

Solar Images Processing in Parallel Environment

Nelson D. A. Mascarenhas¹, Hanumant S. Sawant², Célio E. Morón¹, Reinaldo R. Rosa³,
José H. Saito¹, Hélio C. Guardia¹, Lilian N. Faria¹

¹ Department of Computer Science, Federal University of São Carlos - UFSCar
C.P. 676 - CEP 13565-905 São Carlos, SP, Brazil
{nelson, celio, saito, helio, lilian}@dc.ufscar.br

² Astrophysics Division (DAS) and
³ Associated Laboratory for Computing and Applied Mathematics (LAC),
National Institute for Space Research - INPE
C.P. 515 - CEP 12201-970 São José dos Campos, SP, Brazil
{sawant@das.inpe.br, reinaldo@lac.inpe.br}

Abstract - This paper presents the PVA-BDA project (Processing, Visualization and Analysis in Parallel Environment of the BDA Data) that has been developed for processing of solar images that will be captured by the BDA (Brazilian Decimetric Array), a radio telescope under development at the National Institute for Space Research (INPE). In a joint effort between the Department of Computer Science at Federal University of São Carlos (DC/UFSCar), the Astrophysics Division (DAS) and Associated Laboratory for Computing and Applied Mathematics (LAC) at INPE, a high performance parallel system is being developed with capacity to support realistic applications, involving a reasonable amount of parallel processing, in order to carry out the processing, visualization and analysis of solar images captured by BDA, in real time. The aim is to create the conditions for starting a study of the solar weather forecast. The forecast of solar explosions are important as they may cause serious perturbations in terrestrial communication systems. An application for 3D reconstruction of X-ray tomographic images of the solar atmosphere was developed at DC/UFSCar. Due to the need for the 3D reconstruction of solar magnetic structures, in real-time, this application was implemented to execute in a parallel machine using DSPs.

Key-words: High Performance Systems, Tomographic Reconstruction, Solar Explosions.

I. Introduction

With the creation of the International Program of Space Weather Forecast [3], started in 1989, in Japan, several laboratories of space weather forecast had been established in institutes and universities around the world, in order to investigate the solar activity. The main objective is to forecast the solar explosions that may cause serious perturbations in terrestrial communication systems, damage satellites and energy transportation systems, as well as the emission of extremely dangerous radiation levels for astronauts in space missions.

In Brazil, scientists at the Astrophysics Division (DAS) and Associated Laboratory for Computing and Applied Mathematics (LAC) at National Institute for Space Research (INPE) are participating in the Latin-American Space Weather Forecast Program.

There are suggestions that energy is liberated to solar explosions, known as solar flares, in the region where decimetric emission is generated. Therefore, Brazilian Decimetric Array (BDA) is being developed by scientists of DAS/INPE to obtain high spatial resolution solar images for better understanding of the fundamental problems of solar physics, consequently space weather prediction [20].

The angular resolution of a single (parabolic) radio telescope is proportional to the diameter of the dish. In order to obtain telescopes with high resolution, it would be necessary to build bigger diameter antennas. Alternatively, the technique of radio interferometry allows building radio telescopes with high resolution, by adding the signals from a number of small antennas in phase and amplitude to simulate large telescopes. The resolution of an interferometer is proportional to the length of

baseline, a distance between extremities of the antennas.

The BDA project, financed by FAPESP (Proc. 01/00056-9), includes the construction of 38 parabolic antennas of 5 meters diameter, which will be distributed in a "T" shaped array with base lines of 2.5 Km in the east-west direction and 1.25 Km in the south direction. This instrument will be able to capture up to 10 solar images per second, in the frequency range of 1.2 to 5.0 GHz.

A Prototype of the Brazilian Decimetric Array (PBDA) composed by 5 antennas has been built and tested at INPE in São José dos Campos at the beginning of 2003. Currently this prototype is being moved to its definite location at INPE in Cachoeira Paulista/SP.

Due to the high computational costs involved in the processing, visualization and analysis of a great amount of solar images captured by BDA, a high performance computer system becomes necessary to carry out the forecast of the solar explosions. As a joint effort between the Department of Computer Science at Federal University of São Carlos (UFSCar), and the DAS/LAC at INPE, a high performance parallel system is being developed with capacity to support realistic applications, involving a reasonable amount of parallel processing. This project is being financed by FINEP (Studies and Projects Funding Agency).

The DC/UFSCar acquired, in a previous project, a system composed by a cluster of processors, plus a parallel machine based on DSPs (Digital Signal Processors). The high performance cluster is composed by eight 1 GHz Pentium III processors to execute the application, an administrative station and an access station. The system based on DSPs (Atlas system), developed by Eonic Solutions GmbH, is composed by one host PC Pentium and four DSPs using the Virtuoso real-time operating system. During this project another cluster with 32 Xeon 2.4 GHz processors and an administrative station was incorporated to this architecture.

In order to offer support to the development of parallel programs in the Atlas system using the kernel Virtuoso, a *Visual Environment for the Development of Parallel Real-Time Programs* [4] was developed at the Department of Computer Science at UFSCar. Using this tool, an application for 3D reconstruction of X-ray tomographic images of the solar atmosphere was implemented at DC/UFSCar to execute in a parallel machine using DSPs.

This paper is organized as follows. In section 2, the BDA project is presented. Section 3 describes the parallel systems and the Visual Environment for the Development of Real-Time Parallel Programs

used for the development of the parallel application. Section 4 describes a parallel 3D reconstruction method of coronal loops. Section 5 presents the results. Finally, the Section 6 presents the conclusions.

II. Images Capture by the BDA

The BDA is a radio telescope that employs modern techniques in radio interferometry, being the first radio interferometer developed in Brazil, and the unique in the Latin America.

Some international Institutes are collaborating in the project, including the National Centre for Radio Astrophysics (NCRA-TIFR, Pune, India), the Indian Institute of Astrophysics (IIA, Bangalore, India), and the Department of Radio Astrophysics of the University of Berkeley (CA, USA). The BDA is being developed with resources from INPE, FAPESP, CNPq and FINEP.

