SBIR Topic A05-121

AUTOMATIC EXTRACTION OF URBAN FEATURES FROM TERRESTRIAL LIDAR SYSTEMS

1. COVER SHEET (see attached)

2. IDENTIFICATION & SIGNIFICANCE OF THE OPPORTUNITY

The objective of this proposal is to demonstrate the feasibility of using data from terrestrial LIDAR-based systems to extract 3-D complex urban features (e.g. Building facades, walls, fences, poles, etc.) automatically, in detail, and eventually with near real-time feedback.

While there has been a steady increase in the resolution and size of LIDAR datasets, there has been an increase in demand for full-resolution interactive processing and visualization of this data. The requirements of this processing create a computational bottleneck, suggesting the application of Graphics Processing Units (GPU) to improving performance. The advancements in GPU technologies to processing digital imagery indicate that tremendous speedup of image processing algorithms can be realized through the use of GPUs. GPUs prove to be much faster than the CPU at applying convolutions like Gaussian smoothing, edge detection, and averaging filters at high resolutions (1024x1024 and higher). In the case of a 3x3 convolution filter, the GPU is up to 30 times faster than the CPU, even after communication between main memory and the GPU is accounted for. For larger convolutions such as 5x5 filters, the GPU gains even greater advantage, performing 50-60 times faster. [Payne, Owen, Belkasim, Flynn 2005]

This proposal focuses on the issues surrounding the efficient use of GPU resources to alleviate the computational bottleneck, particularly on what kind of hardware and with which GPU libraries to implement the processing. In the background we will discuss an overview of GPU processing, how it applies to the processing of LIDAR data, and the approaches to implement the 3-D urban feature extraction algorithms.

2.1 Background

Previous Work

The Galaxy Global / ISR research team proposes to investigate the application of high performance GPU-based parallel algorithms for processing very large datasets obtained from ground-based LIDAR. The team has worked together previously to develop digital image processing algorithms for processing airborne LIDAR data and parallel algorithms for improving processing performance. The team will leverage this experience in completing the work described in this proposal.

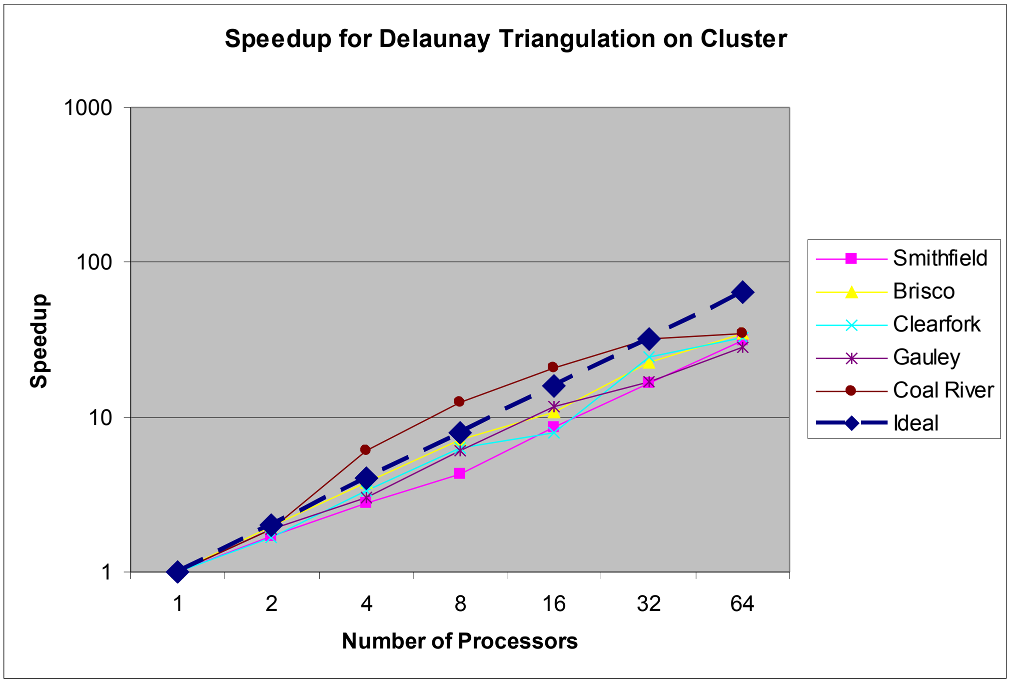


Figure 1. Our previous work on distributed-memory processing of large LIDAR datasets shows linear speedup for a variety of data sizes (Ideal linear speedup is depicted as a dashed blue line). Use of GPU computation promises to increase this performance.

