Radar Image Processing with Clusters of Computers

Alois Goller[1] and Franz Leberl

Institute for Computer Graphics and Vision
Technical University Graz
Inffeldgase 16, A-8010 Graz, Austria
Leberl@icg.tu-graz.ac.at

[1] Currently at Chalmers University
Göteborg, Sweden
algo@chl.chalmers.se

Abstract - Some radar image processing algorithms such as shape-from-shading are particularly compute-intensive and time consuming. If, in addition, a data set to be processed is large, then it may make sense to perform the processing of images on multiple workstations or parallel processing systems. We have implemented shape-from-shading, stereo matching, resampling, gridding and visualization of terrain models in such a manner that they execute either on parallel machines or on clusters of workstations. We were motivated by the large image data set from NASA's Magellan mission to planet Venus, but received additional inspiration from the European Union's Center for Earth Observation program (CEO) and Austria's MISSION initiative for distributed processing of remote sensing images on remote workstations, using publicly accessible algorithms.

We have developed a multi-processor approach that we denote as CDIP for Concurrent and Distributed Image Processing. The speedup for image processing tasks increases nearly linearly with the number of processors, be they on a parallel machine or arranged in a cluster of distributed workstations.

Our approach adds benefits for users of complex image processing algorithms: the efforts for code porting and code maintenance are reduced and the necessity for specialized parallel processing hardware is eliminated.[1]

TABLE OF CONTENTS

[1] 0-7803-5846-5/00/$10.00© 2000 IEEE

1. INTRODUCTION

As we experience the enormous changes caused by the Internet, we also see an ever-growing flood of data, particularly visual data, and specifically in the field of satellite remote sensing. This causes an increasing need to process large data sets quickly, thereby employing increasingly complex and demanding algorithms to perform very specific tasks.

When put together, these developments inspire initiatives to provide users with tools to take advantage of large remote sensing data sets over the Internet, to employ algorithms operating on remote computers, and to provide access to remote experts. When processing time is excessive, then it makes sense to employ parallel computers, or clusters of computers, to speed up the creation of the result. We have thus a motivation for the study of web-based image processing in remote sensing. Specifically we are reporting on work inspired by thoughts to reprocess the 400 Gbytes of image data from NASA's Magellan mission to planet Venus. However, one can see this Magellan-challenge as a model for things to come as part of the increasing data stream from the Earth Observation System EOS, and from the European Union's Center for Earth Observation CEO.

We have developed a concept and system to employ multiple computers, some of them massive parallel processing machines, to process large quantities of radar image data using very specific algorithms and doing this at a very high throughput [2] and [3].

Many advantages result from successfully implementing such technology. There is not only the obvious decrease in elapsed time before a particular computing task is completed; there are also simplifications for the user. He might not need, nor desire, to always understand where the data or software actually reside. And there is the increased ease of creating and supporting oftentimes complex software that executes best on a particular computing environment, and needs local expert support not available at a particular user's site.

We show that the use of remote computers, either in clusters or specific parallel architectures, together with remotely residing software and large data sets can successfully be combined to a system with a nearly linear increasing speed-up and throughput as the number of processors increases. It may take 2 hours to obtain a certain Digital Elevation Model (DEM) from overlapping radar images on one computer, and we may see this reduced to a mere 10 minutes using multiple computers.

2. ALGORITHMS

Processing of radar images begins with the image formation from the raw signals. Particularly time consuming radar-grammetric algorithms are then shape-from-shading [6], stereo matching [4], gridding of elevation models, re-sampling for ortho photos and terrain visualization and rendering. Our work has addressed all of these algorithms [2], but focussed on shape-from-shading and stereo matching. Figure 1 illustrates the data flow through a system for processing radar images, in particular those from NASA's Magellan mission.

