

5
SAND98-0013C

SAND-98-0013C

Implementation of SAR Interferometric Map Generation Using Parallel Processors

CONF-980715--

19980529 056

Neall Doren
Daniel E. Wahl
Systems Assessment and Research Center
Sandia National Laboratories¹
P.O. Box 5800, Mail Stop 1207
Albuquerque, NM 87185-1207, USA

Voice: +1 505 845-8660 Fax: +1 505 844-4484 Email: nedoren@sandia.gov

RECEIVED
APR 10 1998
OSTI

ABSTRACT

Interferometric fringe maps are generated by accurately registering a pair of complex SAR images of the same scene imaged from two very similar geometries, and calculating the phase difference between the two images by averaging over a neighborhood of pixels at each spatial location. The phase difference (fringe) map resulting from this IFSAR operation is then unwrapped and used to calculate the height estimate of the imaged terrain. Although the method used to calculate interferometric fringe maps is well known [1], it is generally executed in a post-processing mode well after the image pairs have been collected. In that mode of operation, there is little concern about algorithm speed and the method is normally implemented on a single processor machine. This paper describes how the interferometric map generation is implemented on a distributed-memory parallel processing machine. This particular implementation is designed to operate on a 16 node Power-PC platform and to generate interferometric maps in near real-time. The implementation is able to accommodate large translational offsets, along with a slight amount of rotation which may exist between the interferometric pair of images. If the number of pixels in the IFSAR image is large enough, the implementation accomplishes nearly linear speed-up times with the addition of processors.

INTRODUCTION

Interferometric phase difference maps, also known as fringe maps, are generated from a pair of SAR images that have tightly controlled, highly similar relative collection geometries. The constraints required on the collection of the two images are outlined in [2]. The images can be taken from either a single pass system utilizing a pair of antennas or a two pass system using only a single antenna. SAR is a *coherent* imaging system, and the complex reflectivity functions are processed using the interferometric method so that scene reflectivity common to both images is canceled, yielding the scene topography as transduced by the image domain phase data.

The challenge in generating a fringe map is that first, the images in the pair must be registered to one another within a fraction of a pixel in order to produce the desired result. Due to motion compensation errors, velocity errors and other errors which are often of unknown origin, the images may be translated relative to the other by up to several hundred pixels, in range and/or azimuth. Furthermore, in some two-pass systems incorporating a relative squint, the images may have a rotation or a height dependent misregistration.

Once the images have been registered to subpixel accuracy, the optimal solution in a Maximum Likelihood sense for the phase difference between the two images is given by the equation:

$$\phi_k = \angle \left(\sum_{l=1}^N f_l^* g_l \right) \quad (1)$$

where ϕ_k is the phase difference at pixel k in the image, f_l^* indicates the complex conjugate of pixel l in image 1, and g_l represents pixel l in image 2. The quantity is averaged over a neighborhood of N Pixels to increase the SNR of the phase difference estimate. Typically a 5x5 or 7x7 box is used, making N equal to 25 or 49, respectively.

PROCEDURE

The parallel processing implementation described in this paper is designed to process images that may have large relative translational offsets but very little rotation. First, a two-step registration process is implemented whereby the first, or *coarse* stage, is done serially and determines the translational offset between the images in both range and azimuth, within a few pixels accuracy. Since the rotation between the images is constrained by the imaging geometry to be very small, the image data is divided along the range direction and each processor receives a given number of range lines from each image, in preparation for the second, or *fine* registration step, which is a parallel procedure. This parallelism is by virtue of data partitioning, with each processor receiving a subset of range lines from image 1 and also the corresponding range lines from image 2, via gross

¹ Sandia is a multi-program laboratory operated by Sandia Corporation, a Lockheed-Martin Company, for the United States Department of Energy under contract DE-AC04-94AL85000.

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED



MASTER

DTIC QUALITY INSPECTED 1

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

shift in image 2 data with respect to range, as determined by the coarse registration procedure. The individual processors then work independently to perform fine registration and calculate the phase-difference map corresponding to their assigned portion of the overall image.

