

# Interferometric Synthetic Aperture Radar Processing on a Massively Parallel Supercomputer

C. Yerkes, E. Webster, and P. d'Arnaud  
Naval Command, Control, and Ocean Surveillance Center  
Research, Development, Test and Evaluation Division  
San Diego, CA

## Abstract

*In recent years interferometric synthetic aperture radar (IFSAR) processing technology has made available the ability to generate high accuracy digital terrain elevation (DTE) data. As with most SAR applications, the computation involved is very intensive. Current generation massively parallel processors (MPP) provide the high performance required to do scalable SAR processing at a relatively modest cost. In this paper we explore the performance of one implementation of an IFSAR DTE processor on a MPP.*

## 1 Introduction

High resolution topographic elevation maps of the earth have traditionally been produced using a labor intensive procedure of geometrical comparison of two moderate resolution photographic images. Resolution in elevation of maps produced by such procedures is often only as good as on the order of hundreds of meters and turn around time on production is on the order of weeks.

Current interferometric synthetic aperture radar technology facilitates the production of very high resolution digital terrain elevation maps in a relatively fast and efficient manner. Commercial IFSAR technology yields elevation resolution on the order of meters with a turn around time which is theoretically on the order of days for what took weeks by traditional methods. This technology could be useful in such commercial applications as rapid damage assessment for natural disasters such as earthquake or flood, or for rapid production of DTE for rough survey maps for large construction projects such as highways or railways.

The generation of IFSAR DTE requires the formation of a stereo pair of synthetic aperture radar (SAR) images. Digital production of SAR imagery has long been known to be very computationally demanding. Current SAR processors require on the order of several hundred to several thousand computer operations per output image pixel. This translates to on the order of a hundred million to several billion operations to produce a single 1024 by 1024 frame of radar imagery. One would need to produce many, many such images 'on the fly' to facilitate an IFSAR DTE production capability.

The computational load of SAR processing has traditionally dictated hardware and software design of

special purpose signal processors which are very expensive and have short life cycles. Current generation massively parallel supercomputers are approaching the performance level required to accommodate near real-time production of SAR imagery. The desirable feature of an MPP platform as opposed to a special purpose design is its relatively modest cost, its almost built in feature of scalability to problem size, and its potential for long term hardware and software support.

In this paper we explore an implementation of an IFSAR DTE processor on an Intel Paragon XP/S-25 MPP. The SAR processor itself is the ground processor for the NASA TOPSAR airborne interferometric synthetic aperture radar which flies on the NASA DC-8 [4]. The SAR image generation and elevation estimation algorithms were developed by the NASA Jet Propulsion Laboratory (JPL). Section 2 provides a short description of the MPP system used. Section 3 gives a brief background on the theory of IFSAR DTE production. In Sections 4 and 5 we discuss the particular implementation on the MPP architecture and our performance results, respectively. In Section 6 we give concluding remarks and a short discussion of future work.

## 2 MPP Platform

The Intel Paragon XP/S-25 supercomputer is a distributed memory multiprocessor, with the nodes connected by a 2-D mesh network. Each node contains two Intel i860-XP microprocessors (with one processor used for computations and the other as a communication co-processor). For the results presented in this paper, the Paragon system used had not received the hardware upgrade that allows use of the communication co-processor. When the upgrade is performed, improved system performance should be observed for many applications.

The system used in this study contains 336 compute nodes; with 68 nodes containing 32MB of memory each, and the other 268 nodes each containing 16MB of memory.

## 3 IFSAR DTE Production

As was alluded to above, IFSAR DTE is produced by analyzing a stereo pair of SAR images. A diagram of the data collection geometry is shown in Figure 1.