As depicted in Figure 1, our previous use of distributed-memory CPU-bound LIDAR processing algorithms has yielded very good results. As an example, on our largest dataset (80.5 Megabyte LIDAR point cloud), processing time for a triangulation computation required almost 24 hours on a single CPU versus 10 minutes on our parallel cluster comprising 64 processors. In this previous work, the research team investigated and implemented computational geometry algorithms (rasterization and triangulation), as well as feature extraction algorithms. The feature extraction algorithms included building detection, road detection, and a variety of other image processing tools useful in identifying features in a LIDAR point cloud. Figure 2 depicts the feature extraction results from a sample dataset using these algorithms. While the feature extraction algorithms are currently serial (single CPU), this Phase I effort investigates approaches to parallelizing them for higher performance.

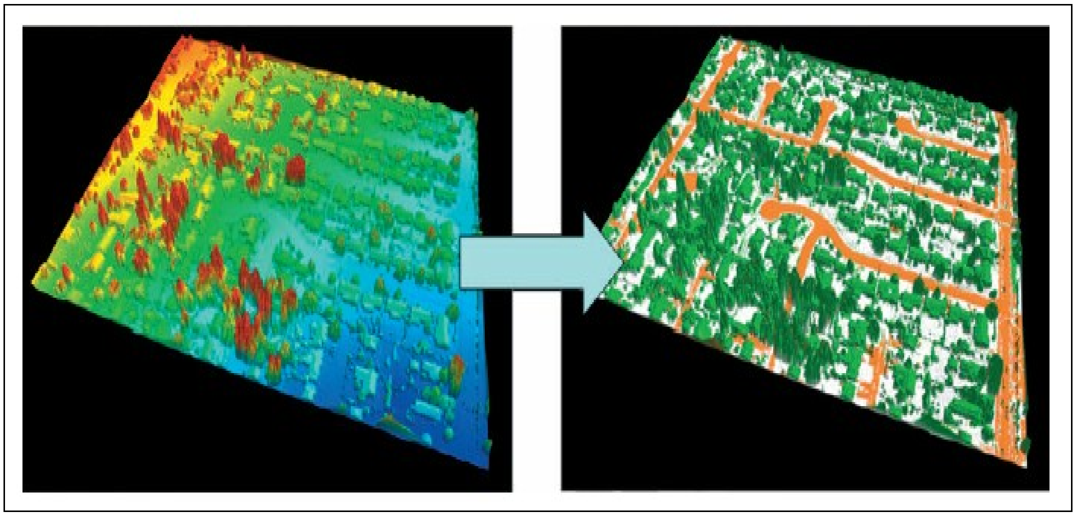


Figure 2. The Galaxy Global / ISR research team has worked together previously to design feature extraction algorithms for airborne LIDAR data. On the left is a LIDAR image colored according to elevation, while on the right is the same image with buildings, roads, ground, and trees identified and colored according to classification.

Current Research

Research has been previously reported on the advantages of GPU computation over CPU computation with respect to convolution filters [Viola et al. 2003], Fast Fourier Transforms (FFT) [Moreland and Angel 2001; Wloka 2003], and wavelet analysis [Wang et al. 2004], but little research has addressed the application of these approaches with respect to LIDAR data and near real-time 3-D feature extraction.

Many advancements have been made in the GPU industry. Among those advancements are the addition of high-throughput programmable floating point processors, efficient special purpose engines, and a flexible memory subsystem that supports a wide range of DRAM types. The NVIDIA GeForce 6800 supports calculation of up to 400 million vertices per second, and 120+ Gflops peak (equal to six 5-GHz Pentium 4 processors). [Montrym et al. 2005]

2.1.1 GPU Architecture

There are three major factors driving the current GPU architecture, Programmability, Parallelism, and Memory.

Programmability: Programmable elements, consists of two programs, vertex programs operating on each independent vertex, and fragment programs for operation on each independent pixels. Each program is executed in parallel on every pixel and vertex in the command stream. Each execution creates a thread with an independent input and output registers, allowing developers to add value with proprietary algorithms.