The Prototype of Brazilian Decimetric Array (PBDA) consisting of 5 element altitude-azimuth mounted parabolic dishes has been successfully developed at INPE in São José dos Campos, and it is in operation since April 2003. This array operates in the frequency range of 1.2 -1.7 GHz [20].



Figure 1 – Prototype Brazilian Decimetric Array, INPE, São José dos Campos.

The final version of the BDA will be an interferometer array consisting of 38 parabolic antennas of 5 m diameter distributed in a "T" shaped array, located at Cachoeira Paulista-SP, Brazil.

The BDA will obtain solar images with high (~5 x 8 sec of arc at 5.0 GHz) space resolution, in the frequency range of 1.2 to 5.0 GHz, at the rate of 10 images per second.

Once in full operation, in 2006, the BDA will be able to obtain, in a regular basis, radio images of the solar disc and those of isolated active regions. The observation of solar phenomena will allow a better understanding about the mechanisms of the Coronal Mass Ejection (CME). In addition, the investigation of solar phenomena will contribute to improve the modeling of the space weather forecast, which is one of the aims of the project.

III. System for the Development of Programs for Parallel Machine

A parallel system [5] is being developed by the Department of Computer Science at UFSCar in cooperation with the Astrophysics Division at INPE, aimed at the processing, visualization and analysis, in real-time, of 3D tomographic images of the solar atmosphere.

A. Parallel system based on DSPs

The ATLAS™ system, from Eonic Solutions GmbH, includes hardware and software to implement and execute applications that need high performance and digital signal processing. This system is composed by one host PC Pentium with Windows NT, and four processors ADSP-21160 (Hammerhead SHARC™) from Analog Devices. These high performance signal processors are used for communications, graphics, and imaging applications, which combine floating-point operations with multi-processing support.

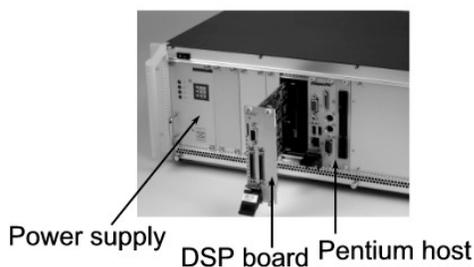


Figure 2 – Atlas parallel system.

ATLAS™ system is shipped together with the fully installed Virtuoso™ real-time operating system (RTOS) from Wind River Systems, Inc.

1. Virtuoso real-time operating system

Applications developed using Virtuoso kernel (Virtual Single Processor Programming System) [17] are divided into tasks; that is, independent program modules that can interact with other tasks through communication and synchronization services. In multi-processor systems, the tasks can be easily distributed among the different processors, until the real-time requirements are met.

The Virtuoso kernel is an operational system that concentrates only the objects and services necessary for the development of real-time applications in multi-processor systems. Each microkernel object – task, semaphore, resource, and so on – has specific attributes and supports a set of services.

The main microkernel objects are: tasks, semaphores, mailboxes, queues, channels, memory partitions, resources and timers.

Task - A task is a program module that exists to perform a defined function or a set of functions. A task is independent of other tasks but may establish relationships with them.

Semaphore - Semaphores are used to synchronize two tasks and/or events. A signaling task will signal a semaphore while there will be another task waiting on that semaphore to be signaled. One can wait on a semaphore with a time-out or return from the wait if no semaphore is signaled. This can be useful to make sure that the task does not get blocked.

Mailbox - Messages are used between a sender and a receiver task. This is done using a mailbox. The mailbox is used as an intermediate agent that accepts messages. Messages work with arbitrary sizes and allow a selective transport between sender and receiver.

Queue - Queues are also used to transfer data from a task to another one but here the data with fixed size is actually transferred in a buffered and time-ordered way. The advantage is that no further synchronization is required between the enqueueing and the dequeueing task, allowing the enqueueer to carry on.

Channel - A channel consists of queued writer(s) and reader(s) and an optional channel buffer. In the unbuffered case, data with arbitrary size will flow directly from writers to readers. When using the option of channel buffers, data will probably first be copied to the buffer before being finally transferred to the reader. Channels should be thought of as software 'pipes' that allow one task to put data in and another one to take it out. In addition, channels allow communication between a task and an external program.

Memory - In any system, memory is a resource for which tasks compete. Memory management is an area where various techniques can be applied. Many techniques are very fast and use elegant models for allocation and deallocation of memory.

Resource - The resource protection calls are needed to assure that the access to resources is done in an atomic way. Unless the processor can provide real physical protection, the locking and

unlocking of a resource is in fact a convention that all tasks using a resource must follow.

Timer - This class of calls allows an application task to use a timer as part of its function. From then on, the timer can be started to generate a timed event at a specified moment (one shot) or interval (cyclic). This event can then signal the associated semaphore. Timers are mainly used to regulate the execution of tasks in relation to a required timely behavior.

2. Visual Environment for the Development of Real-Time Parallel Programs

In order to offer support to Virtuoso, a set of tools is required to help in the development of parallel applications, debugging and analysis of real-time requirements.

To address these needs, a Visual Environment for the Development of Real-Time Parallel Programs [4, 6] was developed at the Department of Computer Science at UFSCar.

In the Visual Environment, applications are built through the construction of a graphical model. This model is represented by a graph, where nodes denote the data structures that compose a parallel program (tasks, signals, resources, mailboxes, etc.), and arrows denote the communication and synchronization operations between the structures. The information in the graphical model can be complemented with code written by the user. Based on this information, the source code of the application is automatically generated.

