [4] or cubic spline approximation [5]. However, none of the techniques include any enhancement capability to account for the inherent distortions of the images and to retain the integrity of the geometry of the images especially at the edges of the images which are seen to convey very important information. This problem is more pronounced in the case of low resolution images such as radar and medical images.

At the current stage of high resolution and high speed image recognition/inspection systems, new techniques with reduced cost are required. At Marmara Research Centre we have been interested in the use of interpolation techniques with application to inspection and classification of radar and medical images with emphasis on speed of operation and resolution. The cubic interpolation technique presents attractive results in terms of accuracy. However, on its own, it is deemed unsuitable as it suffers from the heavy computational effort incurred which severely hampers the crucial throughput rate. Another drawback of using this technique on its own is associated with the inherent distortions taking place at the resampled image edges as the technique does not consider any edge information. We have developed a novel technique for decomposing the standard convolution interpolation algorithm into a set of the more computationally cheap bilinear interpolation

processes using two smaller neighbouring subwindows around any given pixel position in the resampled image. This has the effect of considering separately the vertical and horizontal pixel information, thus taking into account the edge information each pixel conveys. The technique also uses a dedicated contrast enhancement operation which, unlike most contrast enhancement techniques, enhances a given pixel in the resampled image by measuring the relative decrease or increase of the luminance associated to any pixel in a given surrounding neighbourhood using the computed values of the resulting subpixels corresponding to the two neighbouring subwindows.

II. INTERPOLATION TECHNIQUES: A REVIEW

The process of resampling an image to a new set of coordinates very often results in a noticeable loss in image quality. To preserve to some extent image quality, the use of interpolation techniques is often desired. Currently, several interpolation functions are in use for image resampling. These range from simple functions which consider one or two interpixel distances to the more complex functions which involve more than three interpixel distances. The simplest of these is the nearest neighbour function, where the value of the new point in the interpolated image is taken as the value of the old coordinate which is located the nearest to the new point. Another function frequently used is bilinear interpolation, where the value of the new point is computed using a linear interpolation of the four pixels surrounding the new point. More elaborate functions which consider a bigger number of pixels in the interpolation process are also in use. The cubic interpolation and B-spline functions [4,5] are representatives of the second category and generally use sixteen pixels in the interpolation process. As such these functions are computationally expensive though they provide better approximation results.

The choice of an interpolating function for use in resampling an image depends on the task at hand. In practice, however, factors such as accuracy and speed very often govern the choice of a particular algorithm. If speed is the most crucial factor for the task at hand, then functions of the first category provide the best choice whereas functions of the second category are best suited if accuracy is the most critical factor. Therefore, functions providing a trade-off between speed and accuracy are very much desired. This paper describes the development of such an interpolation technique and its implementation onto a

network of transputers.

III. THE TECHNIQUE IN DETAIL

The essence of the present technique is the computation of a cubic interpolation algorithm using the more computationally cheap bilinear interpolation functions [6].

The technique devised first considers separately the two windows surrounding the new point in the interpolated image as shown in Fig.1. This is achieved by transforming every row and column in both the subwindows into a new pixel whose grey level value is the average value of the pixel values and centred at the middle of the corresponding row or column (see Fig.1). This has the effect of considering in both window the edge information conveyed by each pixels in the horizontal and vertical directions. A conventional bilinear interpolation process [1] is then applied separately on the pixels of both windows. This is followed by a dedicated contrast enhancement operation which, unlike most contrast enhancement techniques, enhances a given pixel in the resampled image by measuring the relative decrease or increase of the luminance associated with the computed bilinear interpolation as follows: Let p and a be the interpolated results obtained for the outer and inner windows, respectively. Then a local contrast for this neighbourhood at position (x,y) is defined as follows:

$$C_{xy} = \frac{|p - a|}{|p + a|} \quad (1)$$

This contrast is then enhanced using a dedicated enhancement function F satisfying the following conditions:

- (a) $C_{xy} \in [0,1]$
- (b) $C'_{xy} \geq C_{xy}$
- (c) $C'_{xy} \in [0,1]$ where $C'_{xy} = F(C_{xy})$