Parallelism: Graphics rendering problems involve a great deal of data parallelism. These high levels of parallelism permit the efficient deployment of broadly and deeply parallel computational resources. Contrasting CPUs and GPUs makes it easier to understand the motivation behind GPU architecture. The GPU offers more independent calculations that a typical CPU. The GPU can afford larger amounts of floating-point computational power because the control overhead per operation is lower than that for a CPU.

Memory: Because of bandwidth limitations, GPU manufacturers aim for 100 percent memory bandwidth utilization. This forces the internal processors to be latency tolerant and to respect page locality. The memory model supports lossless compression and decompression, and effectively saturates the memory bandwidth. This accounts for the high data through-put that can be expected from current GPUs.

2.1.2 Algorithm Design

When approaching the issue of how to implement computational geometry and feature extraction algorithms for LIDAR processing on a GPU system, there are a number of GPU programming paradigms from which to choose. The choice is typically governed by the available hardware. However, for any given hardware architecture, there are often a number of conflicting options when deciding how best to program the system to accomplish the task at hand. The basic issues arise from the limited amount of memory available for processing on the GPU. This will require determining a methodology of splitting the very large data sets into smaller subset to generate 3-D models of detailed features. Among the technologies available for 3-D modeling are OpenGL, cg, and DirectX. Each technology has its own set of features and issues that will be implemented and tested for performance, accuracy, and usability.

Designing a GPU algorithm involves partitioning a problem into independently computable pieces that can be evaluated independently. The portions of the program produce an output from the GPU, which is then combined with the output from the prior processing step to result in a high resolution virtual environment. GPU programming in essence entails writing data-parallel algorithms (parallelizable by partitioning data prior to computation and distributing it to distinct processing elements for parallel execution and are best served by execution on distributed memory machines. In essence a GPU can be thought of as separate machine within a computer, substantial speed-up of a GPU based algorithm can be achieved with either a single computer with multiple GPU's or a cluster of computers each with a GPU. A large area of the research involves determining the best method for achieving near real-time feature

extraction from LIDAR with equipment that is field deployable.
As mentioned above, there is a certain amount of flexibility in deciding how best to implement an algorithm on a given GPU resource. Once the architecture and attributes of the algorithm have been identified, the rest of the development process proceeds by addressing the following:

- Partionioning -- also known as domain decomposition -- a candidate algorithms examined to assess its parallelizablity, and identify which portions of the program may be enhanced by execution on a GPU and eventually split among several GPUs.
- Communication -- the partitioned algorithm is recast in terms of information flow and control among processing elements on the the GPU(s) and CPU(s). The aspect is often the crux of designing efficient GPU and parallel programs, in that a computation-bound serial program can easily become a communications-bound parallel program.
- Agglomeration -- this aspect represents a synthesis of partitioning and communication analysis. Here, the software design in examined in light of the available CPU, and GPU resources and modified to improve performance. Typically, this takes the form of maximizing the size of messages passed from CPU to GPU, or among multiple CPUs and GPUs. For example, for even the fastest communication fabric, it is generally desirable to agglomerate messages within a local processor before broadcasting those messages together as a monolithic whole. This increases system performance (decreases total time spent in communications with respect to computation).. Generally, more coarsely-grained algorithms typically are associated with very rare, relatively large messages.
- Mapping -- is the physical process of determining how the programmatic elements identified above map to the processing elements in an GPU/CPU system.

3. PHASE I TECHNICAL OBJECTIVES

The overall objective of the proposal is to develop software for near real-time automatic high-resolution 3-D urban feature extraction from ground based LIDAR.

During Phase I, we will modify the existing code from our prior aerial LIDAR feature extraction work to extract similar features from ground based LIDAR. The existing work used a structured grid to store the data point and standard image manipulation code, slightly modified, to extract buildings, roadways, trees, and power lines. The existing code was designed for operation on a CPU, or Beouwulf cluster. Table 1 outlines our technical objectives for translating this work to the GPU domain.

Table 1: Phase I Technical Objectives
1. Benchmark the existing code on the new larger datasets and look for areas that can be enhanced by the processing power of the GPU. (Task I) 2. Rework the code to operate on the GPU. This should provide a substantial speed-up of the algorithm. (Task 2) 3. Investigate the speed-up of the system by benchmarking the GPU code set. Also compare the results from the two code sets to ensure accurate feature extraction. (Task 2) 4. Use the extracted features, (i.e. feature edges from Sobel edge detection, or Fast Fourier Transform(FFT), straight line edges extracted using a Hough transform, trees and roadways from intensity filtering of the LIDAR data) to build a 3-D model of the urban environment in OpenGL for a simulated walk-through. (Task 3)

4. PHASE I - WORK PLAN

Phase I research will be restricted to showing feasibility of producing an accurate 3-D model of an urban environment using a single GPU for processing, Investigation of parallelizing the code to use multiple GPU's with hope of achieving near real-time modeling will be undertaken in Phase II.

