Evaluating the DIPORSI Framework: DIstributed Processing Of Remotely Sensed Imagery*

J. A. Gallud[1], J. García-Consuegra[1], J. M. García[2], and L. Orozco[3]

[1] Departamento de Informática, Universidad de Castilla-La Mancha, Campus Universitario, 02071 Albacete, Spain jgallud@info-ab.uclm.es

[2] Departamento de Ingeniería y Tecnología de Computadores, Facultad de Informática, Universidad de Murcia, Campus de Espinardo, 30080 Murcia, Spain jmgarcia@ditec.um.es

[3] School of Information Technology and Engineering, University of Ottawa, 161 Louis Pasteur St, Ottawa, ON, K1N 6N5, CANADA lbarbosa@uottawa.ca

Abstract. The recent advances in remote sensing are contributing to improve emerging applications like geographical information systems. One of the constraint in remotely-sensed images processing is the computational resources required to obtain an efficient and precise service. Parallel processing is applied in remote sensing in order to reduce spatial or temporal cost using the message passing paradigm. In this paper, we present our experiences in the design of a workbench, called DIPORSI, developed to provide a framework to perform the distributed processing of Landsat images using a cluster of workstations. We have focused in describing the results obtained in the implementation of the georeferring function on a biprocessor cluster. In addition, this work offers a set of reflections about the design of complex distributed software systems like the distributed geographic information systems are.

1 Introduction

Recent advances in remote sensing are contributing to improve the application domain of geographical information systems. Remote sensing involves the manipulation and interpretation of digital images which have been captured from remote sensors on board of satellite or aircraft systems. Such images collect information about the Earth's surface, which allow scientists to perform many environmental studies.

Remotely sensed image processing is an interesting application area for distributed computing techniques, which has become more and more attractive both for the spatial resolution increase of the new sensor generation, and for the emerging use of Geographic Information Systems (GIS) in environment studies [12]. The large data volumes involved and the consequent processing bottleneck

* This work has been partially supported by Spanish CICYT, under grant TIC 2000-1151-C07-03

may indeed reduce their effective use in many real situations, and hence the need for exploring the possibility of splitting both the data and processing over several storing and computing units [14, 1].

Remote sensed imagery processing involves several steps that begin when a image is acquired by the satellite, and continue with the image analysis and processing to obtain, finally, a variety of results. All the procedures involved in the image manipulation of such images may be categorized into one or more of the following four broad types of computer assisted operations: image rectification and restoration, image enhancement, image classification and data merging [11].

In this paper, we describe a distributed workbench designed to perform a considerable number of the former tasks by using a cluster of workstations composed by either Windows NT/Windows 2000 or Linux platforms which are connected by means of an ethernet network using the Message-Passing Interface standard (MPI). MPI provides an interface to design distributed applications that run on a parallel system [13].

DIPORSI is the name of our distributed workbench, and it is related with the distributed processing of remotely sensed imagery. DIPORSI has been designed to be one of the modules of a more ambitious software project which is a distributed geographic information system.

In addition, this paper shows how the long computation time required by remote sensing procedures can be reduced without using high-cost parallel machines. All the distributed applications are implemented by splitting the original images into several small ones, which are processed in each node parallelly.

These practical results have been obtained from the implementation of the Landsat imagery rectification operation on several platforms, as a computer intensive function useful to show the global behaviour of the system.

Another important consideration about the practical results is that they can help us to improve DIPORSI in order to implement real time GIS.

The following section explains the georeferring function. Section 3 outlines the related work. In section 4, we describe the structure of our distributed workbench. In section 5, the distributed version of the georeferring function is presented. Section 6 shows the results obtained. Finally, in section 7 the conclusions and future work are presented.

2 The Georeferring Process

In remotely sensed of images, a number of different errors appear distorting them, in such a way that they cannot be used as maps. In order to use a remotely sensed in a GIS, a rectify process must be carried out before images can be used in the GIS applications[11].

The process of georeferring a satellite image consists in the application of a set of mathematical operations on the original image to obtain a geometrically corrected image. The rectification function is the process carried out to correct both radiometric and geometric errors from a remotely sensed image due to different causes (the influence of the atmosphere, earth surface and so on). So, the purpose of geometric correction is to compensate for the distortions introduced by different factors (earth curvature, relief displacement, etc). The aim of this process is that the corrected image has the geometric integrity of a map [10].