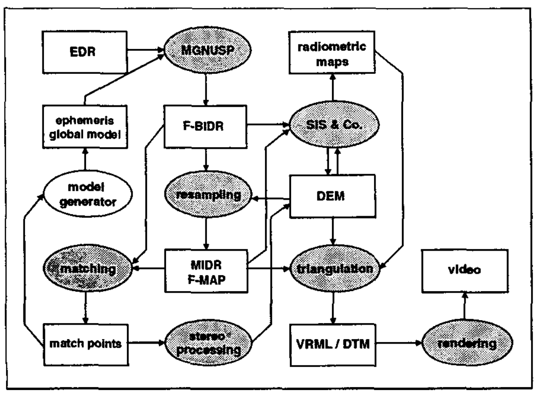

Figure 1 Key radar image processing algorithms for Magellan processing, beginning with signal processing into Full-Resolution Basic Image Data Records (F-BIDRs), then producing Mosaicked Image Data Records

Table 1 Typical image processing algorithms and their performance on a single processor computer, in this case often an SGI Indy.

MGNUSP SAR Image Formation	6 hours per F-BIDR
Resampling	a few minutes
Image Matching	several tens of minutes
Stereo Processing	a few minutes
Shape-from-Shading	1 hour
DEM Surface Triangulation	some minutes
Rendering and Visualization	from seconds to hours

Separately marked in Figure 2 are typical iterative data flows connected with the extraction of digital terrain models. A sequence of stereo matching and shape-from-shading produces the most detailed representation of the terrain. Figure 3 shows an example of stereo matching and refinement by the exploitation of shading variations.

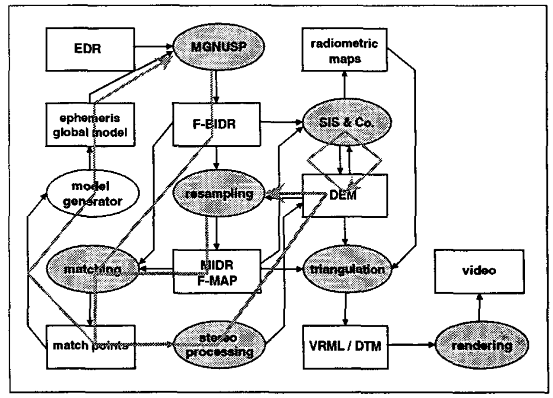

Figure 2 Some iterative processing sequences added to Figure 1. To the left is the sequence that produces improved F-BIDRs after identifying match points. In the center there is the sequence to refine the DEM by improving the input images by geometric resampling. To the right is the iterative use of shape-from-shading.

Table 1 summarizes typical performance and throughput times for some of the algorithms. It is to be noted that in the case of the Magellan data, a systematic process will involve multiple images each of a size in the 100 Mbyte range, while some of the performance assessments are based on patches of 1k x 1k pixels.

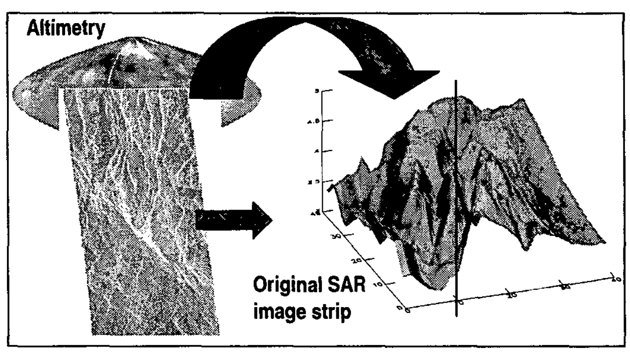

Figure 3 Concept of *Digital Elevation Modeling* from Magellan radar images, using stereo matching and shape-from shading. The final DEM evolves from a preliminary altimeter data set, then improves via stereo matching into a refined DEM (left half of the 3D diagram), followed by shape-from-shading to produce the final result (right half of 3D diagram).

3. PARALLELIZATION

Data decomposition is the obvious approach for parallel processing or processing on multiple computers. Figure 4 illustrates the concept that is applicable if processing for an image location needs only pixels in a limited Region-of-Interest (ROI). Patches in the decomposition should overlap in cases where certain irregularities occur along the edge of a patch: in matching, the edges cannot be processed due to the template-based algorithm, and in SfS, the edge of a patch displays so-called "ringing" effects. Overlaps permit one to eliminate such problem areas.