The proposed research represents very recent developments in high-performance computing and is not without risk. It is possible that the GPU-based feature-extraction algorithms will not achieve the high performance we have observed with our CPU-based parallel algorithms. In this event, we will mitigate this risk by deploying the feature extraction algorithms on a distributed memory parallel CPU system.

The Phase I work plan will include the following tasks for achieving the stated objectives:

4.1 Task I - Sobel, Prewitt, Hough, and FFT feature edge extraction on the CPU.

Task I is the main feature extraction algorithm design phase. Most of this work was completed during prior research and will now be applied to the ground based LIDAR data. Feature extraction was performed on aerial LIDAR data using Sobel, Prewitt, and FFT to detect feature edges. This gave a much clearer picture of structures on the ground. It was later found that linear features, such as buildings and roadways could be further highlighted using a Hough transform. The performance of the Hough transform was greatly enhanced by splitting the data into perfect square sections and performing a localized Hough transform on those sections of data, The resulting lines were then marked on the original data set. This was very effective in locating man made structures in very large

5,000,000 point data sets in approximately 15 minutes. This of course would prove too slow for near real-time processing. Much research indicates that a GPU is capable of rendering 1,920,000 points of data per frame at a frame rate of around 200 frames per second. These rendering processors can be used to extract feature edges at near the same rates giving the potential to extract the feature edges from 384,000,000 points per second.

4.2 Task II - Sobel, Prewitt, Hough, and FFT feature edge extraction on the GPU.

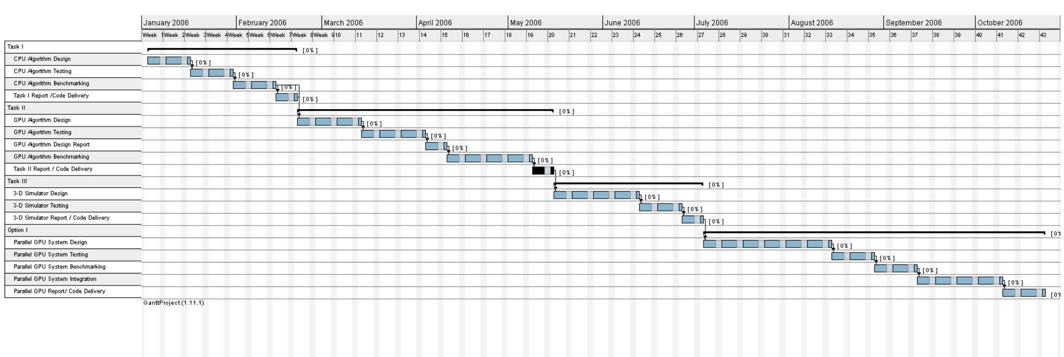
Task II reworks the algorithms from Task I to perform the feature extraction on the GPU. After the rework, results will be compared with those from Task I and both the CPU and GPU algorithm will be benchmarked to determine the speedup. The GPU algorithm will be tested multiple brand GPUs to give a general benchmark for different hardware classes.

4.3 Task III - 3-D Modeling.

Task III will use the fastest technology from Task I & II to generate a 3-D walk through model of the LIDAR data. An engine will be prototyped to allow simulation of the real-world environment. The results here will go on to Phase II where a fully functional system will be designed, possibly utilizing multiple GPU's for even faster performance.

4.4 Reporting

Weekly status reports will be provided to insure the project is progressing, and a monthly demonstration of the research conducted will be provided.



5. RELATED WORK

A paper on a past research was presented at the Appalachian Remote Sensing Conference, IFSAR & LiDAR Workshops May 10-11, 2005 titled, "High performance algorithms for LIDAR data processing and visualization" and is available upon request. The research involved the use of High Performance

Computing (HPC) resources to improve performance of processing and visualization of full-resolution LIDAR data. Triangulation and rasterization algorithms were implemented as a proof of concept and further development efforts for feature extraction were performed using commodity computer hardware.