This process is usually implemented using one of two possible procedures [8, 11]. In the first one, the procedure uses the orbital model reproducing the physical conditions of the image acquisition. The second procedure applies the traditional polynomial correction algorithm with or without using a digital model terrain.

The most frequent cases are based on the second step related to random distortions. This kind of distortion is corrected by analyzing well-distributed ground control points occurring in an image. The GCP (ground control points) are relevant features of some known ground locations that can be accurately located on the digital imagery. Then, these values are submitted to a least-squares regression analysis to determine coefficients for two coordinate transformation equations that can be used to relate the geometrically correct coordinates and the distorted image coordinates. Once the coefficients for these equations are determined, the distorted image coordinates for any map position can be precisely estimated by using this expression:

$$X = f1(x, y)$$
$$Y = f2(x, y)$$

Where (X,Y) are the distorted image coordinates, (x,y) are the correct coordinates (map) and $f1$ and $f2$ are the transformation functions. The process is actually made inversely. An undistorted output matrix of empty map cells is defined, and then each cell is filled in with the correct coordinates, from the value of the gray level of the corresponding pixel, or pixels (depending on the method employed) of the distorted image. In a nutshell, the process is performed using the following operations:

1. The coordinates of each element in the undistorted image are transformed to determine their corresponding location in the original distorted image.
2. The intensivity value or digital number (DN) of the undistorted image is determined by using one of these usual methods: the nearest neighbor, bilinear interpolation and cubic convolution.

Therefore, it can be used three methods to obtain the digital number of each pixel. The nearest neighbor needs one only read for pixel. The bilinear interpolation uses four read operations to compute the digital number of each pixel. Finally, the cubic convolution method needs sixteen read operations to compute the DN of each pixel in the target image.

3 Related Work

There are a number of sensor features that must be considered before proceeding to perform the correction. So, firstly related work using the Themathic Mapper sensor aboard the Landsat satellite is presented, because this is the sensor we are interested in.

There are a few research work regarding the remotely sensed image geometric and radiometric correction process. An interesting paper covering the atmospheric correction of Landsat imagery can be mentioned [2]. In [4], we can find the radiometric and geometric SAR image rectification.

Despite the fact image rectification has not been enough treated in previous work, there is a explicit reference in many papers focused in image classification [1, 14, 6].

In [7], we can find the design of the MMIPPS system, a software package for multitemporal and multispectral image processing on parallel systems. The work offers a parallel approach to the classification process.

The purpose of many others research groups is the design of sophisticated remote sensing systems. In [15], the application of parallel paradigm to ecosystems monitoring can be found. A more ambitious system is the one designed by the U.S. Geological Survey to cover a variety of applications [9].

4 The DIPORSI Workbench

A first approach to the DIPORSI system was introduced in the development of a previous work [3]. As it was mentioned there, DIPORSI was configured as an execution module inside an image processing system.

The basic idea behind remotely sensed image processing is simple. The digital image is fed into a computer one pixel at a time, usually together with its neighbors. The computer is programmed to insert these data into an equation, or series of equations, and then store the results of the computation for each pixel. These results make up a new digital image that may be displayed or recorded in a raster format or may itself be further manipulated by additional programs [8].

Many remote sensing algorithms work in a parametric way, that is, the application performance is related to the initial parameters. The final process usually employs classifier algorithms and their behaviour are governed by a set of initial parameters. To get a classified image, the algorithm starts its computation with such inital values as yield the resultant image. This resultant image depends on both the initial parameters as well as the original multidimensional image. In many cases, the process must be repeated with other parameter values because the results are unacceptable. Thus, it is easy to understand the need to reduce both spatial and temporal costs when a trial-and-error process is employed.

Our goal was to implement an environment to perform remotely sensed image processing distributely where both the process and data can be distributed. So, we designed DIPORSI workbench, which stands for *DIstributed Processing Of Remotely Sensed Imagery*.

The final objective of our research team is to design a distributed geographic information system (D-GIS). One of the functions offered is the image processing module. In this module, a user can enter a set of complex image operations by means a script language.

