NASoftware Limited
1 Prospect Road, Birkenhead, CH42 8LE. UK
Tel: +44 151 609 1911 Fax: +44 151 550 7830

Real Time MTI from SAR Imagery

InfoSAR Ltd, N.A Software Ltd, Raytheon NCS,
Intel Corporation, Embedded Computing Division

May 14th 2008

SARMTI Image Analysis

SAR provides all-weather high resolution images of ground scenes; typically these are presented to an operator for interpretation. However, the data collected to form these images contains substantial information about the scene which can be extracted automatically to provide operator assistance or even to trigger a response to the scene, InfoSAR Ltd is well known for its advanced feature extraction techniques. These are typically computationally expensive; the ability to run them in real time would make them operationally attractive.

One novel technique developed by InfoSAR is SARMTI: the ability to produce moving target information directly from SAR data. This technique has major advantages compared with the currently used MTI methodology:

- No need for a separate MTI mode or additional antennae.
- Estimates across-track velocity and acceleration and along-track velocity, for both slow and fast movers. Motion estimates are about 10 times better than MTI
- Automatic registration within SAR image, to within one pixel. Azimuthal registration is about 100-1000 times better than MTI
- Distributed targets are imaged at full resolution so that target recognition techniques can be applied.

Figure 1 gives an example of the power of the technique; and the accuracy of the retrieval is shown in Table 1.

Figure 1 A strip of SAR imagery on which has been imposed a number of moving targets (position shown in green). The detected position of the moving targets is denoted in red

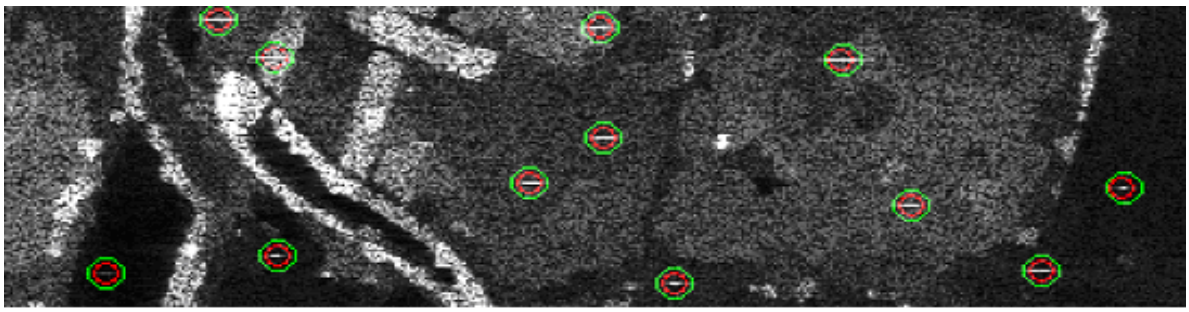

Table 1: Comparison of the imposed along-track velocity with the measured value for each target in Figure 1.

Target Position	Imposed Velocity	Measured Velocity
350,105	10	9.6
34,106	8	8.4
91,20	6	6.0
92,99	4	4.8
306,79	2	2.4
201,8	1	1.2
378,72	-1	-1.2
226,110	-2	-2.4
177,70	-4	-3.6
202,52	-6	-6.0
72,5	-8	-8.4
283,21	-10	-10.8

Real Time SARMTI?

The SARMTI technique is necessarily compute intensive. InfoSAR and N.A. Software, with Nallatech Ltd, Raytheon NCS, and Intel Corporation, is currently studying it to see how difficult a real time implementation would be. In the first stage of the project, the hardware targeted is a single Intel Caneland 4-socket system populated by 1.86GHz 45nm, Quad-Core Intel "Tigertown" processors; the aim is to see how fast the chosen application (represented by four model scenarios) will run. InfoSAR, N.A Software, Nallatech Ltd, and XtremeData are providing consultancy for this work.

Why This Hardware?

A conventional approach to providing fast implementations of a DSP application would be to utilise COTS DSP boards populated with currently popular DSP-oriented processors. However, the Intel solution is seen by Raytheon NCS as having several potential advantages:

- A single Caneland system populated with quad-core processors provides 16 cores in an SMP environment, making it relatively straightforward to utilise all cores on a single executable by multithreading the code.
- Current Intel processors will run at clock rates very significantly higher than more specialised DSP processors.
- Intel enhancements to their instruction set, and their unrivalled experience in producing low power consumption processors, make them increasingly attractive for today's advanced airborne DSP applications.
- Provision is made on the Caneland board for including standard DSP accelerators; the Nallatech FPGA modules are socket compatible with the Four socket Caneland system. The XtremeData FPGA modules are socket compatible with Intel Xeon® processor-based dual-socket server boards. Thus, the potential advantages of a tailored FPGA solution to handle standard parts of the code, such as FFTs, is not lost by the primary use of COTS general-purpose Intel processors.
- Intel provides a well-developed C compiler (icc) and a rich set of porting and optimisation tools with which to tune a multithreaded port of the application.

Results

The original application represented demonstrator code for the technique being studied; it generated simulated SAR data prior to processing this data and provided four runtime parameter sets generating four typical scenes.

Excluding the data generation, the code contained 20,000 lines of C and was unthreaded. N.A. Software (NASL) who were already familiar with the application, profiled the code and produced an initial threaded version; minor alterations to the code were made for better efficiency. The unthreaded and initial threaded versions utilized FFTW library functions. The threaded code was then optimized by calling the FFTW functions with Intel Math Kernel Library (MKL) FFTW wrappers and using vector operations from the MKL.

The following table shows the performance improvement that has been achieved on the Caneland system. The table below shows that the code is between 13.3 and 17 times quicker after the above optimisations have been carried out.

Table 2: Phase 1 timings

	Demo 1	**Demo 2**	**Demo 3**	**Demo 4**
Original time	85 secs	120 secs	104 secs	166 secs
Optimised time	6.4 secs	8.5 secs	6.1 secs	11.8 secs
Speed up (percentage)	86.7%	92.9%	94.1%	92.9%
Speed up (times quicker)	13.3 times	14.1 times	17.0 times	14.1 times

System Configuration Information: Intel® SFC4UR: "Foxcove" Rack Mount Server with 4 Quad Core Intel Xeon™ Processors L7345 ("Tigerton"), (1.86 GHz, 8MB L2 Cache, 1066 MHz FSB). The system includes 4 memory boards with 16MB of 667MHz FBDIMMs and a 73 GB SAS hard disk drive. CentOS* 5.1, 64-bit release ("x86_64) . gcc version 4.1.2 ; FFTW library; Intel icc 10.1, Intel Math Kernel Library (MKL) 10.0. Compile flags: gcc –O3 –xT, icc –O3 –xT; icc –O3 –xT –ip –fno-alias –fargument-noalias.

Next Steps

We are all pretty pleased with these results. The planned work from here on includes:

Phase 1:
Replace the processors to provide a total of 24 faster cores.

Phase 2:

We plan to study how best to utilise both Nallatech® and XtremeData® FPGA technology to further improve the overall times. This work has started; it is likely to require some code restructuring to maximise the throughput, and current work is concentrating on estimating the improvements to be expected. If the estimates look encouraging we hope to implement the restructuring needed.

Phase 3:

The application being considered would be most useful if it were done on the fly. At the end of Phase 2 we hope to be in a position to estimate the viability of this.