Performance Tradeoff Considerations in a Graphics Processing Unit (GPU) Implementation of a Low Detectable Aircraft Sensor System

Christopher SCANNELL, Kevin COX, William SMITH, Carlos MARAVIGLIA

Naval Research Laboratory, Washington, DC

Abstract. The United States Naval Research Laboratory (NRL) is developing a Large Area Scanning and Surveillance Optical System (LASSOS) for identifying and tracking low detectable manned and unmanned aircraft. The system employs altitude-azimuth swept Optical sensors to scan the surrounding airspace and give timely warning of pre-attack targeting operations. Due to their size and standoff distances, the smallest of these aircraft present very small sensor footprints, requiring high-resolution, high-data scans which must be processed in real time. Given packaging size and weight constraints and given the image feature-extraction nature of the sensor data processing problem, NRL is investigating the GPU technology for the high-computational-load front end of the processing chain.

Topic. Computer Systems

Sub-topic. Computer Systems

Keywords. INFORMATION SYSTEMS: Computer Systems, AEROSPACE SCIENCES: Modeling & Simulation, SPACE & MISSILES: Missile Systems.

Introduction

Low detectable aircraft present a challenge to national security and our nation’s military forces. Unmanned aircraft typically pose a particularly difficult challenge to detection due to their small size. Such threats can present a very small radar cross section and be difficult to detect optically due to their small spatial extent. LASSOS is a system to selectively scan large sectors of the sky to detect these threats using very large optics and image processing techniques in a cost

Report Documentation Page	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE **JAN 2013**	2. REPORT TYPE	3. DATES COVERED **00-00-2013 to 00-00-2013**

4. TITLE AND SUBTITLE **Performance Tradeoff Considerations in a Graphics Processing Unit (GPU) Implementation of a Low Detectable Aircraft Sensor System**	5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Research Laboratory,4555 Overlook Ave., SW,Washington,DC,20375**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S) 11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, Grapevine, TX, 7-10 Jan 2013.

14. ABSTRACT

The United States Naval Research Laboratory (NRL) is developing a Large Area Scanning and Surveillance Optical System (LASSOS) for identifying and tracking low detectable manned and unmanned aircraft. The system employs altitude-?‐azimuth swept Optical sensors to scan the surrounding airspace and give timely warning of pre-?&#8208 attack targeting operations. Due to their size and standoff distances the smallest of these aircraft present very small sensor footprints requiring high-?‐resolution high-?‐data scans which must be processed in real time. Given packaging size and weight constraints and given the image feature-?‐extraction nature of the sensor data processing problem NRL is investigating the GPU technology for the high-?‐computational-?&#8208 load front end of the processing chain.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**7**	

effective design. LASSOS uses a variety of sensors that cover several spectral bands (visible, near IR, Shortwave IR, and potentially mid and long wave infra red) and generates a very high data-rate video output. Techniques for using Graphics Processing Units (GPUs) for processing this high-rate video will be discussed that allow the real-time identification of targets.

System Description

LASSOS is an optical sensor system intended for use in maritime and land-based operations. It is designed to scan a very large sector of the surrounding airspace for small airborne craft with difficult or uncooperative detection characteristics. Such target craft are defined as uncooperative due to such characteristics as low metallic signature, small size, evasive flight profiles or other covert characteristics. And, they may be manned as well as unmanned. LASSOS can be deployed in single or multiple unit configurations depending on the number and spatial extent of the target craft and can be deployed on stationary or moving platforms such a ships.

In operation, LASSOS employs adaptable search patterns in order to surveil a wide extent of airspace at the resolution and scan rate necessary for automated detection of targets at required engagement ranges. In order to provide the necessary high resolution, a long focal-length (2000 to 4500 mm) optical system, rate stabilized about azimuth, elevation and roll axes is used. Stabilization is achieved with a combination of inertial reference unit oriented to the host platform (e.g., ship, vehicle) and the use of gyroscopes in the positioning mirror stage.

LASSOS uses multiple sensor types covering several spectral bands, including visible, near IR, shortwave IR and potentially mid wave and long wave infra red. The sensors are line scanned but could use CCD or focal planes due to the design of the optical path behind the telescopic optical element. The line scanners produce a digital video stream that is sent to an image processing system for automated detection of targets. The video stream is not designed for display to a human operator human for detection purposes because of its varying and non standard size and its very large pixel count. The multiple spectral bands of line scan video streams are fused together to improve target signature detection and extraction. Extracted detections are then defined as regions of interest for further inspection. That is, the final regions of interest are presented to a user for threat confirmation. In support of the user, LASSOS has a remote control software station that allows assessment of combination of detections as well as the ability to review and inspect image subsamples for detects and identifications of interest. Also, the control software can provide tracks of regions of interest combining more than one LASSOS input and yielding one continuous track over time.

