TRANSPUTER BASED SIMULATION OF SAR IMAGERY

G. Sudhir R. Bahl

Centre for Applied Research in Electronics, Indian Institute of Technology, New Delhi, 110016, INDIA.

K.K. Biswas

Dept. of Computer Science & Engg., Indian Institute of Technology, New Delhi, 110016, INDIA.

email: gsudhir@netearth.ernet.in

ABSTRACT

Synthetic Aperture Radar image simulation approach based on Point Scattering Model, calculation of the echoes based on actual reflected field, and signal processing of the echoes is fundamental, rigorous and general and does not rely on any closed system representation and prior assumption of the detection method in the receiver. However, due to large number of scatterers, the data collection part of the simulation is very compute intensive. We show that when many independent processors are working in an interleaved manner, not only can the data collection time be drastically reduced, but also the signal processing on the collected data can be completely done in parallel leading to further reduction in overall simulation time.

KEY WORDS : Synthetic Aperture Radar, Strip-mapping, Transputer.

1. INTRODUCTION

The latest technology being used in remote sensing is the Synthetic Aperture Radar (SAR) to generate very high resolution images of the terrain. Simulation of SAR imagery helps in better understanding and interpretation of the images. The four basic components involved in SAR image simulation are: (1) the imaging model, (2) the radar geometrical propagation phenomena, (3) the ground model, and (4) the reflectivity model. The simulation requires the operating parameters of the radar, a symbolic representation of the relief and dielectrical properties of each selected ground site, and the reflectivity properties of various categories in the ground truth data base [2].

For the simulation to be realistic, it is necessary that the simulated image have, along with objects of interest, the natural speckle noise ("grainy appearance"). Many radar image simulation programs have incorporated oversimplifications in either the model or in the representation and calculation of microwave reflectivity of various categories. The "Point Scattering Model" that we use, being fundamental, rigorous and general, does not rely on arbitrary oversimplifications.

Unfortunately, this type of simulation turns out to be quite compute intensive. This is due to the large number of scatterers present in the radar beam each time when an echo is being recorded and the need for a large number of such echoes to generate the radar image. The most time-consuming part in the simulation is the data collection process. We show that the task can be subdivided very efficiently amongst a number of independent processes by exploiting the "Strip-Mapping" nature of SAR. We explain this in the following section. We also show that it is possible to do signal processing in parallel with the data collection process.

2. STRIP-MAPPING NATURE OF SAR & SIGNAL PROCESSING

<u>A) Strip-Mapping nature of SAR (Fig. 1)</u>

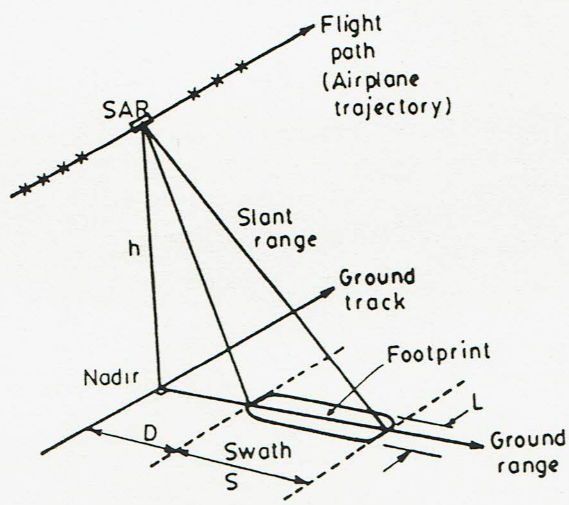


Fig 1. Strip-mapping nature of SAR

The radar is mounted on an airplane/shuttle/satellite that moves along the trajectory, at an altitude 'h', shown as 'airplane trajectory'. At each point on the trajectory marked 'x', it transmits a pulse and records the reflected echo from the terrain. The radar beam has a width S and length L on the terrain and is called radar "Foot-Print". It is apparent that as the radar moves along the trajectory, the radar "Foot-Print" covers a strip of terrain of width S at a distance D from the projected radar trajectory on the terrain. Hence, the radar, at each transmit/receive (T/R) point on the trajectory, would illuminate a length L of the Strip of width S and record the reflected echoes. These echoes are processed (range compressed and then azimuth compressed) to generate the SAR imagery of the Strip. All these parameters are the inputs to simulation from the SAR system design module [3].

In our simulation, the terrain to be imaged is modelled as a matrix of bins. Each bin consists a number of scatterers and information about their locations and the dielectric and reflectivity properties.

At each T/R position on the trajectory, the Foot-Print covers some bins of the matrix and only the scatterers inside these bins will contribute to the echo recorded.(The recorded echo is the vector sum of the echoes from the individual scatterers.) It can be seen that at each T/R point on the radar trajectory, echoes have to be computed from a large number of scatterers and a lot of such T/R events happen to generate the radar image. So a sequential data collection process would be very compute intensive and hence the need for several independent processors to do the data collection. We describe in the next section how several independent processors could be employed to reduce the simulation time by a factor approximately equal to the number of independent data collection processors.