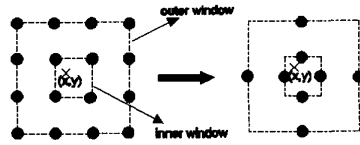
The following functions: $\sqrt{C_{xy}}$, $1 - \exp(-kC_{xy})$, $\ln(1 + kC_{xy})$, $\tanh(kC_{xy})$ satisfy the above conditions and have all been tested. The value of k was selected to give a high slope in the intermediate input range (0.1-0.6) and slope of about 40-50 degree for low input ranges (0.0-0.1) so that it does not enhance the noise due to digitisation. After enhancing the contrast C_{xy} the value of the resampled pixel g is computed as follows:

$$g = \frac{a(1 + C'_{xy})}{(1 - C'_{xy})} \quad \text{if } a \geq p \quad (2)$$

or

$$g = \frac{a(1 - C'_{xy})}{(1 + C'_{xy})} \quad \text{if } a \leq p \quad (3)$$

This has the effect of enhancing the contrast of the different objects while preserving the geometry of the image at the edges.



Window transformation used

IV. TRANSPUTER IMPLEMENTATION

The advent of transputer technology has heralded an unprecedented growth in the realisation of more powerful digital image processing systems. Many of these systems have found widespread use in vision applications and related topics. The use of transputers [7,8], in conjunction with the purposely designed Occam programming language, can provide a significant saving in computing power and offer a great deal of flexibility. Thus such processors are ideally suited to being used in image processing systems where the transfer of a large amount of data between processors is required, however, the inter-processor communication overhead can be a major constraint. Hence it is essential that these problems must be dealt with carefully in order to use efficiently the parallelism offered by the transputers. The analysis comprises a suite of operations ranging from accessing the data, filtering (averaging), to contrast enhancement. The nature of these images makes the analysis suitable for both "image parallelism" and "task parallelism" methods. Preliminary speed analysis was carried out to determine which of the parallelism techniques was best suited to the current application. It was found that "image parallelism" was superior. This is due to the fact that the "image parallelism" method involves a relatively small amount of inter-transputer communication which is inversely proportional to the number of transputers used in the network [9].

Hardware system

The hardware system comprises a scanner and a TMB16 transputer development board situated within

an IBM PC compatible. The IBM PC clone provides the interfacing to the scanner and the TMB16 board. A number of transputer modules reside on the TMB16 where one of them (host) performs the interface to the PC bus. The remaining transputers can be wired up together to make up the specific structure which maps the algorithm.

Software system

Due to the unavailability of a transputer based scanner board a set of modules were required to interface the TMB16 board to scanner via the PC bus for the reading and writing of images. This consisted of a set of C++ procedures for enabling the C012 link of the TMB16 transputer board and for setting up the correct configuration required for the accessing of data. In the case of using a multi-transputer network for the analysis, then many factors have to be taken into account when deciding upon the system configuration structure. These range from the accessing of the image down to the amount of parallelism introduced into the processing on a pixel basis. Another factor which greatly influences the performance of a multi-transputer network is the inherent inter-transputer communication overhead. Optimisation of this is achieved by maintaining a balance of the load allocated to each transputer module within the multi-transputer system. With all these requirements in mind, a tree-structure comprising seven transputers was used as shown in Fig.2. This structure is regular and has a very small communication overhead. This regularity makes it extendable with a minimum alteration of the Occam program code. One of the transputers acted as a host to the TDS system while the remaining four acted as the workers. The host transputer is a T800 with 4 MBytes of RAM so as to be able to run the D700D software whilst the other four transputers each have 1 MByte of RAM. The host transputer is responsible for the distribution of tasks to the workers and for receiving and dealing with the results as is appropriate. Each one of the four worker transputers acted on 1/4th of the image in parallel thus carrying out the "image parallelism" strategy.