Here the locations of the two antennas illuminating a patch of

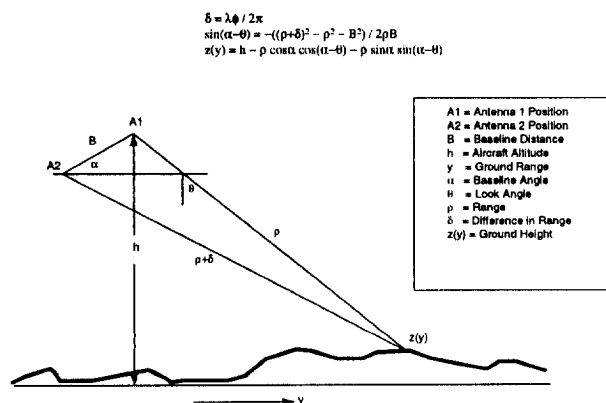


Figure 1: IFSAR Geometry

terrain are indicated by A1 and A2. The difference in the pathlengths,  $\delta$ , between the point  $z(y)$  on the ground and the two antennas can be measured in units of wavelengths of the radar energy,  $\lambda$ , as the phase angle,  $\phi$ , between the two return radar signals divided by  $2\pi$ , or

$$\delta = \frac{\phi}{2\pi} \lambda .$$

Using the law of cosines we get

$$\sin(\alpha - \theta) = \frac{-((\rho + \delta)^2 - \rho^2 - B^2)}{2B\rho}$$

We can also see from simple geometry that the elevation of the point  $z(y)$  is equal to the difference in the height of antenna A1 and  $\rho$  times the cosine of the angle  $\theta$ , or in terms of the expression above

$$z(y) = h - \rho \cos(\alpha) \cos(\alpha - \theta) - \rho \sin(\alpha) \sin(\alpha - \theta)$$

As long as the position and attitude of the aircraft are known fairly well, the relative elevation of the illuminated terrain can be computed with the measurement of the phase difference between the signals received by the stereo pair. A more complete discussion of the theory of interferometric SAR can be found in [1], [2], and [3].

A block diagram of the IFSAR DTE processor is shown below in Figure 2. The radar data from the two antennas must first be compressed into a SAR image. A standard range-doppler algorithm is used in the TOPSAR processor. A detailed discussion of the SAR processor in terms of block functionality will be given in Section 4. After formation of the images a cross correlation estimate on a pixel by pixel basis is performed to estimate the phase difference (modulo  $2\pi$ ) between the two signals.

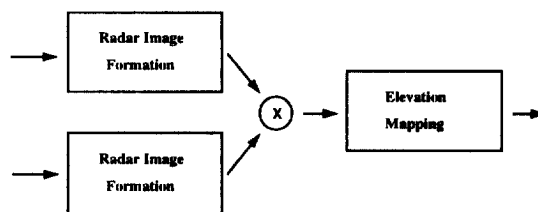


Figure 2: Block Diagram TOPSAR Processor

Once the phase estimate has been made, the data is passed to the portion of the processor which estimates height from phase. The fact that the phase difference is only known modulo  $2\pi$  creates a problem since the derivation dictates that the 'unwrapped' relative phase be known. The key to the success of IFSAR DTE is in the development of high precision radar hardware and sophisticated phase 'unwrapping' algorithms for 2 dimensional data sets. A complete description of the phase retrieval and height estimation algorithms for the TOPSAR processor is given in [2] and [3].

Of the two different portions of the processing shown in Figure 2 (imaging and elevation mapping) the radar imaging is the one which requires the greatest throughput by a large factor. The radar imaging portion basically consists of a sequence of separable two dimensional linear filtering and interpolation steps and is therefore very well suited to the two dimensional mesh architecture of the Intel Paragon. The imaging portion of the processing consumed roughly 85 percent of the total computational load on a shared memory vector supercomputer. The elevation mapping portion is much less computationally demanding but its load is very data dependent.

## 4 Implementation on an MPP

A block diagram of the range-doppler radar imaging processors in Figure 2 is shown in Figure 3. The raw radar phase data is fed to an azimuth presummer followed by a range compression (dechirping) step. Next the data is motion compensated to correct for deviations of the actual flight trajectory from the assumed flight trajectory. A doppler centroid estimate is derived from the data next to estimate the level of unwanted squint in the radar beam induced by deviations in the aircraft attitude. The azimuth processing is composed of a transformation into the frequency domain in azimuth, a range walk correction, a frequency domain matched filtering (compression) in azimuth, and a transformation back into the spatial domain in azimuth. Finally the image is post filtered in azimuth to achieve the correct resolution for the height estimation.