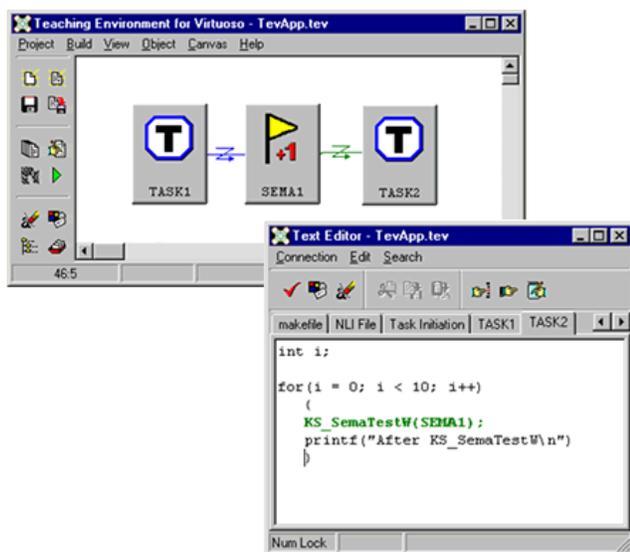


Figure 3 – Visual Environment for the Development of Real-Time Parallel Programs.

B. High Performance Clusters

The cluster of Pentium III processors is composed by four processing units (Dual Pentium III 1GHz), an administration server (Dual Pentium III 1GHz) and an access station (Pentium III 1GHz).

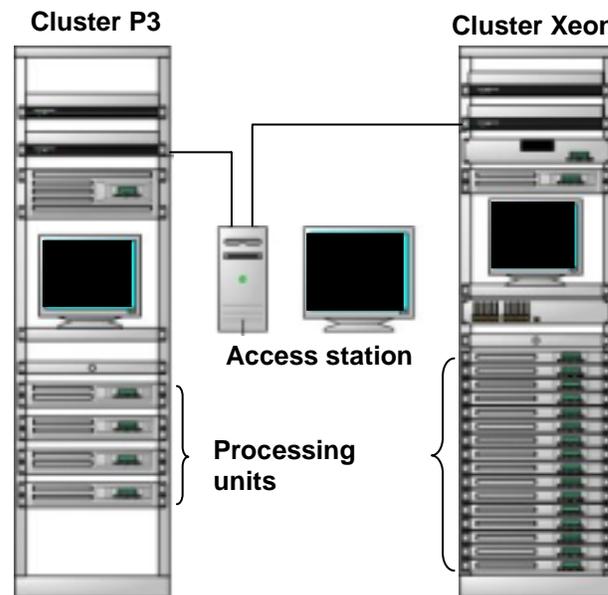


Figure 4 – Parallel system formed by clusters of processors.

A data network with Gigabit technology links the processing units used for ultra-rapid communication during the processing of the applications, and a service network using Fast-Ethernet technology links all the equipment for general communication.

The processing units are responsible for the execution of the applications.

The administration server is employed in administrative functions such as starting up and shutting down, configuration of the cluster, monitoring, backup, etc. The administration server also executes important services for the cluster such as clock synchronization, file server, DNS server, among others, as well as being the interface for the cluster with the corporate environment network (system gateway).

The access station offers to the users a local interface with the system. Using this station, users can submit jobs to the processing units, edit and compile programs, schedule their jobs, and execute backups and visualization of the results.

This system was expanded with the acquisition of a cluster of Xeon processors composed by 16 processing units (Dual Xeon 2.4 GHz) and an administration server (Dual Xeon 2.4 GHz).

The Visual Environment for the Development of Parallel Real-Time Programs [4] is being ported to Linux to aid in the development of parallel applications at the high performance cluster.

IV. Solar Images Processing

The Sun emits energy at different wavelength of the electromagnetic spectrum, from radio waves up to higher energy emissions such as X-rays and gamma rays. The radio waves can be detected by radio telescopes on the Earth, but X-rays and gamma radiations do not penetrate in terrestrial atmosphere and, therefore, only telescopes on board of satellites can detect them.

With the advance of space technology, solar images with high spatial and temporal resolution, captured by satellites equipped with X-ray telescopes, revealed dynamics of structures, known as *coronal loops* [15, 16], or the active regions in the solar corona. The coronal loops (Figure 5) consist of the magnetic field lines that connect magnetic regions on the solar surface, holding a hot moving plasma inside a tube having an arc shape.

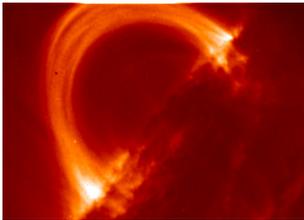


Figure 5 – Coronal loop.

Frequently, X-ray images of coronal loops show twisted structures with sigmoidal shape (S or inverted S shape), as shown at Figure 6. This occurs due to a continuous motion of fluid rotation at both ends of the loop that generates torsional waves that propagate along the tube. After several rotations, the coronal loop can become unstable, and suspect able to plasma instability giving rise to solar explosion/flare and or ejection of mass – CME (Figure 7).

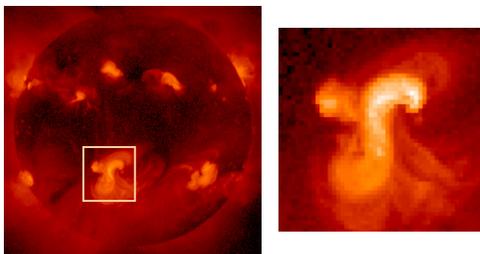


Figure 6 – Twisted coronal loop.

Thus solar explosion or flare is nothing but a sudden release of great amount of stored magnetic energy in the solar corona, which can be observed at various wavelengths/energy bands, including in X-ray and ultraviolet, as bright structures (Figure 7a). The emissions of energetic particles may cause serious perturbations in terrestrial communication systems, damage to satellites and energy transportation system. The intensities of these emissions are extremely dangerous for astronauts in space missions.