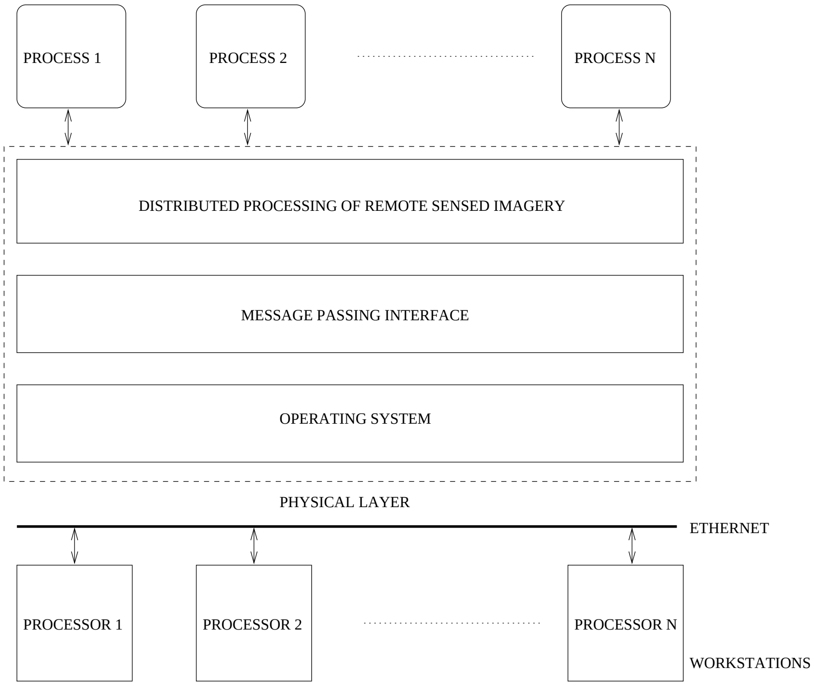

Fig. 1. DIPORSI functional diagram

In figure 1, the functional diagram of DIPORSI is exposed. DIPORSI appears as a layer between MPI functions and the code of each process. DIPORSI offers a set of functions and a message structure to the user for performing easily and distributively whatever remote sensing algorithm it chooses. DIPORSI runs in a batch way, and it needs a parametric file with the following information generates by the user:

- The algorithm, or the sequence of algorithms, to compute: i.e. the georeferring algorithm using the bilinear interpolation method.
- The workload allocation: how many nodes and how much information in each node must be defined.
- The number and the location of the data images.

DIPORSI offers a set of defined functions to generate special messages to interpretate them, a number of functions for managing files of different kinds such as text files, data files, raster images, etc.

Our workbench allows the user to work by distributing both spatially and temporally either of the remote sensing algorithms. That is, either user can make all the nodes perform the same computations on different data or the user can make each node runs different computation on the same image.

When a distributed approach of a particular algorithm is defined, its special features must be studied in order to improve the efficiency. DIPORSI follows a general scheme which is applied to all remote sensing algorithms. Such a scheme is structured into three steps:

1. The images (in our example, the distorted images) are broadcasted into the nodes.
2. The computations are performed following the particular algorithm in each node.
3. The resultant image is restored by the nodes usually on the master computer.

In order to confer a greater generality to our development, and with a view to further applications, two types of messages have been defined: control and data. The first ones are used to specify the activity to be carried out and the parameters necessary for the algorithm, as well as for the control of the MPI implementation. The second type of messages are used for the passing of data or image files, as well as pixel streams.

5 The Distributed Georeferring Process

In this section, the distributed algorithm is described and the platforms used to implement it are presented. So, we are going to show the scheme of the distributed algorithm.

```
compute_target_dimensions(max_rows,max_columns)
compute_distribution(nrows) distribute_control_data
[distribute_data]
 for b=1 to max_bands do
 for i=firstrow to nrows do
  for j=1 to max_columns do
    get_coordinates(i,j,x,y)
    nd=read_digital_number(b,x,y,method)
    write_pixel(b,i,j,nd)
  next j
 next i
next b
```

The *compute_distribution* function gets the number of rows each node has to compute. The *distribute_control_data* function is the method used by the master process to communicate each node the task to perform and the data to process. The *distribute_data* function appears between brackets because it is an optional function depending the data location (local or remote).