The SAR processor diagramed in Figure 3 is designed to run in a block processing strip mode. That is, it produces a continuous strip of imagery by per-

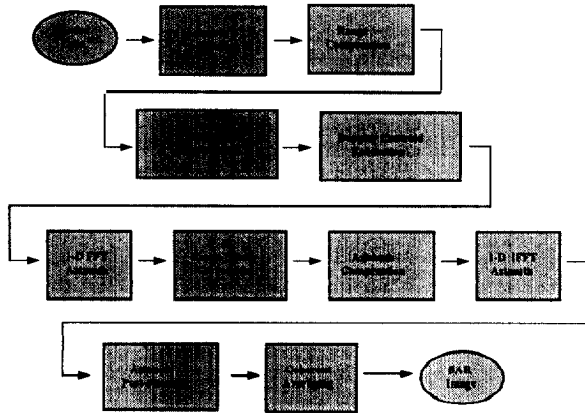


Figure 3: Range Doppler Processor

forming fast convolution in azimuth on sequential blocks of data. In any application where frequency domain fast convolution is used, the data processed in the blocks must overlap to facilitate removal of gross aliasing effects.

The SAR processor was mapped onto the two dimensional mesh architecture of the Intel Paragon in a scalable data parallel fashion. It was required that the block nature of the processing mentioned above be integrated into the data parallel mapping. A block diagram of the data partition at each sequential step in the range-doppler processing is shown in Figure 4 for a hypothetical four node configuration. The enclosing rectangle at each step represents the total radar data matrix at that step. The smaller horizontal or vertical rectangles within the enclosing rectangle indicates the partitioning of the radar data matrix among the four processors at that step. Note that at several steps in the processing only a subset of the total matrix is divided up among the nodes, and at one step each processor stores part of the array partitioned horizontally (by azimuth sample) and the other vertically (by range sample). These features were designed to accommodate efficient scalable overlapped block processing on the data parallel two dimensional mesh. Note also that during the motion compensation a copy of the range compressed data is saved for use in the next block. Each shaded rectangle within the total matrix in Figure 4 indicates used or useless data to be replaced by new data (read from external storage) at the next processing step.

The matrix transposes indicated in Figure 4 are the only portion of the radar processing besides I/O which require data communication between nodes. The transposition operation was implemented as shown in Figure 5 (again for a hypothetical four node configuration). The communication occurs in  $P - 1$  sequential stages in time, where  $P$  is the number of nodes ( $P$  is four in Figure 5). Each node both sends and receives a block of data at each stage. At the  $i^{th}$  communica-

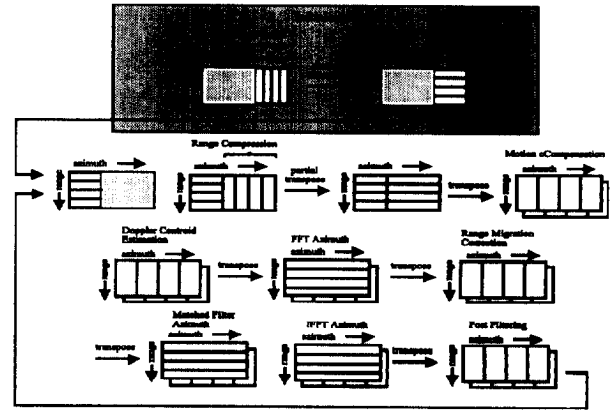


Figure 4: Block Processing Data Partition

tion stage, the  $n^{th}$  node,  $0 \leq n \leq P - 1$ , sends the  $mod((i \oplus n), P)^{th}$  sub-block of data from its memory to node  $(i \oplus n)$ . (The symbol  $\oplus$  represents the binary exclusive-or operation where the node number is represented as a positive base 2 integer). Each node then stores the received data block in its own  $mod((i \oplus n), P)^{th}$  memory sub-block. The last communication shown in Figure 5 is obviously not necessary and is shown only for completeness.