The image under test is split into halves in the host transputer where one half is transferred to T_0 and the other half to T_1 . These then split the two halves into fourths and pass a fourth to T_2 , another to T_3 whilst retaining a fourth each. The above algorithm is then performed in parallel on the four worker transputers where each one uses its respective image data. This requires the merging of the results obtained for each part of the image data segment to account for

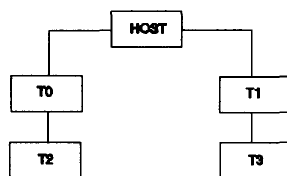


Fig.2. Multi-transputer structure used

the inherent image "slicing". Fig.3 shows the flow diagram used in the analysis.

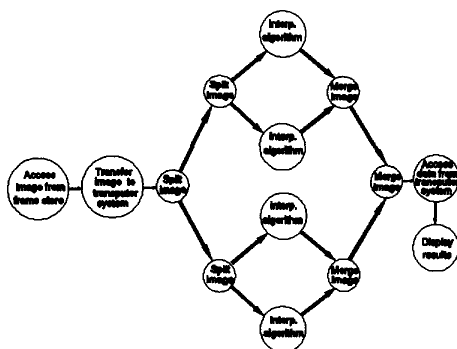


Fig.3. Flow diagram used in the analysis

The implementation issues for both intra and inter-worker communication are of major importance in the design stage.

The primary requirement of a worker processor is to perform the interpolation computations. In addition, due to the topology of the processor network, each node receives messages from its parent, and in the case of a non-leaf node, from its child/children and processes the results according to the required destination and intention. This is achieved using three communicating subprocesses operating concurrently within a worker. A process model representing this configuration is illustrated within the context of the tree topology in Fig.4 where the processes are named *GetData*, *Compute* and *Send*.

The *Compute* process performs the modified cubic interpolation algorithm for its allocated data segment. It also merges the resulting flaws with those received from its child/children. Using the *Send* process, results are communicated to its parent until an appropriate message (a byte of value 0) is outputted.

Thus informing the parent of the termination of the process. The process of getting image data from the parent processor and passing the appropriate segment to the child/children processors is accomplished using the *Getdata* subprocess.

In the case of the host process, a similar methodology was adopted. The process comprises two subprocesses namely *GetSend* and *Send*. While the *GetSend* process allows for the acquisition of the image data its distribution to the relevant children processors, the process *Send* performs the operation of the final merging of the appropriate flaws and the transfer of the final result to the PC display. This is terminated by the sending of the value 0 as above.

These simple mechanisms for the task and data distribution on the processor modules permit a very simple implementation with only a small communication overhead which is a crucial factor in the design implementation.

An evaluation of the system was conducted using different tree-like topology sizes of one, two and four worker transputers. The performance of each

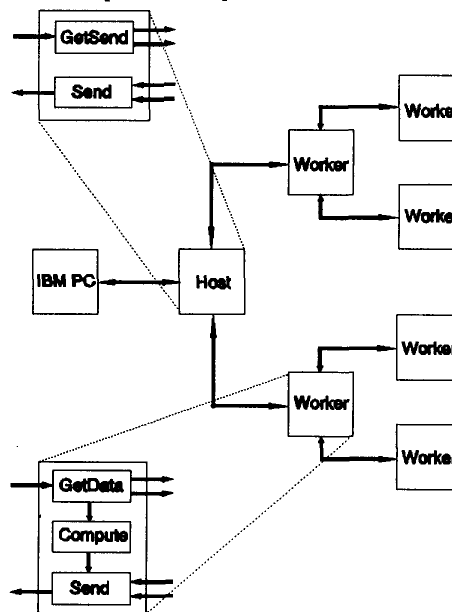


Fig.4. Process model used in the analysis

configuration is given as a measure of the corresponding efficiency and the speedup achieved over a single worker processor. The performance of each configuration is given in terms of the efficiency

algorithm to run on a single processor, and t_n the time for the concurrent version to run on n processors. Then the efficiency factor for a concurrent implementation is given by:

$$e = (t_1 / nt_n).100\% \quad (4)$$

Configuration	Performance	Speedup
1 worker		1.00
2 workers	92%	1.84
4 workers	90%	3.60

Table.1. Performance results for the system

Table.1 illustrates the performance obtained for a network of one, two and four transputers. Although, a near-optimum speed up was achieved, the results also show that they do not match the theoretical ones. This is mainly due to the fact that the whole image has to be sent to the host and parent processors for reconstruction purposes.