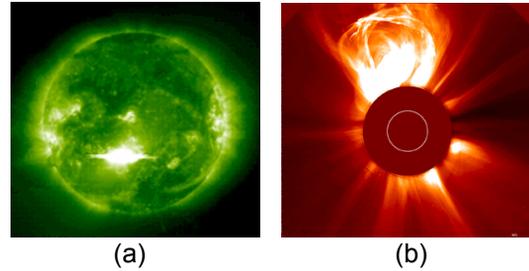


Figure 7 – Solar flare and CME.

Thus it is important to investigate dynamics of coronal loops for improving capability of predictions of solar explosions/activities.

At INPE, space scientists have been carrying out a detailed study of the spatio-temporal dynamics of the coronal loops [8]. However, these studies are carried out with bidimensional images without any information about the 3D structure of the loops. In this way, physical parameters of the magnetic loop (temperature, density and magnetic field) and geometric parameters (height, rotation and position) are subjected to errors due to the spatial limitation imposed by bidimensional images.

Motivated by this need, efforts have been made in order to allow the tridimensional reconstruction of the coronal loops from bidimensional images obtained through satellites. Rosa *et al.* presented methods for reconstruction of magnetic loops from the top and bottom images of the Yohkoh satellite [11, 12, 13].

The techniques developed for the processing of solar images obtained through satellite can be used in 2D images captured by the BDA after the conclusion of the project.

A. 3D Reconstruction of Coronal Loops

The X-ray tomographic images captured by telescopes on board of the Yohkoh satellite show the energy emission of a magnetic loop in two different depths of the solar atmosphere. The bottom of the loop emits hard X-ray (HXR), that is, X-ray of high energy, detected by the Hard X-Ray Telescope. The top of the loop emits soft X-ray

(SXR) of lower energy that is detected by the Soft X-Ray Telescope. Figure 8 shows the X-ray images with a resolution of 128x128 pixels, with information of the top and bottom of the loop, respectively.

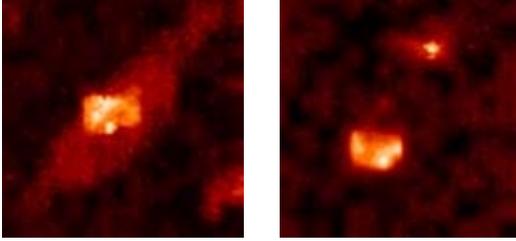


Figure 8 – Top (SXR) and bottom (HXR) images of the loop observed by the Japanese satellite Yohkoh.

In order to reconstruct the 3D structure of the coronal loop from these X-ray tomographic images, a reconstruction method is used to generating the intermediate images of the loop. However, as there is no information about the tridimensional structure of the loop between the top and bottom images, conventional methods of image interpolation do not reconstruct the magnetic loops, because the shape of the transversal sections between the top and bottom images must gradually be modified to generate the arc.

In order to solve this problem, a reconstruction method was developed to reconstruct the coronal loop using image metamorphosis [14] with a transition function controlled by a Bezier curve in arc shape to generate the intermediate images.

1. Image Morphing

Image metamorphosis (or image morphing) is the gradual transformation of shape and color of one image into another one by the generation of a sequence of intermediate images [19]. Pairs of corresponding points in the two images are specified to compute mapping functions that define the spatial correspondence between all pixels in both images.

The morphing process consists of image warping, so that the two images have the same shape, and color interpolation with cross-dissolve between the deformed images. Cross dissolving is the process of blending the colors of two images in order to obtain colors in an intermediate image.

Image warping maps each pixel of an image into a new position of the other image. In order to specify the deformation from the source image to the destination one, we need a set of pairs of corresponding points $\{(\mathbf{p}_i, \mathbf{q}_i) \mid \mathbf{p}_i, \mathbf{q}_i \in \mathbb{R}^2, i \in 1, \dots, n\}$, where \mathbf{p}_i specifies a location in the

source image that corresponds to a location \mathbf{q}_i in the destination image. These pairs of control points are used to move the pixels at corresponding points from the source position to the destination one and move all other pixels along in a consistent manner [18].

A mapping function for image warping using the inverse distance weighted interpolation method can be formulated as follows:

Given n pairs $(\mathbf{p}_i, \mathbf{q}_i)$ of control points, we need to find at least one continuous function $f: \mathbb{R}^2 \rightarrow \mathbb{R}^2$ with $f(\mathbf{p}_i) = \mathbf{q}_i, i = 1, \dots, n$. For each pair of corresponding points $(\mathbf{p}_i, \mathbf{q}_i)$, a local interpolator $f_i: \mathbb{R}^2 \rightarrow \mathbb{R}^2$ with $f_i(\mathbf{p}_i) = \mathbf{q}_i$ is determined. Linear local interpolators are normally used to move the control points by a linear transformation [14]. The movement of neighbour pixels is controlled by a mapping function that depends on the distance between the pixel to each of the control points of the image. Therefore, pixels more distant of a control point are not affected by the displacement of this point.

The mapping function $f: \mathbb{R}^2 \rightarrow \mathbb{R}^2$ is a weighted average of the local interpolator functions, with weights that depend on the distance from the pixel to its respective control points. The inverse distance weighted function is given by:

$$f(x, y) = \sum_{i=1}^n w_i(x, y) f_i(x, y). \quad (1)$$

The weight function $w_i: \mathbb{R}^2 \rightarrow \mathbb{R}$ depends on the inverse distance from a pixel $\mathbf{p}(x, y)$ to the control point $\mathbf{p}_i(x_i, y_i)$, normalized by the sum of the inverse distances to all control points:

$$w_i(x, y) = \frac{d_i(x, y)}{\sum_{j=1}^n d_j(x, y)}, \quad (2)$$

where $d_i(x, y)$ is the inverse Euclidean distance from a pixel \mathbf{p} to the control point \mathbf{p}_i .

The weight function w_i must satisfy the conditions:

- $w_i \geq 0$, and $\sum_{i=1}^n w_i(x, y) = 1$;
- $w_i(x_i, y_i) = 1, w_i(x_j, y_j) = 0$, with $i \neq j, i, j = 1, \dots, n$.