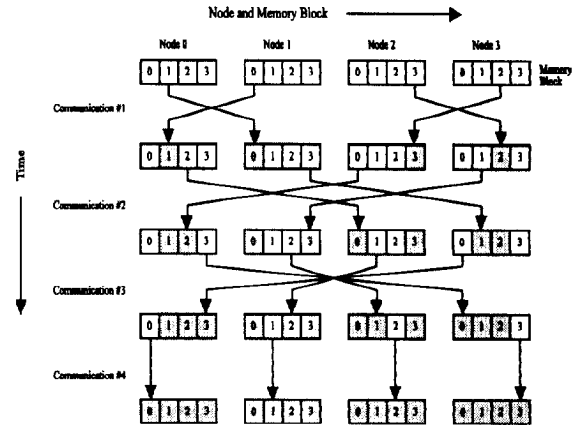


Figure 5: Transpose Communication

The algorithm for elevation estimation is inherently sequential and processing direction is very data dependent. It was therefore deemed un-feasible to perform elevation estimation in a data parallel fashion. In that light, the elevation mapping was performed in a non-scalable three stage pipeline. A high level block diagram for the full IFSAR DTE processor indicating the pipeline stages is shown in Figure 6. Note also that the memory requirements for each stage is given.

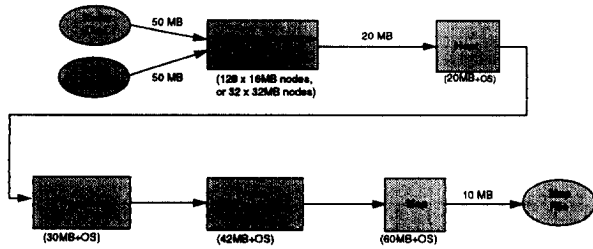


Figure 6: Full TOPSAR Processor

## 5 Results

### 5.1 SAR Processing

The performance of the SAR processor on the Intel Paragon MPP was assessed in terms of computation time, internode communication time, and total time (computation plus communication). The performance was measured for 32, 64, 128 and 256 node configurations. For each of the node configuration the measured quantities were averaged over 30 realizations (runs). The wall clock time for SAR image formation of a stereo pair of 2560 by 4096 pixel images is shown in the graph in Figure 7. Real time for the data collection of that image was about 13 seconds.

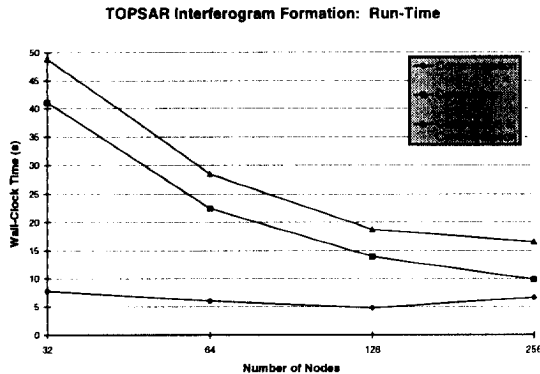


Figure 7: TOPSAR Interferogram Run-Time

The curves in Figure 8 show the speed-up with respect to 32 nodes for each node configuration for the SAR image formation. It is evident that although there is a performance improvement as the number of nodes increases, the speed-up is not quite linear as would be most desirable. The authors attribute this to the fact that much of the early processing had to be performed on overlapped data across nodes to accommodate the filtering in a data parallel fashion. This overlapping of computations is evident in the estimated floating point operation (FLOP) count for each con-

figuration in Table 1. We define a FLOP as one single precision floating point addition.

Table 1: Floating point operations for each compute configuration.