The first step is made by the master process, which opens a file that contains the information of the task to be solved. The root process is initiated with a parametric file, in which all the activities to be performed and their parameters are specified in a ordered way. The master process sends such information to all the nodes involved. Thus, each node knows what it must do (correction method) and how much information it must receive (number of bands -the resultant image to a given frequency-, resolution, the transformation functions, etc).

At this point, the master process sends the Landsat image to all the nodes. The special features of the georeference algorithm force us to send all the data to all the nodes. As explained above, the resultant image is a combination of rotations and translations of the original distorted image.

On the other hand, the MPI broadcast function allows us to distribute the image easily, but unfortunately at a high cost.

The second step is executed in each node, which acts on the region of the distorted image where the computations must perform the corrections. As soon as a row of the resultant image is computed, it is sent to the master process, so computation and communication are overlapped. In addition, when a node is involved in a communication, the others can carry out correction computations.

The second and third step run at the same time. The master process receives partial data from nodes, together with the information to restore the resultant image. This is made by using a special message with information about the location (x and y coordinates and the number of the band) of the received data.

6 Evaluation Environment and Results

A Landsat scene has 7 bands, 6 with 30x30mts and 1 with 120x120 mts of resolution respectively, with approximately 40MB each band, which explains the high value of the response time when geometric correction is computed in a single machine.

DIPORSI works by splitting the original distorted image into number-of-rows/number-of-nodes blocks in accordance with an uniform workload allocation, then each node computes a small submatrix and the computation time can be reduced and the overall performance of the algorithm can be improved.

There are different MPI implementations, most of them designed to run over Windows 2000, Unix or Linux workstations. We have used a MPI implementation on Windows NT named WinMPICH 0.9b version [5] and two known MPI implementations for Linux called LAM and MPICH.

We have tested different georeferring implementations of DIPORSI over different hardware and software platforms. In a previous work [3], we describe a detail of the results obtained from different platforms and different georeferring methods.

In this work we show the evaluation of DIPORSI in the next hardware platform: Pentium II 300MHz dual, 64MB and Linux RedHat and NFS over fast ethernet.

To the interpretation of the results, we must take into account the following considerations:

1. All data times show service or response time.

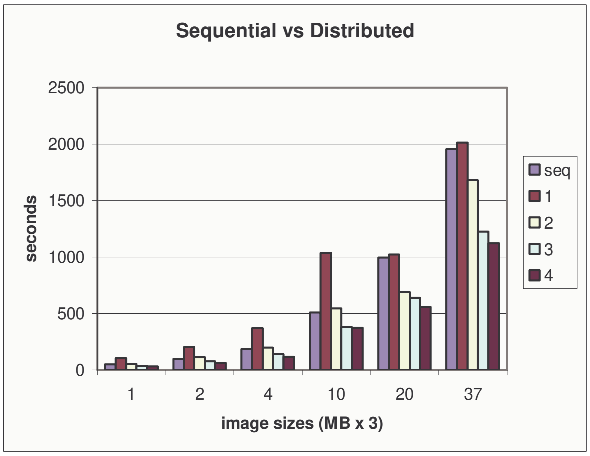

Fig. 2. Sequential vs distributed georeferring

2. In order to measure the service time we have used the times function.
3. The geometric rectification of a three band scene means that we have to open three original images and three target ones. In the worst case, we have to open three 37MB original images and three over 50MB target images simultaneously.
4. We consider only the cubic convolution interpolation method.

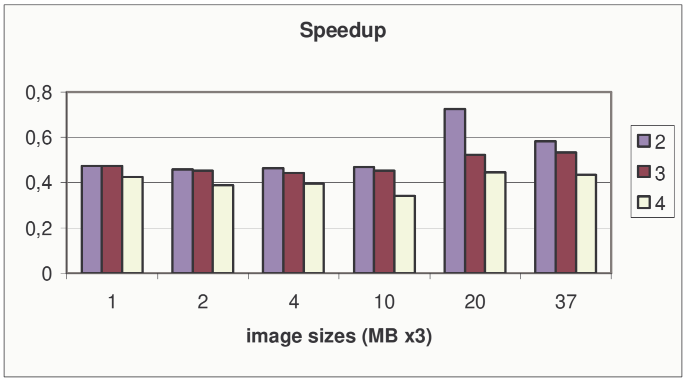

Fig. 3. Speedup of the data showed in the figure 2

Figure 2 shows a real view of the performance obtained when we compare the sequential version and the distributed ones. Figure 3 shows the speedup corresponding to the figure 2 execution. The benefits of using a cluster appears when large image sizes are used together with a intensive algorithm, as cubic convolution, is needed. We have tested that there is a lower advantage when nearest neighbor is selected.