Figure 9 shows the mapping process of a pixel $\mathbf{p}(x, y)$ to a new position $\mathbf{p}'(x', y')$. The position (x', y') is calculated by a mapping function defined as $f(x, y) = w_0 f_0 + w_1 f_1 + w_2 f_2$, with $f_0(\mathbf{p}_0) = \mathbf{q}_0, f_1(\mathbf{p}_1) = \mathbf{q}_1$ and $f_2(\mathbf{p}_2) = \mathbf{q}_2$. The weights satisfy the conditions $w_1 > w_0 > w_2$ and $w_0 + w_1 + w_2 = 1$.

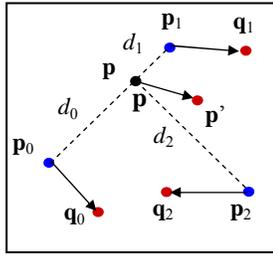


Figure 9 – Mapping of a pixel with inverse distance weighted interpolation method.

In order to obtain the metamorphosis of one image into another, we generate a progressive deformation of the source image and inverse deformation of the destination image. At the same time, a cross-dissolve of the deformed images generates the intermediate images. Therefore, the influence of the source image is attenuated while the influence of the destination image increases.

2. Reconstruction of 3D coronal loops using image morphing with Bezier curves

In order to obtain the 3D reconstruction of the loop, the deformation of the top and bottom images is controlled by a Bezier curve that approximates the shape of the loop (Figure 10).

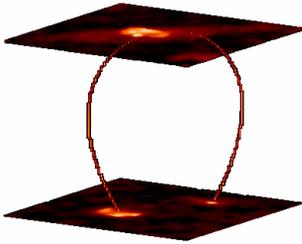


Figure 10 – Original images and Bezier curve.

We need to specify four control points that define the 3D Bezier curve. In Figure 11, points P_0 and P_3 correspond, approximately, to the loop footprints observed in the bottom image. Points P_1 and P_2 are adjusted to define the shape of the magnetic loop.

The parametric polynomial function, $P(t)$, of a cubic Bezier curve is expressed by:

$$P(t) = (1-t)^3P_0 + 3t(1-t)^2P_1 + 3t^2(1-t)P_2 + t^3P_3, \quad (3)$$

where t is the normalized parameter of the curve, with values range between 0 and 1; P_0 , P_1 , P_2 and P_3 are control points with Cartesian coordinates x , y and z .

For each point $P(t)$ of the curve, the tangent vector is:

$$\vec{P}(t) = \frac{P'(t)}{\|P'(t)\|}, \quad (4)$$

where $P'(t)$ is the first derivative at the point $P(t)$.

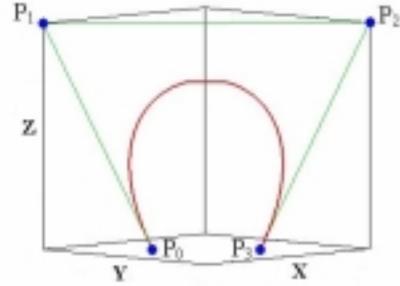


Figure 11 – Bezier curve for approximation of the tridimensional loop.

We perform the image morphing using the inverse distance weighted interpolation method to deform the original images, and a cross-dissolve of the deformed images to generate the intermediate images [14]. Two points P_{c0} and P_{c1} with the same coordinate z in the Bezier curve, are used as control points in a slice, for the partial deformation of the top and bottom images.

For each control point P_{c_i} of a specific slice, a local interpolator $f_i: \mathbb{R}^2 \rightarrow \mathbb{R}^2$ is specified. We defined the interpolator f_0 and f_1 at the control points P_{c0} and P_{c1} by:

$$f_0 = 1 - \vec{P}_{c0} \quad \text{and} \quad f_1 = 1 - \vec{P}_{c1},$$

where \vec{P}_{c0} and \vec{P}_{c1} are the tangent vector to the Bezier curve at the points P_{c0} and P_{c1} . Figure 12 shows the warping parameters using the Bezier curve.

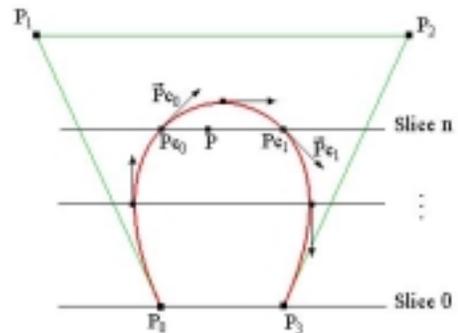


Figure 12 – Warping parameters represented by a Bezier curve.

The inverse distance weighted interpolation function $f(x, y)$ is a weighted average of the local functions f_0 and f_1 , given by:

$$f(x, y) = w_0(x, y)f_0(x, y) + w_1(x, y)f_1(x, y).$$

On a slice, the weight function $w_i: \mathbb{R}^2 \rightarrow \mathbb{R}$ depends on the inverse distance from a pixel $P(x, y)$ to the control point $Pc_i(x_i, y_i)$, normalized by the sum of the inverse distances of all control points. So, the weight w_0 is given by:

$$w_0(x, y) = \frac{d_0(x, y)}{d_0(x, y) + d_1(x, y)},$$

where $d_0(x, y)$ and $d_1(x, y)$ are the inverse distances from the pixel $P(x, y)$ to the control points Pc_0 and Pc_1 , respectively.

In order to synthesize the deformed images, each pixel at the position (x, y) of the top or bottom image is moved to a new position (x', y') at the intermediate slice n , given by:

$$(x', y') = (x, y) + \alpha(\Delta x, \Delta y),$$

being α the inverse distance weighted interpolation function, and $(\Delta x, \Delta y)$ a displacement that depends on the nearest control point.

The displacement $(\Delta x, \Delta y)$ for each point at the slice is defined as the distance xy between the nearest control point in the slice n and the corresponding control point in the bottom image, for deformation of the bottom image (or in the top image, for deformation of the top image).

