Shell Mars extension raises brownfield stakes
Statoil settles Åsgard gas compressor contest
Umbilicals up the ante

PLUS: WHY SPEC BUILD FLOATING PRODUCTION UNITS SHOULD CARRY A FINANCIAL HEALTH WARNING
How gaming has aided GPU rendering for volume visualization

The latest top end 3D graphics cards can now support GPU rendering for volume visualization and instantaneous processing of volume data. Paradigm’s Huw James, Evgeny Ragoza and Tatyana Kostrova explain what this generation of cards makes possible.

3D surface visualization and volume rendering are extremely graphic-intensive and both have typically used top end graphics computers and graphics cards for the best possible performance.

The consumer-driven appetite for very large video game environments has been behind the push for performance improvements and price reductions for leading 3D graphics cards. The variable opacity used in volumetric rendering causes an increase of many orders of magnitude in the number of operations performed. For example, an opaque display of a seismic cube of 1000 lines, 1000 cross lines and 1000 depth samples requires rendering of 6x1000x1000 samples for the six faces for a total of 6 million operations. This same cube, rendered with variable opacity so that its interior can be seen (Figure 1), requires rendering of 1000x1000x1000 samples, for a total of 1 billion operations.

This difference makes volume rendering of large data objects dependent on the bandwidth and computing capabilities all the way from the disk to the graphics card and final display to the user. The latest graphics cards have increased their on-board memory and on-board compute power to offer new opportunities for significant performance increases for volume rendering and similar parallel problems.

3D graphics cards

3D graphics cards have evolved to offer a greater parallelism of mathematical computations. Surface rendering requires large amounts of coordinate transformations to compute screen coordinates from 3D world coordinates and to solve the lighting equations involved in visualization. Hardware manufacturer use of pipelining and many parallel arithmetic units have sped up these operations. Once the location and lighting effects are known, the surface elements have to be rendered across many screen positions; this operation requires many similar pixel interpolation operations which have also been parallelized in hardware for greater performance.

3D graphical applications and video games have typically generated a stream of graphics commands from the CPU to the graphics card using a 3D graphics language such as the open standard OpenGL or Microsoft’s proprietary Direct3D. The optimization of this graphics stream rendering was previously left to the manufacturer of the graphics card.

The parallelism, speed and compute power of these graphics cards have also attracted high performance computing (HPC) applications; this activity is commonly called general purpose graphic processing unit (GPGPU). The video game market provides the economy of scale of the broader consumer market to support the expensive development costs of the
The HPC market has encouraged the development of two new programming languages that allow programmers to exploit the parallel computing power of GPUs. These languages are the open standard OpenCL and Nvidia’s proprietary Compute Unified Device Architecture (CUDA). These languages allow the programmer to manage parallelism including the Single Instruction Multiple Data (SIMD) paradigm that GPU cores support. Where problems can be suitably parallelised on the GPU, the chips can drive GPU memory to its maximum extent and chips can omit many of the circuits that allow CPU chips to deliver fast serial performance. This allows GPUs to deliver fast results from massive parallelism with smaller power requirements for a added advantage of less energy expended per operation.

**Typical top end**

A typical top end interpretation workstation using a modern graphics card is shown in Figure 2. In this example, the interface from CPU to GPU is limited to 5GB/s, the interfaces from GPU cores to display memory is limited to 144GB/s and the interface from CPU core to GPU memory is limited to 32GB/s. In this instance GPU memory size has grown to 6GB. This memory has typically been used for graphics data in 3D visualization and volume rendering systems.

The recent growth in size of display memory from 500MB to 1GB and now to 6GB has completely changed the game. With this memory size there is room to allocate greater than 2GB of display memory to post stack seismic data and still have room for working buffers to hold compute results, buffers of geometric data to display and the display buffers themselves. Once the seismic data is on the graphics card, it is possible to perform rendering on the GPU using OpenGL or a GPU language such as CUDA wherever appropriate.

