Tracking of translated and rotated features within satellite images using a tightly coupled MIMD computer

M.R. Saradjian, M. Hayes, L. Hayes

Department of Applied Physics and Electronic & Manufacturing Engineering, University of Dundee, Dundee DD1 4HN, UK

Abstract

A technique for establishing sea surface velocities in sequential satellite infrared images has been developed using a multiprocessor computer system. A consecutive pair of Sea Surface Temperature (SST) images is produced from satellite-derived data and brought into register. A Maximum Cross Correlation (MCC) approach is then used to determine the amount of sea surface feature movement over time. The amount of feature movement, and the time interval between images, determine the surface velocity. As the water patterns observed tend to exhibit rotational movement as well as translation, the MCC method has been extended to detect rotational motions. In general, the correlation coefficient is computed for all possible translations and rotations of the MCC template within the search area. Depending on the size of the template in use, a small amount of template rotation may have little or no effect on the correlation coefficient. It is accordingly possible to use an adaptive approach to minimise correlation searches. By utilizing an Alliant FX/2800 shared memory MIMD computer, and applying appropriate algorithmic and compiler optimisation, the task is implemented sufficiently fast that a complete satellite overpass may be processed before the next co-incident satellite orbit. The system implemented suggests that the operational tracking of rotating sea features is possible in a multiprocessor machine environment.

Introduction

This paper presents early results of work undertaken to develop a parallel implementation of a pattern recognition technique for computing sea surface velocities directly from infrared satellite images. Similar work has been reported by Emery [1], Garcia [2], Kamachi [3], and Tokmakian [4]. The Advanced Very High Resolution Radiometer (AVHRR) instrument on board the polar orbiting TIROS-N to NOAA-12 meteorological satellites provides imagery at approximately six hourly frequency (Kidwell [5]). The thermal infrared channels of the AVHRR may be used for determining sea surface temperature. In this
study calibrated AVHRR thermal infrared channels are used for determining the satellite received radiances. Each satellite received radiance corresponds to a brightness temperature (Lauritson, et al. [6]). The SST is then calculated from the brightness temperature using the U.S. National Oceanic and Atmospheric Administration (NOAA) Multi-Channel (MCSST) algorithm due to McClain et al. [7].

The AVHRR imagery is geometrically distorted by a combination of factors. After a consecutive pair of SST images is produced, one image must be brought into register with the other in order that the two images may be compared for the determination of movement. Given image congruency, it is then possible to compare the two images on a pixel by pixel basis. In this manner, SST values can be determined for the same point over an extended time sequence. The package satell which was implemented by Kwong [8] and further developed by Khan [9] has been used to produce registered satellite image sequences in this study.

Automated Recognition of Translated Sea Surface Features

The movement of a particular feature between one image and a subsequent image is determined by correlating a subimage of the first image with the corresponding subimage in a subsequent image. The MCC approach provides a way to automate the process of determining the amount of feature movement:

$$ R(m,n) = \frac{\sum_{j=1}^{J} \sum_{k=1}^{K} F_1(j,k) F_2(j+m,k+n)}{[\sum_{j=1}^{J} \sum_{k=1}^{K} F_1^2(j,k)]^{1/2} [\sum_{j=1}^{J} \sum_{k=1}^{K} F_2^2(j+m,k+n)]^{1/2}} $$

where $F_1(j,k)$ and $F_2(j,k)$ represent two discrete images, and $(j,k)$ are indices in a $J*K$ pixel template located within an $M*N$ pixel search area (see Figure 1).

![Figure 1](image-url)
In general, the correlation coefficient $R(m,n)$ must be computed for all $(M-J+1)(N-K+1)$ possible translations of the template within the search area. In this study, successive SST images are two-dimensionally lag correlated in 75*75 search areas using 25*25 templates. The point where the correlation coefficient is a maximum is assumed to be the displacement of the translated feature in the second image. The amount of feature movement, and the time interval between the two images then determine the sea surface velocity. The correlation procedures described in Equation (1) are well suited to exploit the architectural advantages of a multiprocessor environment.

