Suryakant Patidar

Profile

Suryakant Patidar completed his B.Tech. with honors in CSE from IIIT Hyderabad and his M.Sc. in the field of High Performance Computing with GPUs. His thesis work contributed the fastest sorting algorithm (June 2009) on the GPU outperforming various algorithms published by reputed universities and industry research labs. He has been working with the state of the art parallel hardware for designing various algorithms and primitives for speeding up data and processing intensive systems. He has been among the first movers in the field of GPU computing and has published to reputed conferences. He has also mentored computer vision and computer graphics based projects on CUDA hardware as a part of his research work.

Synopsis

I am looking forward to work with a group which brings out the best in me and which directly works towards the progress of the company and in turn themselves. I believe I am a great learner and can take different paths when required which can maximize my contribution towards the group.

Interests

High Performance Computing with GPUs, Digital Image Synthesis with Ray Tracing Algorithms, Rendering large datasets (terrains, triangulated and point based models), and Parallel & Distributed computing.

Education

MS by Research in Computer Science
Thesis Title : Scalable Primitives for Data Mapping and Movement on the GPU
Advisor : Prof. P. J. Narayanan (Dean R&D)
International Institute of Information Technology Hyderabad (June 2009)
CGPA:: 10.00

BTech in Computer Science & Engineering (*Honors in Visual Information Processing***)**
International Institute of Information Technology Hyderabad (June 2006)
CGPA:: 8.00

Senior Secondary
St. Jude's Hr. Sec. School Khargone (M.P.) (June 2002)
Percentage:: 87%

Awards / Experience

GE Foundation Scholar **July 2005 - June 2007**
One of the 45 graduate students in India to have received this scholarship awarded by The General Electric Foundation.

Entrepreneur **Jan 2008 - March 2009**
I worked as the Chief Architect and Technical Lead on a couple of ideas in the field of Healthcare and GPU based solutions over the duration of a year.

Research Assistant **July 2006 - Dec 2008**
Advanced course work and research in the field of upcoming parallel hardwares(GPU). Submission of research papers to national and international conferences.

Teaching Assistant for Computer Graphics **Spring 2006**
Work included grading of answer scripts, conducting tutorials and grading weekly programming assignments.

Publications

Suryakant Patidar, P. J. Narayanan. Scalable Split and Sort Primitives using Ordered Atomic Operations on the GPU, ACM Conference on High Performance Graphics (Poster), April 2009.

Vibhav Vineet, Harish P K, **Suryakant Patidar**, P. J. Narayanan. Fast Minimum Spanning Tree for Large Graphs on the GPU, ACM Conference on High Performance Graphics (Paper), April 2009.

Kishore Kothapalli, Rishabh Mukherjee, Suhail Rehman, P. J. Narayanan, Kannan Srinathan, **Suryakant Patidar**. A Performance Prediction Model for the CUDA GPGPU Platform. International Conference on High Performance Computing, May 2009.

Suryakant Patidar, P. J. Narayanan. Ray Casting Deformable models on the GPU, In Proceedings of the 7th Indian Conference on Computer Vision, Graphics and Image Processing.(ICVGIP 2008).

Shiben Bhattacharjee, **Suryakant Patidar**, P. J. Narayanan. Real-time Rendering and Manipulation of Large Terrains, In Proceedings of the 7th Indian Conference on Computer Vision, Graphics and Image Processing. (ICVGIP 2008).

Soumyajit Deb, Shiben Bhattacharjee, **Suryakant Patidar**, P. J. Narayanan. Real-time Streaming and Rendering of Terrains In Proceedings of the 6th Indian Conference on Computer Vision, Graphics and Image Processing. (ICVGIP 2006).

Technical Reports

Suryakant Patidar, P. J. Narayanan. Scalable Split and Sort Primitives using Ordered Atomic Operations on the GPU, IIIT/TR/2009/99, February 2009

Suryakant Patidar, Shiben Bhattacharjee, Jag Mohan Singh, P. J. Narayanan. Exploiting the Shader Model 4.0 Architecture, IIIT/TR/2007/145, March 2007.