6. RELATIONSHIP WITH FUTURE RESEARCH OR RESEARCH AND DEVELOPMENT

The research conducted during Phase I will tie directly into future work with the WV View and USA View organizations to enable Galaxy Global to assist these organizations in there endeavor to create a large conference of remote sensing companies and technologies. Future projects for Galaxy Global include the possibility of using the technology not only for modeling, but for feature recognition. The research could lead to future development in the area of biometrics using the feature extraction technology to extract facial features for use in facile recognition systems. Computerized visualization using cutting edge technology could easily bring Galaxy Global Corp into a bright future in product research.

7. Commercialization Strategy

The product can eventually be used to generate 3-D modeling for any number of uses, from generating models for surveying, land development, and tourism. A fairly interesting application would be the ability to generate walk-though simulations of national and state parks which would allow people who are unable to access areas of state parks due to physical disabilities the opportunity to enjoy them through a virtual environment. This same virtual environment could be used by the parks department to increase tourism. A less expensive unit for gathering the modeling data could be developed for use in the real-estate industry to create walk-through simulations of home for sale. The marketing for such a product has endless possibilities, from actual real world applications to the ability to generate 3-D maps for virtual reality games from real world locations.

8. KEY PERSONNEL

SCOTT L. HAMILTON

Galaxy Global, Inc.
Senior Software/Systems Engineer
EDUCATION:
Bachelor of Science in Electrical Engineering, West Virginia University Dec. 1993
CURRENT POSITION AND RESEARCH:
Senior software engineer and research scientist.
RELEVANT EXPERIENCE:
Mr. Hamilton has 26 years of experience in computer programming, and software

development. The past four years have been spend in high end visualization applications such as fingerprint matching algorithm design, facial recognition systems, and signal analysis.   His most recent research project was in 3-D feature extraction from aerial LIDAR data.

FREDRIC BERNAL

Institute for Scientific Research, Inc.
Senior Member Research Staff
EDUCATION:
B.A. in Physics from St. Mary's College of Maryland in 1997
B.S. in Computer Science from the University of Maryland in 2001
Currently enrolled in a Master's/PhD program in Computer Science at West Virginia University.

CURRENT POSITION AND RESEARCH:
He is a senior research member of the Clustering and Advanced Visualization Environment (CAVE), DVRS, DAMAGE, and WVCCG projects at the Institute for Scientific Research, Inc (ISR).

RELEVANT EXPERIENCE:

Mr. Bernal has been a member of the ISR team for four years. Mr. Bernal has been an architect/system administrator of the Black Diamond Cluster at ISR, and designed a 20 million pixel, 16-panel high-resolution display wall. In addition to parallelizing serial algorithms that can run on Black Diamond, he created a parallel rendering API which has been used in to aid ISR scientists in visualizing large scientific data sets that are produced by simulations running on Black Diamond. He has most recently designed parallel and serial algorithms for 3-D feature extraction of airborne LIDAR data that run both on the Black Diamond cluster and on a Windows Network of Workstations.

9. FACILITIES/EQUIPMENT

ISR will provide Galaxy Global time on their Black Diamond cluster. Black Diamond consists of 44 dual-processor nodes connected on a high-speed Myrinet interconnect and benchmarks at 125 GigaFlops. ISR also will provide up to 16 ATI Radeon 9800s with associated computer hardware for GPU programming.   Both of these resources will be used to demonstrate the speedup of the 3-D urban feature extraction algorithms.

10. CONSULTANTS

Institute for Scientific Research Statement of Work

The Institute for Scientific Research, Inc. will support Galaxy Global Corporation, Inc. in the development and design of LIDAR software for extracting very high-resolution 3-D urban features from advanced ground-based LIDAR sensors during:

- Phase I by:
- Providing the existing DVRS framework for LIDAR algorithms developed by ISR for extracting 3-D features from airborne LIDAR data.
- Providing LIDAR algorithm expertise to aid in extending the DVRS framework with ground-based high-resolution 3-D urban feature extraction LIDAR algorithms.
- Investigating the application of GPU programming techniques to ground-based LIDAR algorithms which will aid in processing large data sets on a single computer.
- Providing the Black Diamond Cluster and as a supercomputing resource for processing large datasets.
- Providing ISR's course-grained data parallel airborne LIDAR algorithms developed during DVRS.
- Phase II by:
- Providing parallel algorithm expertise for converting the serial ground-based LIDAR algorithms developed in Phase 1 into data parallel algorithms capable of processing large datasets.
- Investigating the parallelization of ground-based GPU LIDAR Algorithms if significant speedup in data processing was achieved on a single computer in Phase 1.
- Benchmark the results obtained in both Phases to demonstrate the speedup of parallel LIDAR algorithms and serial/parallel GPU algorithms over serial algorithms.