As shown in Figure 13, the control point in the bottom image is mapped to a new position in the slice n , with a displacement Δxy . The pixels that are most distant from the control point are displaced with a smaller Δxy , weighted by the interpolation function. The same procedure is carried out for the top image.

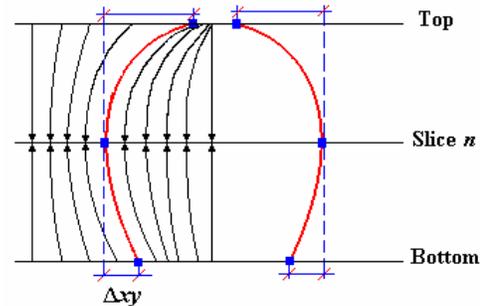


Figure 13 – Mapping of pixels in the warping process with Bezier curve

Simultaneously, we generate the intermediate images of the loop with a cross-dissolve of the deformed top and bottom images. Figure 14 illustrates this process, showing some deformed slices of the top and bottom images, their respective control points and the intermediate slices generated.

Due to the need of obtaining the 3D reconstruction of the coronal loops within a reasonable time, the reconstruction method was implemented to execute in the Atlas system (see section III).

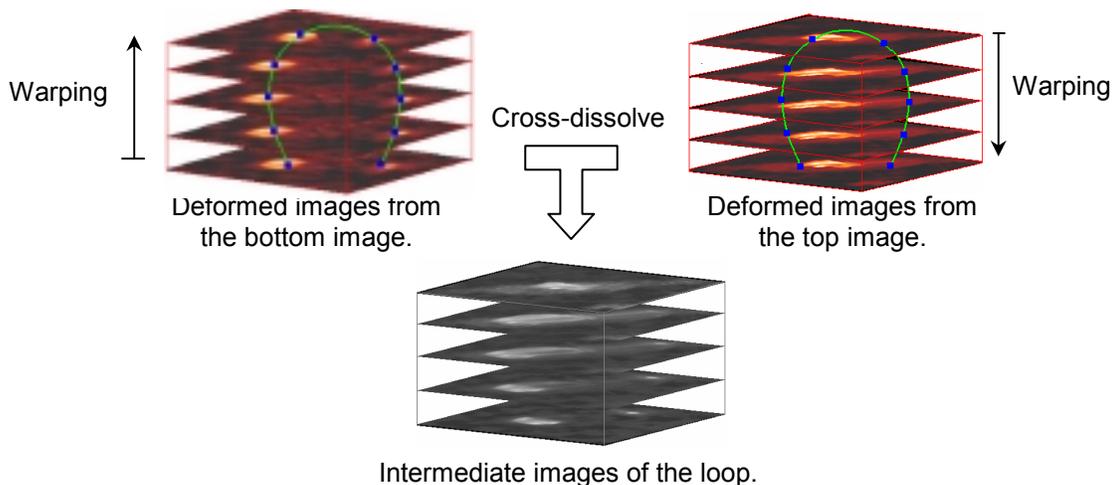


Figure 14 – Image morphing process: warping + cross-dissolve.

3. Parallel program for 3D reconstruction of coronal loops

The application for 3D reconstruction of coronal loops is composed by a main program running in the host PC of the Atlas parallel system, and a 3D reconstruction parallel program running in the four DSPs. The communication between these programs is carried out through a bi-directional communication pipe.

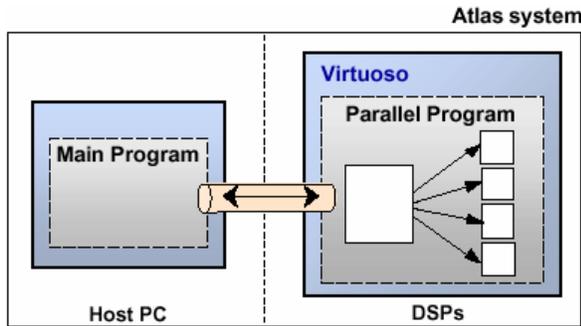


Figure 15 – Structure of the parallel application.

The main program offers a user graphical interface and an interface for 3D visualization of the coronal loop.

The 3D reconstruction parallel program, implemented in the programming model based on tasks and channels, is divided into five tasks (MASTER, TASK1, ... , TASK4).

Task MASTER distributes the original images of the loop between the tasks TASK_{*i*}, where each task TASK_{*i*} executes in a different processor, in order to simultaneously generate a subgroup of intermediate images.

The tasks are interconnected by channels (PIPE, CHANNEL1, ... , CHANNEL4). The external channel (PIPE) allows a bi-directional communication between the main program at host PC and the tasks on the DSPs of the parallel machine. The other channels carry out the inter-processor or intra-processor communication between tasks. Figure 16 illustrates the data flow of the 3D reconstruction application.

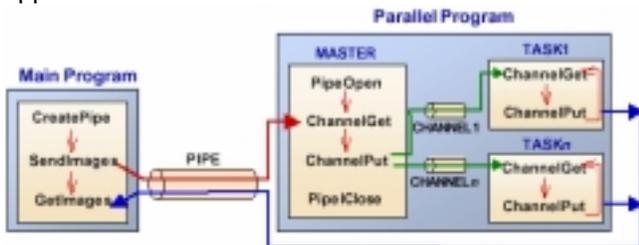


Figure 16 – Flow data of the 3D reconstruction application.

When the main program is executed, a pipe for external communication is created, and the parallel program initiates the execution. Task MASTER opens a connection with the pipe and waits the main program to send the original images of the loop.

When the user requests a 3D reconstruction, the main program sends the original images for task MASTER, who in turn, distributes these images between the other tasks using the internal channels. Each task TASK_{*i*} waits until receiving the original images to initiate the processing.

During the generation of the intermediate images subgroup, tasks TASK_{*i*} send the intermediate images to the main program through the communication pipe. The main program stores the intermediate images to obtain a volume dataset. After the 3D reconstruction, the tridimensional structure of the coronal loop can be visualized through an interface of volume visualization.