Even within the same spectral band, LASSOS uses multiple line scanners or focal planes to support automated image processing algorithms such as clutter reduction and temporal change detection. The image processing algorithms also use segmentation to define different processing regions of the video streams for different algorithmic inspection. Sky versus ground presents a different set of problems and approaches. The image processing, depending on its complexity is either real time or near real time. The video gathered is either not stored or stored depending on mission needs.

The optical/IR system is designed and packaged to provide maximal flexibility and sharing of among the various sensor types. This is accomplished via a mirror-based splitter system after the telescope which allows sharing of the focal image plane of the telescope among various sensors. This mirror design allows several sensors and their placement in the sensor stage so that they effectively share the same stabilized, scanning optical path, also allowing for multiple spectral bands to be used in this sensor stage. Finally, the mirrors are oriented allowing the combination of line scanners with focal planes and CCDs.

LASSOS's greatest effect will be in situations with multiple units in operation. Then, final regions of interest will be brought together in presentation to a user for fusion and final threat determination. In stationary deployment situations, units can be spread out in a diagonally oriented pattern with some depth allowing for tracking over wide range and users to monitor detected targets. In a perimeter defense strategy, detection of circling targets, for example, would require deployment of systems on the defended perimeter. In mobile deployment, ships in transit for example, these systems would be used on deck in multiple ships in the transiting unit.

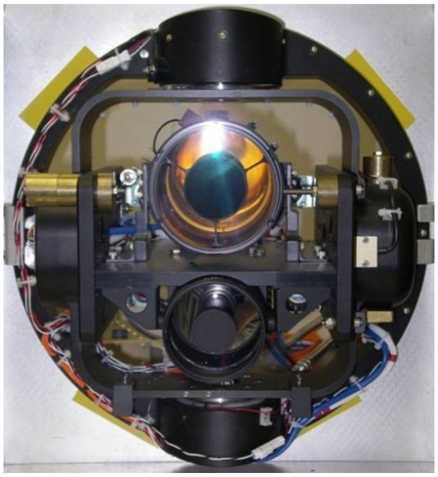

Figure 1. Gimbal-mounted IR and optical sensor package

Figure 2. Efficient use of scanning linear array to focus on the near-horizon area of interest.

Detection Algorithm

The method for finding potential targets within an image is based on a simple region growing algorithm. Raw images from the imager thread are pre-processed using a median filter and horizontal line averages to produce a normalized image. Normalizing the raw image data from horizontal line averages proved successful given the relatively uncluttered nature of the maritime environment in the data set applied to this project. From the normalized image, seed points of highest contrast, both positive and negative, are used as the starting point for region growing. Finding the seed points is accomplished by comparing each pixel in the normalized image against a line dependent dynamic threshold. This threshold is the line average plus a user defined sigma offset. Pixels above the threshold are aggregated into seed points using a nearest neighbor approach. The number of seed points is determined by the sigma offset. A sigma value between three and four generally produced fewer than 5 seed points for our application.

Once the seed points have been identified they are individually expanded into potential target areas. Target expansion is an iterative process where the current target area (initially the seed point) is grown by assigning pixels outside this area as part of the background or part of the target. The area outside the current target is considered the background area and is sized as a rectangle slightly larger (user defined) than the target. Pixels in the background are considered part of the target if they exceed a threshold generated by weighting the difference between the background floor and the target peak pixel values. For this application the background floor and target peak pixel values are the 25th and 95th percentile pixels in the background and target areas. Weight values between .55 and .75 seemed most effective at distinguishing target from background pixels. Once all background pixels have been assigned, the new target area becomes the rectangle encompassing all of the target pixels. If there are no new target pixels region growing stops. Once target expansion is completed, potential targets are then collected and sent to the master tracker for sensor fusion.