B) Signal Processing

The signal processing to be done on the collected data consists of 2 stages :

i) Range Compression

This is done on an echo independent of the other echoes. The result of this operation is basically to split each echo into ,say, m sub-echoes corresponding to m range cells depending upon the swath width of the terrain strip and the required resolution of the image. So after this step, the basic N echoes (N echoes are needed to generate the image) are transformed into a matrix of sub-echoes of m x N dimension as shown in Fig. 2.

ii) Azimuth Compression

This follows the range compression stage. Azimuth Compression is performed on each sub-echo of the m x N matrix (the output of the range compression module) and it depends upon, say, P previous sub-echoes in the same range line and P succeeding echoes in the same range line. This is a sort of moving-window processing centred on a subecho for each of the range lines.

A complete signal processing analysis is given in [1] and [3].

3. PARALLELISM

Suppose the site of interest is of length L and width S and say N echoes have to be collected to simulate the SAR image of the site. If M processors are used to collect the echoes simultaneously from the model of the site, say, the group of (1 to N/M) echoes are collected by the processor 1, (N/M + 1 to 2*N/M) by processor 2, ... , ((m-1)*N/M + 1 to N) by processor M, the whole data collection process time is clearly reduced by a factor M. We show that if the individual processors work in an interleaved manner, one can do the signal processing in parallel in a separate processor without increasing the data collection time.

We illustrate this with just 3 independent processors for data collection. Fig. 3 shows which processor collects which echo. When the processors collect the echoes in this manner, they can immediately transfer their collected echo to the signal processor which can start signal processing. The interleaved nature of the processors helps because for azimuth compression, the consecutiveness of the collected echoes is very important.

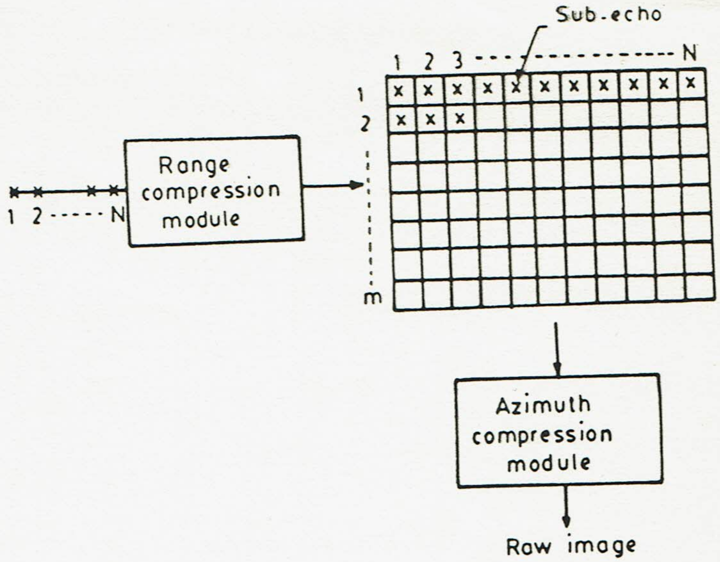


Fig. 2 : Signal processing stages

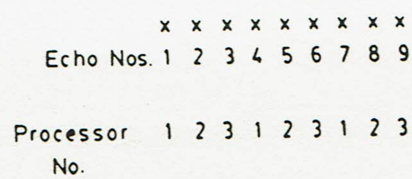


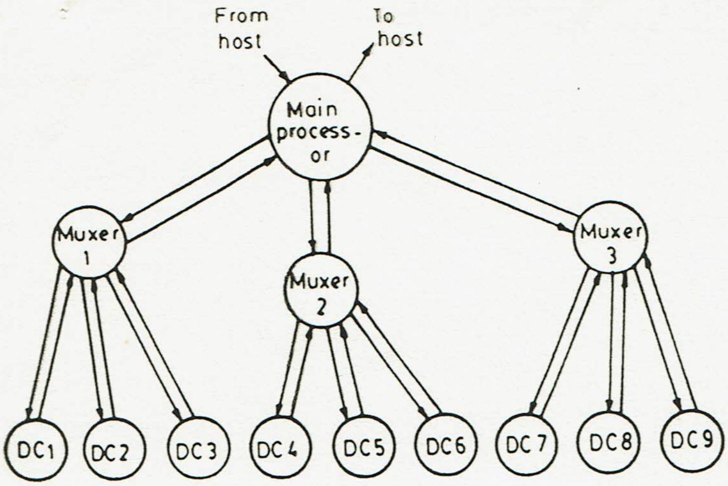
Fig. 3 : Interleaved data collection

4. IMPLEMENTATION

The simulation code was written in OCCAM and tested on a 32 transputer system networked to a SUN workstation. Due to restrictions on processor grabbing, the simulation was tested on a 10 transputer network with one main and 9 slave processors. The main processor did the file i/o and distribution of data to the slave processors. The 9 slave processors were dedicated to the data collection process and the main processor, after data distribution to the slaves, switched mode to receive data from slaves and do signal processing and file i/o in parallel. The modules running in parallel are as shown in Fig. 4.