The Message Passing Interface [5] was used to organize the parallel execution of the algorithms. Patches for SfS comprise 128 x 128 pixels; decomposition is thus an ideal paradigm for parallelization. The Fast Fourier Transform becomes local on every patch, reducing the time complexity from $O(n^2\log(n))$ to $O(n^2)$.

Managing the decomposed data sets is based on the "Manager/Worker" or "Task Farm" approach. The manager is one processor among the many to load and store all data, preparing the patches and communicating them to the other processors. He is also responsible for balancing the load among the remaining processors and for post-processing the patches (i.e. eliminating the edge irregularities).

4. CDIP ARCHITECTURE

The Internet changes everything, and offers numerous new ways of dealing with computing resources, expert knowledge, algorithms, and data sets. Adding Java and

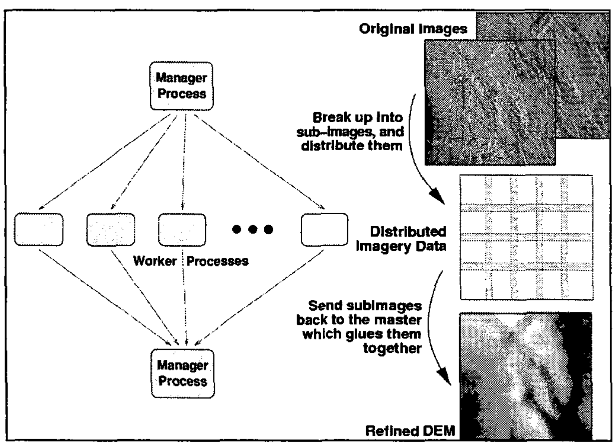

Figure 4 Parallelization by segmenting the data sets into patches and processing patches in parallel (data decomposition).

NetSolve opens an array of new options in using specialists and their image processing knowledge, as well as data sets located at remote sites, at one's own desktop. That is the model for the Center for Earth Observation (CEO) of the European Union's Joint Research Center.

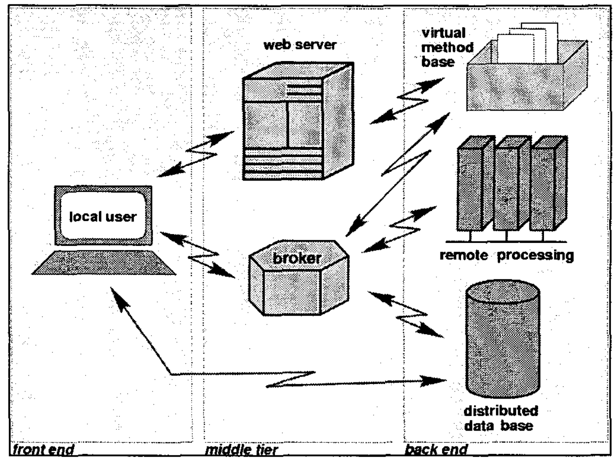

Figure 5 Basic architecture for a Concurrent and Distributed Image Processing (CDIP) scheme.

Netsolve [1] makes computing resources available to every user anywhere in a network. The basic building blocks of Netsolve are shown in Figure 5. There are a front end, a middle tier containing a broker and a web server, and a back end. Figure 6 illustrates the graphical user interface of the front end using the Magellan example.

While the front end is a Java applet running within a web browser, the back end consists of the computing resources such as a cluster of workstations, shared-memory multi-processors, or any other type of powerful computing arrangement.

The broker is the critical and intelligent segment of CDIP. During a session, it first acts as an agent that collects execution parameters of all available algorithms from the back end and presents them to the user via the front end.