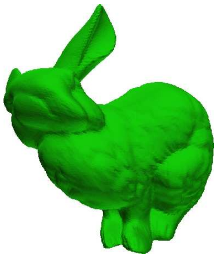
Relevant Courses

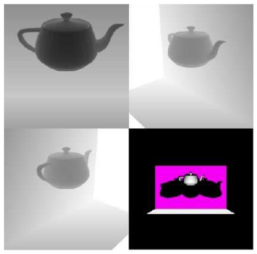
Advance GPU Programming, Computer Graphics, Computational Geometry, Pattern Recognition, Computer Vision, Digital Image Processing, Introduction to Visualization, Artificial Intelligence, Linear Algebra, Information Extraction, Data Compression, Algorithms, Computer Networks.

Project Work

Ray Tracing Heavy Deformable Models on the GPU : Due to high computation cost of building data structures required for ray tracing, only static or semi deformable/animated environments are supported by majority of the GPU based ray tracers. With our existing work in the field of data structures like split and sort, we build an view dependent image-space data structure from scratch for each frame and ray trace completely deformable heavy models at real time rates using an off-the-shelf GPU. The data structure we use for ray cast can be used as a 3-d grid for secondary rays. Fast enumeration of grid cells intersecting a ray can be done using a 3DDDA algorithm. We traverse these grid cells and perform ray-triangle intersections for the secondary rays. Other secondary rays like reflection and refracted rays also require enumeration of cells to compute the effects. The method supports a complete change in the topology of the model making a good candidate for highly deformable objects. This work is an extension of the previous work done which is explained in below two projects.

Scalable Split and Sort Primitives using Ordered Atomic Operations on the GPU : We present efficient implementations of two primitives for data mapping and distribution on the massively multithreaded architecture of the GPUs in this paper. The *split* primitive distributes elements of a list according to their category. Split is an important operation for data mapping and is used to build data structures, distribute work load, etc., in a massively parallel environment. The *gather/scatter* primitive performs fast, distributed data movement. Efficient data movement is critical to high performance on the GPUs as suboptimal memory accesses can pay heavy penalties. The split we implement is a generalization of the binary split and is implemented using the shared memory and the atomic operations available on them. The split performance scales logarithmically with the number of categories, linearly with the list length, and linearly with the number of cores on the GPU. This makes it useful for applications that deal with large data sets. We also present a variant of split that partitions the indexes of records. This facilitates the use of the GPU as a co-processor for split or sort, with the actual data movement handled separately. We can compute the split indexes for a list of 32 million records in 180 milliseconds for a 32-bit key and in 800 ms for a 96-bit key. The instantaneous locality of memory references play a critical role in data movement on the current GPU memory architectures. For scatter and gather involving large records, we use collective data movement in which multiple threads cooperate on individual records to improve the instantaneous locality. The split, gather, and their combinations find many applications and expect our primitives will be used by future GPU programmers. We show sorting of 16 million 128-byte records in 379 milliseconds with 4-byte keys and in 556 ms with 8-byte keys.

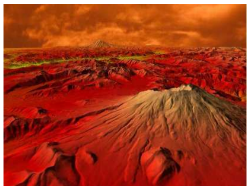
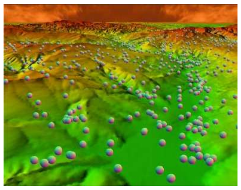
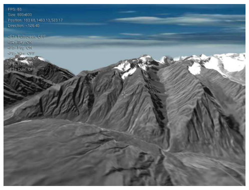
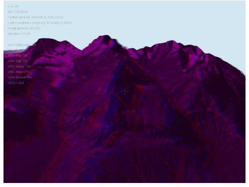




Ray Casting Heavy Deformable Models on the GPU : The GPUs pack high computation power and a restricted architecture into easily available hardware today. They are now used as computation co-processors and come with programming models that treat them as standard parallel architectures. We explore the problem of real time ray casting of large deformable models (over a million triangles) on large displays (a million pixels) on an off-the-shelf GPU in this paper. Ray casting is an inherently parallel and highly compute intensive operation. We build a GPU efficient three-dimensional data structure for this purpose and a corresponding algorithm that uses it for fast ray casting. We also present fast methods to build the data structure on the SIMD GPUs, including a fast multi-split operation. We achieve real-time ray-casting of a million triangle model onto a million pixels on current Nvidia GPUs using the CUDA model. Results are presented on the data structure building and ray casting on a number of models. The ideas presented here are likely to extend to later models and architectures of the GPU as well as to other multi core architectures.