The mux processors are the multiplexing processors running in parallel on one of the 3 data collection processors they are connected to, due to the limitation of the number of direct physical links to a transputer. Hence, 9 different processors were used to simulate 9 data collection processes and 3 multiplexing processes and 1 dedicated processor was used for signal processing, file i/o etc. (the main processor).

In our simulation work we integrated the range compression module with the data collection module as in a simulation, the scatterers can be easily sorted into their ranges to speed up the overall simulation time (because, range compression is done by 9 processors) and distribute the required memory capacity among the slaves. Hence, the data collectors would now send a chunk of range compressed sub-echoes instead of a single echo to the main processor. The main processor would receive the chunk and as soon as 2*P+1 consecutive chunks arrived, it would azimuth-compress the sub-echoes in the central



MUXER : Multiplexer
DCn : Data Collector n
tohost : Channel to Host Processor
fromhost : Channel from Host Processor

Fig 4 : Processes running in parallel

chunk and switch to receive another chunk of range compressed sub-echoes from one of the slaves. Care was taken not to penalize the communication between the data collector and the main processor by giving higher priority to the channels carrying the chunks of range compressed sub-echoes to the main processor than to do the signal processing on the received echoes. This is important as data collection is time consuming and hence data collectors should not be made to wait when they want to transfer to the main processor their chunk of range compressed echoes(OCCAM channels use synchronised communication). Also the parallel signal processing (azimuth compression) is done only on one sub-echo in the chunk of sub-echoes at a time so that main processor is readily available to receive another chunk from one of the slaves if there is one, else, it would az-compress another sub-echo in order and be ready to receive a chunk of sub-echoes from one of the slaves, and so on. This also gives priority to communication between main processor and the data collector. The results are discussed in the section on results and conclusion.

5. APPROXIMATE ANALYSIS OF SIMULATION

Before the actual code was written and implemented an approximate analysis based on our sequential simulation work was done as follows :

Let us assume that range compression is integrated with data collection process as it is simple in a simulation effort. Let the data collector module take time T1 to give one range-compressed sub-echo. Let there be 'm' ranges so that total computation time for m range-compressed echoes is m*T1. So, if N such echoes are required to generate the SAR image, a sequential data collection program would take N*m*T1 units of time. Let T2 be the azimuth compression time on a single sub-echo. Assuming that az-compression can be done on all N*m sub-echoes, the az-compression time is N*m*T2. So, if the azimuth compression follows the sequential data collection, the total time taken to simulate the raw image is N*m*(T1+T2).

Now consider a parallel data collection program with M slaves working in parallel to collect data. It is easily seen that the time for data collection is reduced by a factor M i.e., the time is N*m*T1/M. Now consider the case of doing signal processing (azimuth compression) in parallel on the data being received. As the total computation for m range-compressed sub-echoes (one chunk of sub-echoes) is m*T1, a slave is generating chunks at frequency 1/(m*T1) chunks / second. The other slaves are running in parallel and asynchronously but their chunk generating frequency is also 1/(m*T1). So basically, the main processor has to attend to each slave processor at 1/(m*T1) frequency. Let t1 be the channel communication time in the transfer of one chunk of sub-echoes between slave and main processor. So the total time lost in channel communication of chunks from all the M slaves is M*t1. Time available for signal processing (azimuth compression) is m*T1 - M*t1. Let T2 be the time for azimuth compression of a single sub-echo. Then the number of sub-echoes that could be az-compressed is (m*T1-M*t1)/T2. If t1 is very small compared to T1,we can neglect this time and the number of sub-echoes that could be az-compressed is (m*T1)/T2. It is usually expected in the simulation that T1 >> T2. So (m*T1)/T2 >> m implying that it is possible to az-compress completely all the echoes in the lull time of the main processor when all the slave processors are busy computing the chunks of range compressed sub-echoes.

An actual network was built up and tested with delay loops to simulate T1 & T2 (relative) and the number of sub-echoes az-compressed in parallel was recorded. The results were :

Ratio TC/TP	Percentage of echoes processed in parallel
1	6.23
2	12.43
4	24.70
8	63.19
10	78.30
15	100.0

TC : Time to compute a sub-echo

TP : Time to process a sub-echo

[Results for M=9]

6. RESULTS AND CONCLUSION

The simulation was done on 10 independent processors only due to restriction on the availability of processors though any number of processors could be used by constructing a suitable network. One main processor was used for initial i/o and to distribute the data to other 9 slave processors. Then the main processor would switch to data receive and azimuth compression mode whereas the 9 slave processors, after receiving the data from the main processor, start calculating the echo, do range compression and send it back to the main processor.