Coarse Registration

The coarse registration is done on a single host computer that calculates a Δx and a Δy between the two images as shown in Figure 1. The effect of these offsets is also shown in the example of Figure 3. The variables Δx and Δy , representing a coarse offset in azimuth and range, respectively, are found by low-pass filtering and sub-sampling each image in the pair, followed by a two-dimensional correlation between the images. That is, the *filter and decimate* operation is done over each image separately, and the correlation done globally over the pair. The input images are generally larger than will fit in a single node's local memory. Furthermore, the calculated offsets are used to determine the data partitioning and data distribution for the fine registration stage which follows. For these reasons, and because the coarse registration process is not particularly compute intensive, coarse registration lends itself to being a serial host-node process. The location of the peak of the correlation function indicates the amount of gross shift that is present.

Once an estimate of Δx and Δy are found, the data is distributed to the individual processors by giving each processor a number of range lines from both images. This number is based on the number of processors present, as well as the size of the images in range. The value Δy is used here to range-offset the image 2 data from image 1 before distribution, so the data from each image corresponds in range within a given processor. In addition, more range lines from image 2 are loaded onto each processor than for image 1. The amount of additional data, or overlap, corresponds to the amount of registration error expected in the coarse registration procedure, as well as to allow for a small rotation of the data

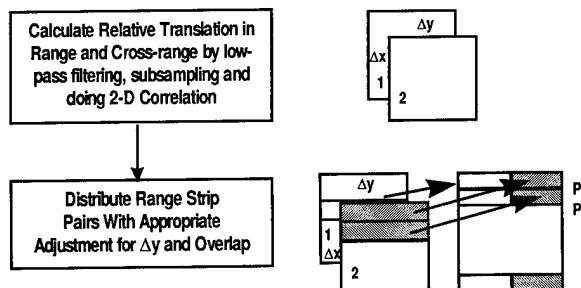


Figure 1. Coarse Registration procedure and data dispersion

after fine registration. This data overlap is slightly wasteful in terms of memory usage, but allows each processor to perform fine registration fully independently, thereby maximizing parallelism and eliminating the need for interprocess communication during fine registration.

Fine Registration

The fine registration step follows the coarse registration step and distribution of range-aligned data, and utilizes all processors in parallel. This step warps image 2 to image 1 within sub-pixel accuracy and calculates the corresponding phase difference map. The steps are shown in Figure 2.

We assume we can model the warping of image 2 relative to image 1 by the affine transform equation:

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \quad (2)$$

where x_1, y_1 and x_2, y_2 are the pixel positions of image 1 and image 2, respectively. The coefficients a and b are estimated from the data using the procedure described below. For each grid position in image 1, the corresponding pixel location in image 2 is determined using (2). A simple bilinear interpolator is used to calculate the pixel value in image 2 and place the result in the warped location for image 2.

The coefficients a and b in (2) are estimated from the image data by calculating local x and y offsets or *tiepoints*

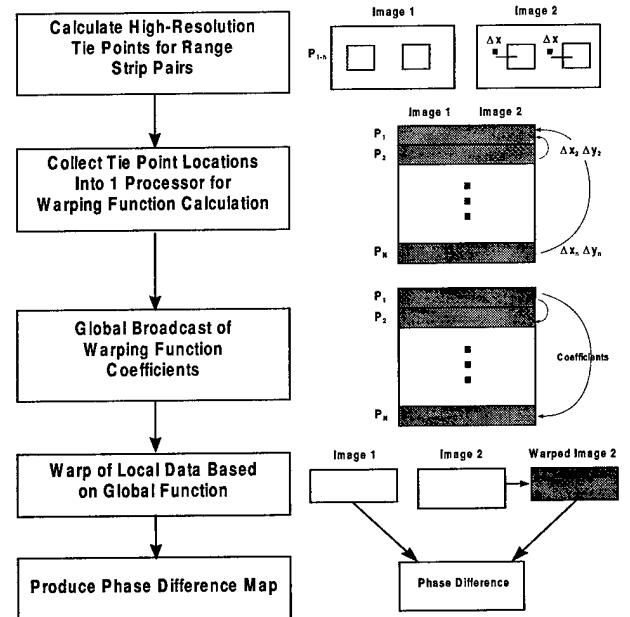


Figure 2. Fine registration and phase-difference map generation on parallel processing architecture

throughout the entire scene, between images 1 and 2. Each processor calculates 3 - 5 tiepoints for the pair of image data within that particular processor's data strip. The local tiepoints are calculated using relatively small correlation boxes of size 64x64 or 128x128 pixels and their peaks calculated to within subpixel accuracy using a quadratic, best fit procedure. Generally, the correlation boxes are equally spaced throughout the strip of image data within a processor. The correlation box in image 2 is offset by Δx , as previously determined by the coarse registration procedure, to accommodate a known azimuth data shift between the images in the pair.