Exploiting the Shader Model 4.0 Architecture : The Direct3D10/SM4.0 system is the 4^{th} generation programmable graphics processing units (GPUs) architecture. The new pipeline introduces significant additions and changes to prior generation pipeline. We explore these new features and experiment to judge their performance. The main facilities introduced that we ponder upon are, Unified Architecture providing common features set for all programmable stages, Geometry Shader which is a new programmable stage capable of generating additional primitives, Stream output with which primitive data can be streamed to memory, Array textures and primitive level redirection to different frame buffers through layered rendering. We analyze our implementations and with experimentation, we draw conclusions on their efficient usage and provide some of their limitations. We thus present a number of applications viz. Rendering Geometry Images, Two-level Culling, Subdivision on Geometry Shader, Multiple Dynamic Light Shadows, Motion Blur using Layered Rendering, Interactive Physics on Terrains.

Infinite GPU Resident Terrains : Terrains are large geometric objects and provide many challenges for real-time rendering and interactive editing. We describe a representation which is built upon regular tiles for terrains using *Fixed-in-Memory Tiles (FMT)*, a fixed-size grid of height values. FMTs have a fixed memory size, but the resolution depends on the view distance. The terrain is cached on the GPU in terms of blocks of FMTs at an appropriate resolution and is rendered from it. We use a novel 2-level frustum culling scheme in which the geometry shader culls the tiles in terms of *tilelets*. The GPU cache is updated as the viewpoint changes to keep it roughly centered around the viewing area. The tiled structure of the terrain representation allows modification and editing of the terrain as well as computing interactions with other objects with low CPU involvement. Our system achieves a rendering rate of over 140 frames per second with terrain modification and interactions and a triangle rate of over 200 million per second on an Nvidia 8800GTX GPU for large terrains.

Terrain Information System : The project was funded by DRDO, India. The aim was to develop a terrain system which provides visualization of terrain data provided with height map and corresponding satellite texture of the region. Various features and optimizations like, Annotation Editing, Fly/Walk/Demo mode, Level of Details, Stereo Modes etc. were provided. A tile based approach was used against a TIN for maintaining the regularity as it makes it easier to query the terrain for factors like 3D position, contours etc. Since we used a tile based approach a discrete Level of Deatail approach was used in order to off load rendering requirements.

Streaming an MPEG video over wireless LAN from a server to PDAs : This projects deals with streaming of MPEG videos over wireless LAN from a server to PDA in the vicinity. VLC is the media player which was cross compiled for a PDA. The system supports IPV6 and IPV4, and can be used with network protocols TCP and UDP. The main contribution of this project was to provide the ability to change the bitrate on the fly during video streaming. A part of VLC Player code was updated which was responsible for bitrate of the streaming data. A compact GUI for the PDA was provided built with Qtopia and VLC was compiled for the ARM processor to play the videos on the PDA.

Computer Skills

Graphics APIs :: Nvidia CUDA, OpenGL, Cg/GLSL
GUI APIs :: Qt, Qtopia, SDL/GLUT
Programming Environments :: Vim, Microsoft Visual Studio and .NET
Programming Languages :: GNU/C, GNU/C++, VC++
Scripting Languages :: Shell, Perl, Matlab
Documentation :: Latex, Doxygen, GNUPlot, Microsoft Office
Tools :: MySQL, GIMP, Adobe Photoshop
Internet :: Ruby on Rails, PHP, CSS, Javascript
Operating Systems :: GNU/Linux, Windows 2000/XP

Referees

Prof. P. J. Narayanan
pjn@iiit.ac.in
Dean R&D, IIIT Hyderabad

Prof. C. V. Jawahar
jawahar@iiit.ac.in
Professor, IIIT Hyderabad

Contact Information

Center for Visual Information Technology
International Institute of Information Technology Hyderabad
Gachibowli, Hyderabad-500032, INDIA

Phone: (+91) 9866590104
E-mail: skp@skp.co.in
WWW: http://skp.co.in