The simulation code was tested on different terrain models of the same size - 10 km in the along- trajectory direction and 1km in cross-trajectory direction. For the purpose of verification, the models were constructed to have some defined objects at defined locations so that we could find them in the final image. The simulated images are at the end of the paper.

All the images perfectly agree with their models and the simulation time was drastically reduced by a factor approximately equal to the number of slaves used in the simulation. When the simulation was done on sequential data collection program and then followed by azimuth compression, the time taken for the terrain models we used was around 65 hours while using the 10 (9+1) transputer network, the time was around 7 hours which is according to our expectations.

We also observed that in our simulation work, linear az-compression(correlation) was only required as "Range Curvature" was very small as we had considered the radar to be moving on a low altitude platform and not on a satellite. If one considers the SAR on a satellite, the data collection and range compression would require only a little modification but azimuth compression would then be in 2 dimensions along with interpolation. In that case, azimuth compression should take more time and as we observed in our simulation, there is more than enough time for the main processor to incorporate some more processing complexity in parallel to give the raw image as soon as data collection process is over. As the number of slave processors for data collection increases, the time available for azimuth compression (m*T1 - M*t1) would decrease as M would increase. But we strongly feel that there is usually enough time for the main processor to carry on with azimuth compression in parallel and completely finish it by the end of data collection. Even if in some cases it is not possible to completely finish the azimuth compression of all the sub-echoes, the main processor can az-compress as many sub-echoes as possible and the remaining can be processed as a sequential process as soon as the data collection process is over.

We conclude that for fundamental, rigorous and general simulation of SAR imagery using the point scattering model and calculation of echo based on the actual reflected field, and doing signal processing of the echoes, one can drastically reduce the time by using a number of independent processors working in interleaved manner in a parallel architecture as we have shown. We also feel that modelling the terrain as we have done as a matrix of bins would help a long way in reducing the overall simulation time.

7. ACKNOWLEDGEMENT

We gratefully acknowledge the active help provided by Paul Francis, Computer Centre, I.I.T. Delhi during the simulation work.

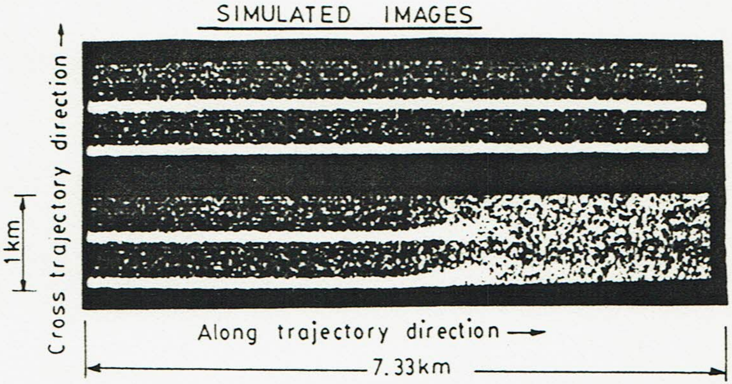
REFERENCES

[1] D.C. Munson and R.L. Visentin,"A Signal Processing View of Strip-Mapping Synthetic Aperture Radar,"IEEE Trans. Acous.,Speech,Signal Processing,Vol 37,No.12,pp. 2131-2147, December 1989.

[2] J.C. Holtzman,V.S. Frost,J.L. Abbot and V.H. Kaupp, "Radar Image Simulation,"IEEE Trans. Geosci. Electron.,vol. GE-16,No.4,pp. 296-303,October 1978.

[3] C. Elachi, Spaceborne Radar Remote Sensing: Applications and Techniques,IEEE Press, New York, 1988.

SIMULATED IMAGES



Above: Image of a terrain containing two hill ranges running parallel to radar trajectory

Below: Image of a terrain containg two hill ranges sloping to ground level at centre

Image of a terrain containing a circular object at centre

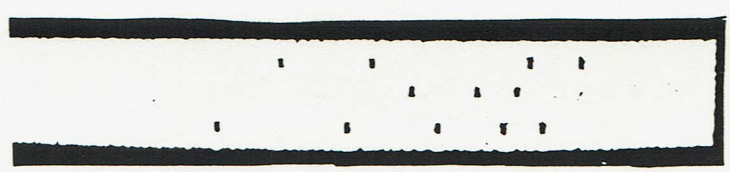


Image of a terrain containing many small rectangular objects showing range overlay and shadow