There is a broad set of characteristics, parameters and tradeoffs that influence and interact with the design and performance of the detection algorithm. We have been looking at many of these factors as part of the current and future work involved with this paper. These factors include:

Application Factors

* Sensor resolution and data rate – e.g. range of 100-to-1, Optical and IR

* Airspace background environment – e.g. clear, haze, fog, rain, cloud formations

* UAS object – e.g. size, speed, orientation, color, reflectivity

* Feature extraction algorithm – e.g. fast but low-complexity blob detection, versus slower but sophisticated object recognition.

* Problem data space segmentation – e.g. small size (altitude-azimuth) spatial processing segments (tiles) which better map to GPU shared memory and isolate background clutter statistics, but may lose target-to-background detection differential, versus larger processing tiles which better preserve target-to-background detection differential but have less uniform background statistics and do not map as efficiently to GPU shared memory.

* Expected UAS spatial density – e.g. very low (<< 1 per tile), allowing prescreening processing optimizations, or greater, requiring full detection processing over all tile spaces.

GPU Hardware and Software Factors

* OpenCL versus CUDA

* Optimal employment of GPU memory classes - global, shared, texture

* GPU core utilization - Structure tile algorithmic processing to allow redirection of idled data threads in a block to an active data region.

* Process modularity and data flow - Combine and sequence tile row/column operations to minimize inter memory transfer and maximize residency of active data in available high-speed shared or texture memory.

GPU Optimizations

The video input data in the visible spectrum is 8-bit data collected in 12K samples at a rate of 60khz which are divided into 256x256 cell tiles for separate processing. A typical target at detection range is roughly 10x10 pixels and we have not yet dealt with the extra processing required when the target is not wholey contained within a single tile. Given the large number of cells needing to be processed (12*1024*60000/256/256=11,250 tiles/sec) and the fact that the processing of each tile is independent and reasonably compute intensive, a GPU-enabled implementation seemed like a good match. One challenging aspect of using the GPU, however, is the high communication cost of sending so much video input data over the PCI-e bus to the graphics card, the primary bottleneck in any GPU application. We took two significant steps to mitigate the effects of this data bottleneck and achieved very impressive speedups not only over serial implementations of our algorithm but also over our OpenMP implementation of our algorithm using all 12 cores of our dual-socket sext-core CPU (which itself achieved very respectable speedups over serial implementations).

The first step to mitigate the problem of high input data to the GPU was to overlap message sending from the CPU to the GPU with kernel computations on the GPU using the stream construct within CUDA. Using this technique, a small amount of input tile data is sent to the GPU initially. Then, while the CUDA kernels process this data, the next set of tiles can be transmitted simultaneously to the GPU on a separate stream. This is only possible because the computations required to evaluate the presence of a threat within a given tile is independent of the data in any of the other tiles. Using this approach, the majority of the message passing work could be hidden. We discovered that the problem (even in the final version of our code) was still memory bound (versus compute bound), but this should permit us in the future (given that we are processing at faster than real-time rates already) to perform additional computations, including future work on target identification.

The second step to deal with the extremely high input data rates to the GPU was to ensure that the data once stored in the global off-chip memory on the video card (the only memory space accessible from the CPU), is efficiently managed by the GPU and cached efficiently in the on-chip registers and L1 and L2 caches of the GPU. This was achieved by two separate design decisions: the collaborative approach to processing individual tiles simultaneously using many individual CUDA processing threads (or cores) and the use of texture memory for the input video data.

Our initial direct port of the algorithm implemented on the CPU (and virtually unchanged when we incorporated the use of OpenMP to make use of all the CPU cores) to the GPU involved assigning each input tile to a separate thread within the GPU. Unfortunately, this resulted in very poor locality of memory accesses and inefficient use of the L1 and L2 caches. When we switched to assigning each data tile to 256 separate CUDA cores with each core responsible for a separate column of the tile, we achieved a very large speedup because now all the memory accesses were contiguous in memory and we simultaneously achieved the 32 fold speedup of coalesced memory loads (over uncoalesced memory loads) and much better use of the GPU caches because all the cores in a CUDA warp were focused on a very narrow range of input data. Secondly, all of the input tile data was stored in read-only cached texture memory which added significantly to the efficiency of the GPUs memory system.

The timing diagram and relative speedup charts, shown below, show the relative advantages achieved by the individual optimizations discussed above.

Conclusions

More work is planned to perform additional computations on the video input data. Efforts are underway to identify the threats that are detected so as to distinguish the UAVs from birds or even close range bugs. The fact that we are still memory bound at this stage and significantly faster than real-time implies that such enhancements to the data processing should be possible.