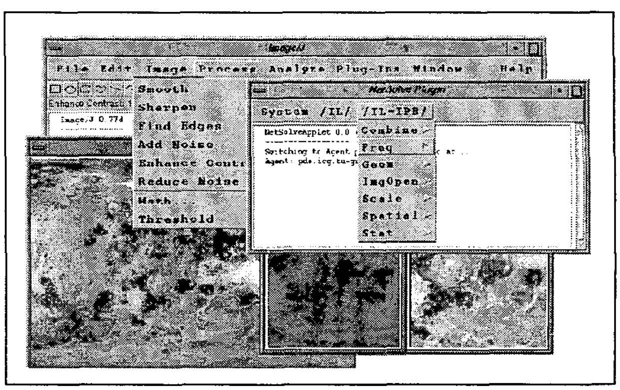

Figure 6 Example of a simple graphical user Interface for CDIP, using Magellan data.

Then, acting as a scheduler, it receives user's processing desires, analyzes them, and finally runs the proper algorithms at the back end. To do so, the broker contacts a method base, which may be a simple database or a sophisticated decision support system.

The entire configuration addresses inter-operability, a unified access to a variety of platforms and algorithms, method databases with various image processing functions, queuing and scheduling by means of meta information, and interfacing with the user.

5. COMPARISONS

We were able to employ the following computers:

SGI Power Challenge with 20 nodes,
Intel Paragon with 56 nodes,
MEIKO CS2 with 128 nodes,
Cray T3D with 512 nodes,
Cluster of Workstations (CoW) with 22 nodes.

Bringing up the various algorithms on these different machines illustrated the importance of being machine independent. Table 2 describes the various ways in which the result of parallelization can be assessed. "Speedup" S(p) is the most commonly used parameter to describe the effect of parallelization, and it is computed as a ratio of execution time T(1) using 1 processor and T(p) using p processors.

Table 2 Evaluation parameters for the usefulness of parallel computing, for example assessing the efficiency α as a measure of the increase in speedup as more processors get used.

Wall-clock time T, CPU time T_{CPU},
Speedup $S = T(1) / T(p)$,
Efficiency $\alpha = S / p$,
Efficacy $\eta = S^2 / p$
Scaleup $\gamma = T(1,d) / T(p, p*d)$

Figure 7 presents the speedup achieved with various systems, addressing stereo matching and shape-from-shading. Experiments were performed with images of 1k x 1k pixels (121 patches), and larger size.

5.1 Matching

On a Cray T3D, we found the speedup for *matching* to be in the range of about 3 when 4 processors were being used, but it is less effective as more processors are added. The efficacy is at 2, the efficiency less than 0.5. All three parameters become smaller as more processors get added, since data transfer speed limits what can be accomplished: the 10 Mbit/s Ethernet connection to the disk was far too slow. On the Power Challenge and other machines those limits were less prevalent.

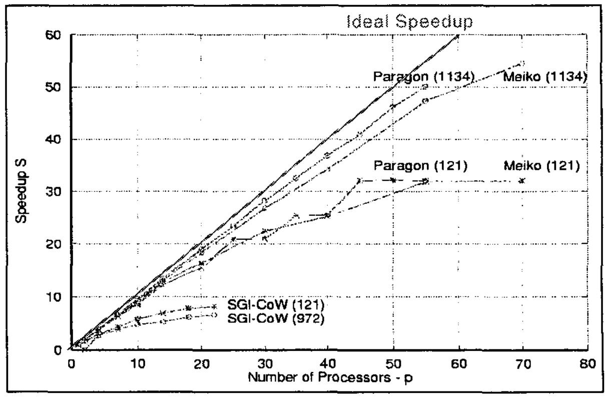

Figure 7a Speedup of SfS using multiple processing nodes, either on a parallel processing computer, or by using multiple workstations in a cluster. The image size is 1k x 1k pixels (121 patches) and 7k x 8k pixels (1134 patches), respectively.