**Automated Recognition of Translated and Rotated Sea Surface Features**

One of the limitations of the MCC approach given in Equation (1) is that this approach is not suited to the detection of nonlinear displacements seen in rotating or deforming patterns. The water patterns observed, however, tend to exhibit deformation and rotational movement as well as translation. In order to make the MCC method more robust for the detection of deformed and rotated patterns, the MCC approach has been extended to detect rotational motions. With this extended approach, the correlation coefficient must be computed for all possible translations and rotations of the template within the search area (see figure 2).

The transformation for a counterclockwise rotation of $\theta$ about the centre of an image may be extracted from:

$$\text{rotated}(i'+hy, j'+hx) = \text{original}(i+hy, j+hx)$$

where $(hx,hy)$ are the coordinates of the centre of the original image, $(i,j)$ are the coordinates of the original grid point, and $(i',j')$ are the coordinates of the rotated grid point, which may be calculated from Equation (3) (Harrington [10]).
Depending on the size of the template in use, a small amount of template rotation may have little or no effect on the correlation coefficient. When rotating, the transformed points do not coincide with the original grid points. Since the location of each pixel can only be specified at grid points, the nearest integer resampling function is used to specify the address of each pixel at each transformed grid point. The closer the rotation angle is to zero and/or the smaller the size of template, the less the coordinates of the rotated grid are likely differ from the original grid. Figure 3 shows the template size against the minimum rotation angle required to provide a non-congruent corresponding template. The implication is that small angles of rotation may have little effect on the original grid points. It is accordingly possible to use an adaptive approach to minimise correlation searches.

![Figure 3](image.png)

Figure 3. Minimum rotation angle for non-congruent nearest integer matching of templates as a function of template size.

Additional optimisation approaches, such as similarity measurement, or efficient selection of windows in the search area, may serve to further reduce the number of searches required, in which case some penalty may require to be paid for unsearched areas. The adaptive approach employed allows the reduction of
searches without cost. It should be noted that in the case of a small time interval between the two images, the rotation of patterns would be relatively small, hence in order to find a match the templates require correspondingly smaller rotation.

Parallel implementation

The computation required by the extended MCC method is very time consuming, particularly when implemented on a sequential machine. Parallel computers, however, are well suited to handling computationally extensive operations such as MCC algorithm. By utilizing the Alliant FX/2800 multiprocessor architecture, and applying appropriate algorithmic and compiler optimisation options, the task is implemented sufficiently fast that a complete satellite overpass may be processed before the next co-incident satellite orbit. The Alliant FX series architecture may be categorised as Multiple Instruction Multiple Data (MIMD) (Flynn [11]) systems where each processor operates on independent instructions and data. This is a general purpose architecture intended to offer speed-ups over a wide class of problems. The Intel i860 processor used in Alliant’s FX/2800 provides single processor performance that is nearly as fast as the first Cray supercomputers (Justin [12]). The Alliant FX/2800 machine is well suited to processing required as the four available processors may be programmed to cooperate on the application, affording the advantages of both concurrent processing and vector pipelining.

Vectorisation allows arrays to be processed with vector instructions. The Alliant FX/2800 system used has four processors each containing instruction pipelines for processing groups of up to 32 elements. Concurrency is a high-level or global form of parallelism denoting independent operation of simultaneous computing activities (Torborg [13]) and accordingly provides much of the performance observed. Theoretically, a multiprocessor with \( n \) processors should be \( n \) times faster than a uniprocessor, but the communication overhead decreases this factor.

The vectorisation and concurrency mix applied to the MCC problem is the subject of further investigation.