Operations such as zoom, translate, rotate or change of color and opacity can be re-rendered exclusively on the GPU with no further transmission of data across the CPU-GPU interface, merely the transmission of control instructions. This latest graphics card is rated by its manufacturer at 4X performance to its predecessor for some operations. The relatively small increase of 1.5X in memory size and 2X in cores has a more dramatic effect since it allows significant sized volumes to be stored on the GPU and thus reduce a lot of the I/O traffic across the CPU/GPU interface. This widens the advantage of GPU processing versus CPU processing. GPU rendering on the latest graphics boards can be greater than 8X faster than graphics rendering on the previous top end board.

GPU-based rendering also re-combines the two streams of GPGPU and 3D graphics usage of the GPU. If HPC data is to be shown to the user for interpretation, there is no need to return results across the GPU/CPU interface and also there is no extra cost to transferring data to the GPU. If the user is going to view the results, then sending data in some form from the CPU to GPU is not optional. This further widens the cost saving of GPU versus CPU. The cost of I/O to the GPU is also reduced on the latest cards since they allow I/O to be overlapped with GPU computation for graphics or for GPGPU usage.

Obviously not all application problems are 100% suitable for GPU computation but the oil & gas industry is rich with parallel data and problems that are well suited to parallel computation,
so we can expect to see broader usage of GPU technology. As an example, 2GB of seismic data can be band pass filtered using forward and inverse FFTs instantaneously using 400+ GPU cores. 

There is no display that can be generated on the GPU that cannot be generated on the CPU, but the increase in speed of GPU rendering allows computational power to be applied to improving the display quality of volume rendering and other 3D displays. Once the seismic data is on the GPU and a view is chosen by the user, then the display can be optimized for the resolution chosen and the display quality improved without cost. Such displays could be calculated on the CPU but would then need to be transmitted over the CPU/GPU interface, because this transfer is currently and historically the limiting bottleneck most users have chosen performance over quality and settled for sub optimal displays. GPU rendering makes such compromises unnecessary and will yield higher quality interpretation that justifies other improvements in seismic acquisition and imaging upstream of the interpretation process.

Volume rendering
An example of higher quality volume rendering is shown in Figure 3. This is an image of a channel extracted from a 3D seismic volume from an offshore Indonesia data set. It can be seen that rendering artifacts are almost non-existent compared to previous publications of this same view of this channel. The channel is at a depth of about 1400m in a very complex data set as can be found in Indonesia. The breaks in the channel occur in areas where near vertical faults have penetrated.

Another example which is from the Taranaki basin Offshore New Zealand is shown in Figure 4. This example shows several strong channel meanders with other channels deeper in the scene, without other rendering artifacts.

Both of these examples show the quality that can be achieved with GPU rendering. The improvement in performance and interactivity can only be fully understood by direct experience.

Conclusions
GPU rendering re-combines the dual streams of GPU usage for HPC and for 3D graphics. The advantages of using GPU computing where appropriate are very clear. Users’ first interaction will be with data resident on the GPU using the compute power of the GPU. Increasing amounts of software will be written in languages that support massive parallelism and the SIMD paradigm.

The authors thank Clyde Petroleum for the data set from Offshore Indonesia and AWE for the Taranaki data set. We thank our colleague Bruno de Ribet for the image from the Taranaki data set.

Huw James has worked in R&D on 3D seismic navigation, acquisition, processing, interpretation and visualization for GSI, Arco Oil & Gas, and Western Geophysical. Schlumberger GeoQuest and Paradigm, developing multiple hardware and software systems and products, as well as working in operations interpreting and processing seismic data from offshore basins around the world.

Evgeny Ragoza has worked in Paradigm R&D on 3D seismic imaging, parallel processing and visualization for 16 years. He worked in R&D developing hardware and software for HPC systems.

Tatyana Kostrova has worked in Paradigm R&D on volume rendering with interpretation applications for 12 years, including GPU rendering.