V. RESULTS

To gauge the effectiveness of the technique, extensive and subjective experiments have been carried out using both simulated images and real images taken directly by a scanner with a resolution of 402x435 8-bit pixels. These sample images consisted of some basic geometric shapes and hand written text images. These images were then resampled using 64x64 and 128x128 point image arrays by applying the conventional cubic interpolation and the new technique. Practical results have shown that the new decomposition technique not only leads to increased computational speed but to increased accuracy, as well. The comparison procedure was performed by subtracting the interpolated images from the original image and taking absolute values. These corresponding "error" images were then displayed and subsequently analysed using standard statistical methods for assessing the accuracy achieved. It has been found that the new technique provides a better approximation especially at the objects edges which are enhanced for a better human visibility. The application of the new technique on an IBM PC AT having a processor running at 25 Mhz took about 10 s and is about twice as fast as the standard cubic interpolation. For the network of Fig.2 comprising four transputer modules, then the analysis took about 1.5 s with an efficiency of 90%. As the speedup factor achieved for the transputer structure increases linearly

with an efficiency of 90%. As the speedup factor achieved for the transputer structure increases linearly with the number of transputer modules (see Table.1), then it is believed that by using a structure of a large number of transputer modules will provide real or near real time analysis.

VI. CONCLUSION

In this paper, a parallel implementation on a transputer network of a novel interpolation technique is described. The technique decomposes a cubic interpolation function into a set of the more computationally cheap bilinear interpolation functions. By utilising a dedicated enhancement procedure, pixels in the resampled images are approximated. Results have shown a large degree of success and the new technique not only provides a better accuracy but also it is computationally cheaper than the standard cubic interpolation technique. Although, the technique is of a sequential nature thus inherently slow, a transputer implementation using 'image parallelism' has helped improve its speed operation. The implementation utilises extensive use of parallelism in both the accessing and the processing of data within the transputer modules of the network.

BIBLIOGRAPHY

- [1]: D. A. LINDEN, D. A., "A Discussion of Sampling Theorems" Proc. IRE, Volume 47, July 1959, pp.1219-1226.
- [2]: R. E. CROCHIERE, and L. R. RABINER, "Interpolation and Decimation of Digital Signals - A Tutorial Review" Proc. IEEE, Vol. 69, No. 3, March 1981. pp. 300-331.
- [3]: K. NICKERSON, and S. HAYKIN, "Scan Conversion of Radar Images" IEEE Trans. Aeros. and Elect. Syst., Vol. AES-25, No. 2, March 1989, pp. 166-175.
- [4]: K. KEYS, "Cubic Convolution Interpolation for Digital Image Processing" IEEE Trans. Acoust. Speech and Sign. Proc., Vol. ASSP-29, No. 6, pp. 1153-1160.
- [5]: H. S. HOU, and H. C. ANDREWS, "Cubic Splines for Image Interpolation and Digital Filtering" IEEE

Trans. Acoust., Speech and Sign. Proc., Vol. ASSP-26, 1978.

- [6]: A. Bouridane, et al.: "A Modified Cubic Interpolation for Rapid Image Processing" to be presented at PCS-93, Lausanne, Switzerland, March 17-19 1993.
- [7]: Inmos Ltd, "Transputer Reference Manual" Prentice-Hall, 1985.
- [8]: L. F. JARDINE, and I. WHITWORTH, "Transputer Analysis of Visual Texture" IEE Computing and Control Eng. Jour. 2(2), 1991, pp.69-73.
- [9]: A BOURIDANE, and K. M. CURTIS, "A Parallel Processing Engine for Multi Level Flaw Detection" Parallel Computing 91 Conference, North Holland Publisher, 1992, pp.577-584.