The parallel program was implemented using the Visual Environment for the Development of Parallel Programs. Figure 17 shows a simplified graphical representation of the parallel program in the Visual Environment, omitting TASK3, TASK4 and other objects such as memory pool and resources.

In the graphical description of the parallel program, the microkernel objects – tasks, semaphores, channels, and so on – and the communication and synchronization service primitives offered by the Virtuoso, are graphically represented by rectangles and connection lines, respectively. The Visual Environment automatically generates the corresponding code for the connections. In addition, specific complementary code of the tasks can be written in the C/C++ language, using the environment’s text editor.

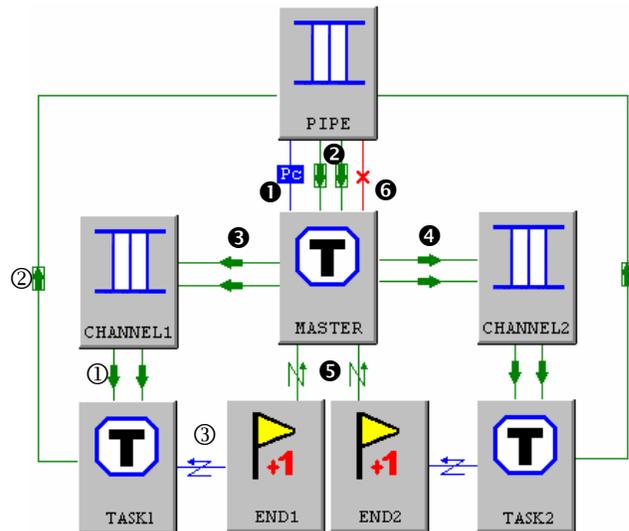


Figure 17 – Graphical representation of the parallel program in the Visual Environment.

Master

```
① Open PIPE channel;  
do {  
② Get original images from the external program through PIPE;  
③ Send original images to the TASK1 through CHANNEL1;  
④ Send original images to the TASK2 through CHANNEL2;  
⑤ Test semaphores END1 and END2;  
} while()  
⑥ Close PIPE channel;
```

Task 1

```
while(true){  
① Get original images from the MASTER through  
CHANNEL1;  
for(...){  
② Send the i-th intermediate image to the external  
program through PIPE channel;  
}  
③ Signal semaphore End1;  
}
```

Figure 18 – Algorithm of the parallel program in the Visual Environment.

After defining all the objects and writing the parallel application code, the microkernel objects can be easily distributed between the processors of the parallel machine, using the object properties editor offered by the Visual Environment. Tasks MASTER and TASK1 were mapped to DSP1, while tasks TASK2, TASK3 and TASK4 were mapped to DSP2, DSP3 and DSP4, respectively.

V. Results

The 3D loop reconstructed using image morphing with Bezier curve is shown in Figure 19. The control points locations of the Bezier curve are manually chosen in order to approximate the shape and size of the loop.

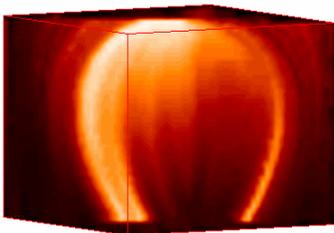


Figure 19 – 3D reconstruction of a loop using image morphing with a Bezier curve.

The images of Figure 20 show a comparison of the results of the 3D reconstruction of the loop using image interpolation with cross-dissolve, image morphing and image morphing using a Bezier curve.

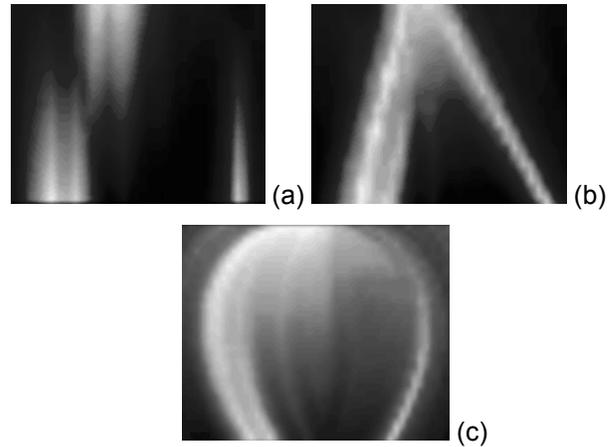


Figure 20 – Comparison of the 3D reconstruction methods: a) cross-dissolve; b) image morphing; c) image morphing with Bezier curve.

An advantage of the method for 3D reconstruction with Bezier curve is the capacity to obtain loops of various shapes and sizes, being enough to define the four control points for the cubic Bezier curve.

A twisted loop with a sigmoid shape can be easily reconstructed from the top and bottom images. Figure 21 shows the Bezier curve that defines the sigmoid shape for reconstruction of the twisted loop shown in Figure 22.

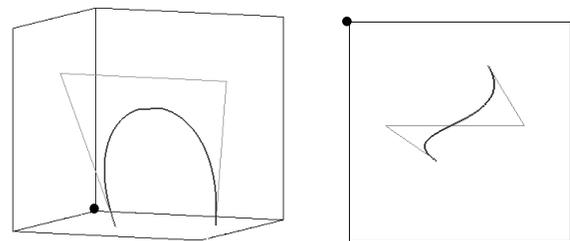


Figure 21 – Bezier curve with a sigmoid shape.

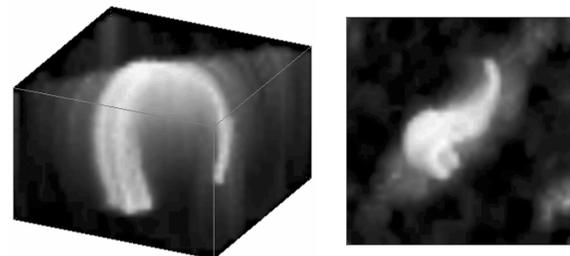


Figure 22 – Hypothetical sigmoid loop reconstructed from the X-ray tomographic images.