The Alliant FORTRAN compiler attempts to generate code that exploits both the vector and concurrent capability of the machine. The compiler optimizes multiple levels of nested DO loops for parallel/vector operation. Concurrent code automatically utilises as many Computational Elements (CEs) as are available. The compiler also supports vector notation extensions. Special explicit compiler directives are provided for the programmer to control compilation modes for loops where necessary. The MCC application has been reprogrammed in an autoparallelisation-explicit way by restructuring program segments which inhibited optimisation. Since the MCC application makes extensive use of loop constructs, many opportunities for code rearrangement were possible, although this loop restructuring involved substantial programming effort. When compiled using full compiler optimisation, the hand-tailored code produced considerable reduction in
execution time. Table 1 shows the cpu time for both the original and the restructured code.

Table 1:

<table>
<thead>
<tr>
<th>Code execution</th>
<th>cpu time (hh:mm:ss)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCC algorithm before restructuring</td>
<td>4 : 54 : 42</td>
</tr>
<tr>
<td>MCC algorithm after restructuring</td>
<td>3 : 32 : 58</td>
</tr>
<tr>
<td>Time difference</td>
<td>1 : 21 : 44</td>
</tr>
<tr>
<td>Percentage speedup</td>
<td>27.7%</td>
</tr>
</tbody>
</table>

The MCC program was also executed on a reference uniprocessor Sun SPARCstation IPX, but with substantially increased execution time when compared to the execution time on the Alliant FX/2800. Table 2 shows the cpu time for both the Sun SPARCstation IPX and the Alliant FX/2800.

Table 2:

<table>
<thead>
<tr>
<th>Machine environment</th>
<th>cpu time (hh:mm:ss)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alliant FX/2800</td>
<td>3 : 32 : 58</td>
</tr>
<tr>
<td>Sun IPX</td>
<td>17 : 47 : 15</td>
</tr>
</tbody>
</table>

Results

Sequential NOAA satellite AVHRR images have been used to compute sea surface velocities (figures 4-6). A Multi-Channel (MCSST) correction technique has been used in order to minimise the atmospheric effects on the AVHRR data. Successive SST images were brought into geometric register, and movement estimated by template correlation. The MCC approach was found not to be robust for the detection of nonlinear displacements such as rotating or deforming patterns. Accordingly, the MCC approach has been extended to detect rotational motions. Since a small angle of template rotation may have little or no effect on the correlation coefficient depending on both the size of template and the amount of rotation, an adaptive approach for minimising the correlation searches has been used to reduce the number of iterations and the processing time. Issues of feature deformation have not been addressed directly in this study, but for long image separations (e.g. one day) the MCC method may not give correct velocity vectors due to extreme template deformation. Template deformation over time is the subject of further investigation at the University of Dundee.
Figure 4. First image (1/8/91, 13:31 GMT, NOAA-11)

Figure 5. Second image (2/8/91, 13:19 GMT, NOAA-11)

Figure 6. Velocity vectors determined from movement observed between the first image and the second image 24 hours apart. Equivalent to 25 km, a velocity of 0.29 ms\(^{-1}\).
218 Applications of Supercomputers in Engineering

Conclusion

Different programming approaches adopted for the implementation of the MCC algorithm have shown that program execution on a multiprocessor computer with an autoparallelising compiler may be substantially improved when the code segments are written in a manner appropriate for the compiler. A comparison between the Alliant FX/2800 and a reference Sun IPX reveals a five fold performance increase obtained in the multiprocessor environment. While the price differential between the machines is much greater than a factor of five, use of the Alliant FX/2800 does enable one satellite scene to be processed before the next orbit of the same satellite - a critical requirement for the implementation of an operational system.

The extended MCC method presented provides a way of automating the process of determining the amount of feature movement over time. The amount of feature movement, and the time interval between the two images, is then used to determine the surface velocity. The main shortcoming of the system is the requirement that the area of interest be reasonably free of cloud. However, it is expected that this technique will have application beyond sea surface feature tracking, where the presence of cloud may not be a problem. The system implemented suggests that the tracking of rotating sea features is possible on an automated basis.

Acknowledgement

Thanks are due to the University of Surrey who kindly supported our use of their Alliant FX/2800 system for the preparation of some of the results presented.

References


4. Tokmakian, R.; Sea surface velocity determination using satellite imagery:
Applications of Supercomputers in Engineering