Once each processor has completed calculating the tiepoint offsets for its local data, all processors' tiepoint results are collected in a single designated processor. This processor then eliminates low-quality tiepoints and determines a least-squared estimate of the global warping coefficients in (2). These coefficients are then globally dispersed to each individual processor, where each independently warps its portion of the image data using (2). After each processor has warped its portion of image 2, it is a simple matter of producing the fringe map using (1). The individual fringe maps are then collected and seamed together to form an overall fringe map covering the entire image, as shown in Figure 3.

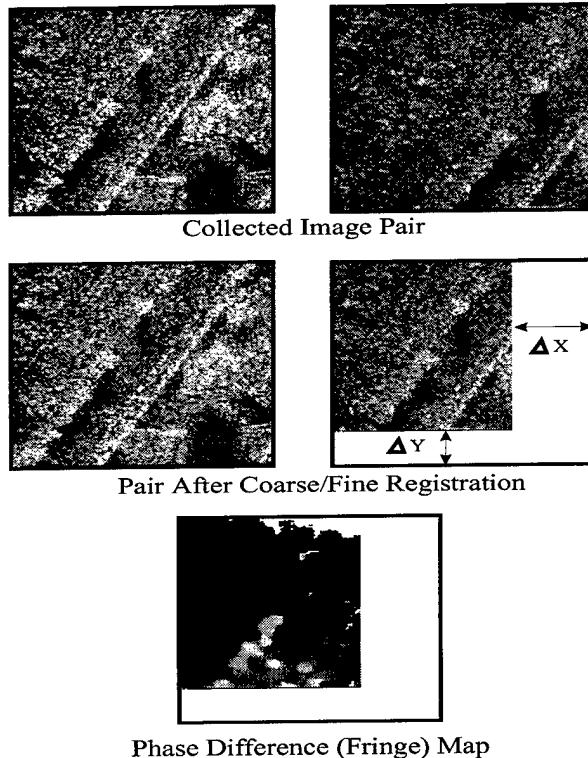


Figure 3. Example of registration and phase difference map generation

TIMING ANALYSIS

The coarse registration and data distribution steps of the algorithm are done serially. These steps are not computationally burdensome and account for only a fraction of the overall processing time, in spite of their serial nature. Computation times for the fine registration and fringe map calculations depend on the input data size, correlation box size and spacing, as well as fringe map averaging filter size. Since this discussion focuses only on speedup via parallelization, these parameters will remain fixed in size and only the variation in number of processors will be considered.

Prior to fine registration, the input data is partitioned with respect to range, and each active processor receives a range-registered strip of data from each of the input images. The minimum amount of range data per node is equal to the correlation box size. Any less, and the correlation calculation can not be performed. Thus, for images with a small dimension in range, not all processors can be utilized. For each processor that can be effectively utilized, parallel speedup for fine registration and fringe map generation is linear. We predict these steps can be performed on a cluster of sixteen 200 MHz Power-PCs in about 4 seconds, which is near real-time. A typical image is about 2000 samples in both range and azimuth.

CONCLUSIONS

The parallel algorithm for interferometric fringe map generation is designed to minimize data communication transfers between processors in a distributed-memory parallel processing machine. This is achieved via a two-step process, whereby the first step, coarse registration, analyzes the gross shift in range registration and distributes the data accordingly. For the parallelized fine registration step that follows, this yields a near linear improvement of processing time vs. the number of processors present, as well as minimizing the need for interprocess communication during computation. However, for maximum parallel computing efficiency, each processor must contain at least as many range lines as the size of the fine registration correlation box, otherwise fewer processors must be utilized and computing efficiency decreases. The use of this parallelized algorithm results in near real-time processing of interferometric fringe maps when implemented on a 16 node Power-PC platform.

REFERENCES

- [1] V. Mrstik, "Terrain Height Measurement Accuracy of Interferometric Synthetic Aperture Radars", *IEEE Trans. On Geoscience and Remote Sensing*, Vol. 34, No. 1, January 1996.
- [2] C. V. Jakowatz, et al., Spotlight-Mode Synthetic Aperture Radar: A Signal Processing Approach, Kluwer Academic Publishers, 1996

M98005017



Report Number (14) SAND--98-0013 C
CONF-980715--

Publ. Date (11) 199807
Sponsor Code (18) DOE/PPNN, XF
JC Category (19) UC-900, DOE/ER

DOE