The 3D reconstruction parallel program was executed for performance analysis in the parallel machine with four processors. Table 1 shows the execution time, the efficiency and speedup of the 3D reconstruction program as the number of processors is increased.

Table 1 – Execution time (in seconds), efficiency and speedup of the parallel program.

Number of Processors	Time (s)	Efficiency (%)	Speedup
1	75.86	100.00	1.0
2	38.36	98.88	1.98
4	19.28	98.37	3.93

Due to increasing overhead and the communication bottleneck, the efficiency tends to decrease and the speedup does not increase linearly as the number of processors is increased. Nevertheless, the estimated values show that the algorithm is scalable allowing an increase in the numbers of processors.

VI. Conclusions

This paper presented the main topics of the PVA-BDA project. At first, the technical features of the Brazilian Decimetric Array – BDA was presented. Then, it was described the parallel systems aimed to be used at real-time 3D tomographic image reconstructions and analysis, and the visual environment for the development of such a parallel application programs. Then it was described a parallel 3D reconstruction method of coronal loops, showing its performance analysis. The PVA-BDA Project is a joint effort between the Department of Computer Science at Federal University of São Carlos (DC/UFSCar), the Astrophysics Division (DAS) and Associated Laboratory for Computing and Applied Mathematics (LAC) at INPE.

The 3D Reconstruction using Atlas System with 4 processors spent 19.28s, with the speedup of 3.93, nearly linear, as shown in Table 1. As a preliminary analysis, using the high performance cluster of 32 processors, and considering the same time of 75.86s with 1 processor, and an ideal speedup, the processing time will be of 2.37s, which is acceptable to support realistic applications, involving a reasonable amount of parallel processing, in order to carry out the processing, visualization and analysis of solar images captured by BDA.

When the solar images became available, in the beginning of the next year, the same techniques applied with satellite images would be applied to BDA images. So the complete and real-time processing,

visualization, and the analysis of solar images will be available.

PVA-BDA Project has been an important mechanism to the joint effort between the institutions involved in the development of a system of solar analysis.

Acknowledgements

We thank the financial support offered by the three Brazilian government agencies: FAPESP (Foundation for the Support of Research for the State of São Paulo), FINEP (Financing Body for Studies and Projects) and CNPq (National Council for Scientific and Technological Development).

References

- [1] EONIC *Atlas system user guide: Atlas2-HS V1.1*, Eonic Solutions (2001).
- [2] Silva, A. F., Rosa, R. R., Roman, L.S., Veje, E., Pepe, I.: *Solid State Comm.* 113 (2000) 703.
- [3] Marubashi, K.: The space weather forecast program, *Space Science Reviews*, 51 (1989) 197.
- [4] Morón, C. E. et al: A visual environment integrating design, implementation and debugging in parallel real-time systems. *12th Brazilian Symp. on Computer Architecture and High Performance Computing, SBAC-PAD, Brazil*, (2000) 313-319.
- [5] Morón, C. E. et al: Parallel architecture using DSPs. *Proc. 9th Brazilian Symp. on Computer Architecture and High Performance Computing, SBAC-PAD'97, Brazil*, (1997) 605-608.
- [6] Ribeiro, J. R. P., Silva, N. C., Morón, C. E.: A Visual environment for the development of parallel real-time programs. *Lecture Notes in Computer Science*, 1388: (1998) 994-1014.
- [7] Rosa, R. R., Sharma, A. S., Valdivia, J. A.: Characterization of asymmetric fragmentation patterns in spatially extended systems. *Int'l J. Modern Physics C* 10 (1) (1999) 147-163.
- [8] Rosa, R. R., Sharma, A. S., Valdivia, J. A.: *Physica A* 257 (1998) 509.
- [9] Rosa, R. R., Campos, M. R., Ramos, F. M., Fujiwara, S., Sato, T.: *Brazilian Journal of Physics* 33 (3) (2003) 605,
- [10] Rosa, R. R., Sawant, H. S., Morón, C. E., Mascarenhas, N. D. A., Saito, J. H.: Gradient Pattern Analysis of Complex Solar Coronal Loops, *Proc. da 30a Reunião Anual da Sociedade Astronômica Brasileira*, (2004) 34-35.

- [11] Rosa, R. R. et al.: Emission measurement density curves for regularization of coronal loop tomography. *Proceedings of HESSE*, NASA (1999).
- [12] Rosa, R. R. et al.: Normalized emission measurement density curves for construction of X-ray loop tomography. *ASP Conference Series*, 206 (2000) 293-296.
- [13] Rosa, R. R. et al.: Phenomenological dynamics of coronal loops using a neural network approach. *Advances in Space Research*, 25 (9) (2000) 1917-21.
- [14] Ruprecht, D., Müller, H.: Deformed cross-dissolves for image interpolation in scientific visualization. *The Journal of Visualization and Computer Animation*, 5 (3) (1994) 167-181.
- [15] Tsuneta, S., Lemen, J. R.: Dynamics of the solar coronal observed with the Yohkoh soft X-ray telescope. *Physics of Solar and Stellar Coronae*, (1993) 113-130.
- [16] Uchida, Y.: New aspects of solar coronal physics revealed by Yohkoh. *Physics of Solar and Stellar Coronae* (1993) 97-112.
- [17] VIRTUOSO™ *User guide for version 4.2*. Wind River Systems (2001).
- [18] Wolberg, G.: *Digital image warping*. IEEE CS Press, Los Alamitos, CA (1990).
- [19] Wolberg, G.: Image morphing: a survey. *The Visual Computer Journal*, 14 (1998) 360-372.
- [20] Sawant, H. S., Neri, J. A. C. F.: Status of Development of the Brazilian Decimetric Array. In: *7 Latin-American Conference on Space Geophysics (VII COLAGE)* (2004).