7 Conclusions and Future Work

This paper describes a distributed workbench to implement the main remote sensing algorithms, using a cluster of workstations, called DIPORSI. The implementation shows the timing results when different image sizes are used, as well as the time gained with the parallel algorithm for georeferring. MPI proved useful for implementing distributed applications on low-cost platforms, which can contribute to designing efficient solutions in remote sensing.

Future work admits of several possibilities: adding dynamic workload allocation, the implementation of other algorithms to solve geometric correction, the implementation of radiometric correction, automatic detection of the ground control points, and other tasks intended to improve the communications between nodes.

References

1. D. A. Bader, *Parallel Algorithms for Image Enhancement and Segmentation by Region Growing with an Experimental Study* (In The Journal of Supercomputing. Vol 10, No 2, 1996) 141-168.
2. H. Fallah-Adl et al, *Fast Algorithms for Removing Atmospheric Effects from Sattellite Images* (In IEEE Computational Science and Engineering, Vol. 3, No. 2, Summer 1996).
3. J. A. Gallud, J. M. García y J. García-Consuegra, *Cluster Computing Using MPI and Windows NT to Solve the Processing of Remotely Sensed Imagery* (Recent Advances in Parallel Virtual Machine and Message Passing Interface. Lecture Notes in Computer Science 1697, Springer-Verlag, 1999) 442-449.
4. I. Glendinning, *Parallelisation of a Satellite Signal Processing Code - Strategies and Tools* (In ACPC '99. LNCS 1557, 1999) 388-397.
5. W. Gropp, E. Lusk, N. Doss, A. Skjellum, *A High-Performance, Portable Implementation of the MPI Message Passing Interface Standard* (Parallel Computing, Vol. 22, No. 6, 1996) 789-828.
6. F. M. Hoffman, W. W. Hargrove, *Multivariate Geographic Clustering Using a Beowulf-style Parallel Computer* (In Proceedings of the International Conference on Parallel and Distributed Processsing Techniques and Applications, PDPTA '99, Volume III, H. R. Arabnia, Ed. ISBN 1-892512-11-4, CSREA Press) 1292-1298.
7. J. Janoth et al, *MMIPPS - A Software Package for Multitemporal and Multispectral Image Processing on Parallel Systems* (In ACPC '99. LNCS 1557, 1999) 398-407.
8. T.M. Lillesand and R.W. Kiefer, *Remote Sensing and Image Interpretation 3rd Edition* (J.Wiley & Sons), 1994.
9. U.S. Geological Survey, *Earth Resources Observations Systems*, (MAGIC-II Project, 1997, http://edciss1.cr.usgs.gov).
10. B.L. Markham, *The Landsat Sensors' Spatial Responses* (IEEE Transactions on Geoscience and Remote Sensing, Vol. GE-23, No. 6, 1986) 864-875.

11. P.M. Mather, *Computer Processing of Remotely-Sensed Images* (John Wiley & Sons), 1987.
12. J.A. McCormick and R. Alter-Gartenberg and F.O. Huck, *Image Gathering and restoration: information and visual quality* (Journal of Optical Society of America, Vol 6, No. 7, 1989) 987-1005.
13. MPI Forum, *MPI:A message-passing interface standard* (International Journal of Supercomputer Applications 8(3/4), 1994).
14. J.C. Tilton, *A Recursive PVM Implementation of an Image Segmentation Algorithm with Performance Results Comparing the HIVE and the Cray T3E* (In Proceedings of the IEEE Symposium of Massively Parallel Processing, 1998).
15. C. J. Turner and G. J. Turner, *Adaptative Data Parallel Methods for Ecosystem Monitoring* (In Proceedings of the ACM Conference on Supercomputing '94, 1994) 281-291.