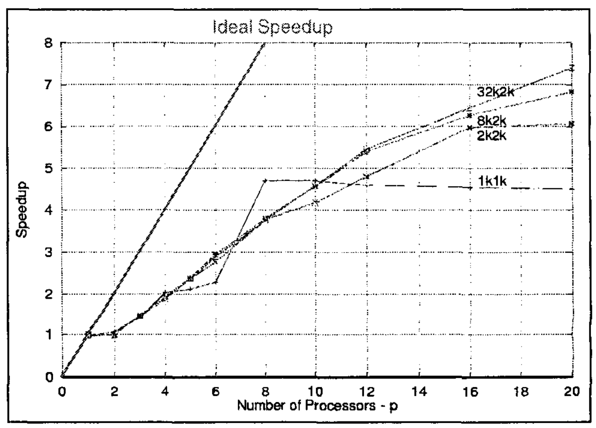

Figure 7b Speedup of matching on an SGI Power Challenge. The image size is given in pixels.

5.2 Shape-from-Shading

For SfS the time used to process a single patch was about 120 seconds using a single processor of Paragon or Meiko, and 500 seconds on an Indy. The speedup is maintained linear as more processors are added, until the number of processors exceeds about ¼ of the number of patches and load imbalancing starts to become visible.

Figure 3 described an example of shape-from-shading computation for a Magellan area. Instead of waiting an hour to obtain a result on a single processor workstation for a 2k x 4k DEM, the approach used here reduces the waiting time to 12 minutes using a cluster of workstations.

6. Outlook

Internet-based advantages in remote sensing image processing have become an intensively researched subject in major organizations. One of the difficulties of accepting remote sensing lies in the complex analysis methods, the need to have expert knowledge about these methods, the many locations at which data are stored, and the need of the individual users to have the results quickly and at their own desktop computer. This cannot be accomplished without the ability to process data remotely and quickly. Multiple processors in one or more computers can be used without the Internet. The conveniece is in accessing the data at many locations.

CDIP has been developed to learn how to support the user with remotely located data and remotely available image processing methods running on parallel computers, thereby exploiting the possibility of concurrent execution of algorithms. We found that with Java and Netsolve we could indeed achieve an ease of access and a gain in efficiency and throughput in remote sensing image processing almost linear with the increase of processors, using Magellan data of planet Venus as the data set to experiment on. The key algorithms were shape-from-shading and stereo-matching, two methods that are to be intimately connected in SAR image analysis to produce detailed and accurate Digital Elevation Models.

References

[1] Casanova H., J. Dongarra (1997) *NetSolve: A Network Server for Solving Computational Science Problems.* Int'l J. of Supercomputer Applications and High Performance Computing, Vol. 11, No. 3, pp. 212-223.

[2] Goller A. (1999a) *Parallel and Distributed Processing of Large Image Data Sets.* Doctoral Diss., Technical Univ. Graz, Austria.

[3] Goller A. (1999b) *Parallel Strategies for Large SAR Image Data Sets in a Distributed Environment.* Computing, Vol. 62, No. 4, pp. 277-291.

[4] Hensley S., S. Shaffer (1994) *Automatic DEM Generation Using Magellan Stereo Data.* IGARSS 94, Pasadena CA., pp. 1470-1472.

[5] MPI (1996) *The Message Passing Interface – Standards Specification V1.1.*

[6] Thomas J., W. Kober, F. Leberl (1991) *Multiple Image SAR Shape-from-Shading.* Photogrammetric Engineering and Remote Sensing. Vol. 57, No. 1

***Alois Goller** is researcher and lecturer at the Department of Electrical and Computer Engineering, Chalmers Lindholmen University College, Chalmers University of Technology, Göteborg, Sweden. He participates in research projects and teaches subjects related to image processing, parallel programming, distributed systems, computer networks and telecommunication. He received his MSc. in 1996 and his Ph.D. in 1999 from Graz University of Technology, Austria.*

***Franz Leberl** is professor of computer graphics and image processing at the Technical University Graz. His education was in photogrammetry, with a Ph.D. in 1972 (Vienna) and a Habilitation in Graz (1976). His work is in academia (Graz) and in business (Colorado). He is a Fellow of the IEEE (1996).*