Phase I

The DVRS framework consists of serial LIDAR processing algorithms for extracting 3-D features from airborne-based LIDAR data. The framework was designed to be both modular and extensible so that future LIDAR algorithms could easily be added to the framework by reusing existing algorithms and code. As a result, the framework supports the chaining of algorithms so that the results of one algorithm can be fed as input into another in order to extract desired features from datasets. Thus, while all of the DVRS algorithms are designed to process airborne LIDAR data, the DVRS building and edge detection algorithms can be fed into the new ground-based 3-D urban feature extraction algorithms being developed by Galaxy Global in Phase 1. ISR will aid in the extension of the DVRS framework and will contribute its LIDAR algorithm expertise to the high-level design of the 3-D urban feature extraction algorithms.

The high-resolution data being processed in Phase 1 results in very large data sets that can quickly overtax the processing resources of a single computer. ISR will investigate the application of GPU programming techniques to serial ground-based LIDAR algorithms in Phase 1 in order to leverage the power of current graphics hardware. The matrix math that the DVRS algorithms force the CPU to do are hardware accelerated by a GPU, so the adaptation of the LIDAR algorithms should produce a significant speedup on just a single node. The results of this investigation will direct the focus in parallelizing the LIDAR algorithms.

The serial LIDAR algorithms were designed to be coarse-grained so that they could easily be parallelized and run on a cluster. The advantage in doing this is that the same algorithm could run on many machines, each processing a different slice of the overall dataset. This facilitates the processing of very large datasets that would normally overtax the resources of a single machine. ISR will provide Galaxy Global with access to the Black Diamond cluster and contribute their parallel algorithm expertise in parallelizing the algorithms. In addition ISR will provide the existing data parallel LIDAR algorithms developed in the DVRS project to aid in this process.

Phase II

Contingent on the results of the GPU investigation in Phase 1, ISR will aid Galaxy Global in the parallelization of the serial GPU LIDAR algorithms developed in Phase 1. There are two approaches that will be investigated:

1. Parallelization of the GPU algorithm to run on multiple GPUs in a single machine.
2. Parallelizing the GPU algorithm to run on a GPU cluster with one GPU per machine over a high-speed network.

The advantage of the first is that the Army could have a mobile high-powered processing workstation that could process urban LIDAR data on the fly. The advantage to the latter is that tremendous processing power can be achieved by scaling the size of the GPU cluster and its data servers. By combining the two options, ISR can then extrapolate statistics of the performance of a super GPU cluster that has multiple GPUs per machine.

11. PRIOR, CURRENT OR PENDING SUPPORT

Galaxy Global has no prior, current or pending support for a similar proposal.

12. COST PROPOSAL

See attached.

13. REFERENCES

1. Bryson R. Payne, G. Scott Owen, Saied O. Belkasim, and Patrick Flynn,

"Accelerated 2-D and 3-D Digital Image Processing on a GPU", Georgia State University (2004)

2. I. Viola, A. Kanitsar, and E. Groller, "Hardware-Based Nonlinear Filtering and Segmentation using High-Level Shading Languages. Technical Report TR-186-2-03-07.", Institute of Computer Graphics and Algorithms, Vienna University of Technology (2003)
3. K. Moreland and E. Angel, "The FFT on a GPU.", Proceedings of the ADM SIGGRAPH/EUROGRAPHICS conference on graphics hardware, 112-119. (2003)
4. J. Wang, T. Wong, P. Heng, and C. S. Leung, "Discrete Wavelet Transform on GPU." World Wide Web site available on line at Http://www.cse.cuhk.edu.hk/~ttwong/software/dwtgpu/. (2004)
5. M. M. Wloka, "Implementing a GPU-Efficient FFT.", ACM SIGGRAPH 2003 Course Notes, pages 132-137. ACM SIGGRAPH, (2003)
6. J. Montrym, and H. Moreton, "The GeForce 6800", IEEE Micro (Vol. 25 No.) 41-51., IEEE (March/April 2005)