Nodes	FLOP's Per Node	FLOP's Total
32	7.97e+08	2.55e+10
64	4.22e+08	2.70e+10
128	2.32e+08	2.97e+10
256	1.38e+08	3.52e+10

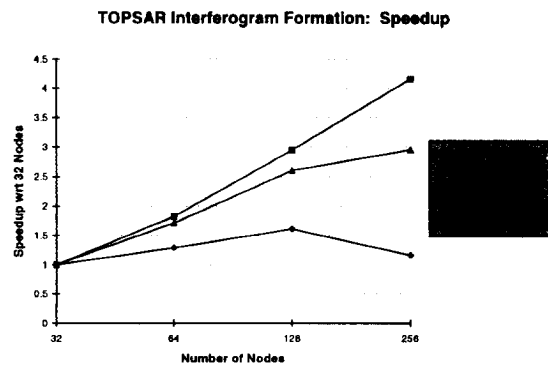


Figure 8: TOPSAR Interferogram Speed-Up

Each Intel Paragon node is rated at a peak FLOP rate of 100 MFLOPS (millions of floating point operations per second). The total MFLOPS per node observed on the SAR processor is shown graphically in Figure 9. We see that the total MFLOPS per node tapers off as the number of nodes used in the computation is increased. This can again be attributed to the overlap of the data in the early processing. The degree of overlap is dictated by the algorithms and is independent of the computation configuration. As the number of nodes increases the overlapped portion represents a larger fraction of the overall data set and therefore the FLOP rate is degraded.

### 5.2 Elevation Estimation

As was mentioned before, the elevation estimation portion of the processing was implemented as a three stage pipeline due to the inapplicability of data parallelism. As a result, it was not scalable. Although scalability is desirable, the fact that it was non-scalable does not severely degrade the results since the SAR processing overwhelmingly dominated the computation on a traditional vector machine. With the three stage pipeline design, we observed a roughly threefold speed-up on the elevation estimation over running it on a single Intel Paragon node. The observed wall

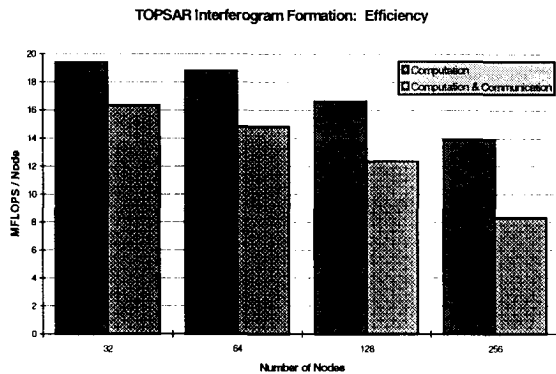


Figure 9: TOPSAR Interferogram Efficiency

clock time for the elevation estimation using the three stage pipeline was 40 seconds per image.

## 6 Conclusion

It has been shown in this paper that IFSAR DTE production can indeed be efficiently performed on an MPP platform. A roughly tenfold increase in speed was observed for the TOPSAR processor over previous implementations on traditional vector supercomputers.

The SAR processing portion of the computation showed a reasonable degree of scalability as the number of processing nodes was increased with the problem size held constant. It is well known that MPP's achieve better performance as problem size is increased. An interesting future investigation might be to observe the effect of increase in data size on scalability for the SAR processing problem.

## 7 Acknowledgments

The authors would like to thank the Radar Science and System Studies Group at NASA JPL, who provided the IFSAR and DTE algorithms, data for test cases, and much technical advice.

This research was funded by the Advanced Research Projects Agency, and was supported in part by a grant of High Performance Computing (HPC) time from the Department of Defense HPC Shared Resource Center, Naval Command, Control, and Ocean Surveillance Center, Research, Development, Test and Evaluation Division Paragon.

## References

- [1] H. Zebker and R. Goldstein, "Topographic Mapping From Interferometric Synthetic Aperture Radar Observations," *Journal of Geophysical Research*, vol. 91, pp. 4993-4999, 1986.
- [2] R. Goldstein, H. Zebker, and C. Werner, "Satellite radar interferometry: Two-dimensional phase

unwrapping," *Radio Science*, vol. 23, pp. 713-720, 1988.

- [3] S. Madsen, H. Zebker, and J. Martin, "Topographic Mapping Using Radar Interferometry: Processing Techniques," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 31, pp. 246-255, 1993.
- [4] H. Zebker et. al., "The TOPSAR Interferometric Radar Topographic Mapping Instrument," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 30, pp. 933-940, 1992.