GPGPU vs. Traditional Processing for Radar Systems

GPUs Offer Dual Solution for Radar and Video Display Function

Using a GPGPU as both a display and a co-processor to the CPU reduces the cost of military C2 radar systems. But careful design of the software to manage the GPU's resources is essential.

Dr. David G. Johnson, Technical Director
Cambridge Pixel

The use of GPUs for general purpose processing offers a lot of advantages in military system designs. GPGPUs make use of their function as a display processor, but they can also be used to combine the two roles to support both real-time data processing and display processing. The dual role of the GPU as a co-processor and a display engine for multi-layered graphics results in some tough challenges for the software design.

The evolution of the general-purpose x86 processor has moved from increasing clock speeds to more processing cores, which allow software tasks to run concurrently. Separate to this evolution, graphics cards have moved from display processors to become general-purpose processing modules that serve as co-processors for the CPU. Development in software architectures and programming languages now offer the potential of a distribution of processing between the CPU and the GPU, possibly even leaving the system to distribute the processing load. This is interesting, but like any attempt to automatically allocate resources, getting it right, and certainly getting it optimal, is hard.

Figure 1

The Norwegian Coastguard's KV Svalbard icebreaker and offshore patrol vessel uses a co-operative CPU and GPU radar and video distribution and display solution to interface three radar and three cameras and deliver real-time data into two multifunction consoles, for display with maps and tracks.

A software engineer can develop code to run on the CPU and separately code for the GPU, but effective interaction of the processes still requires care. This is especially true when the GPU is being used concurrently for general-purpose data processing and complex display processing.

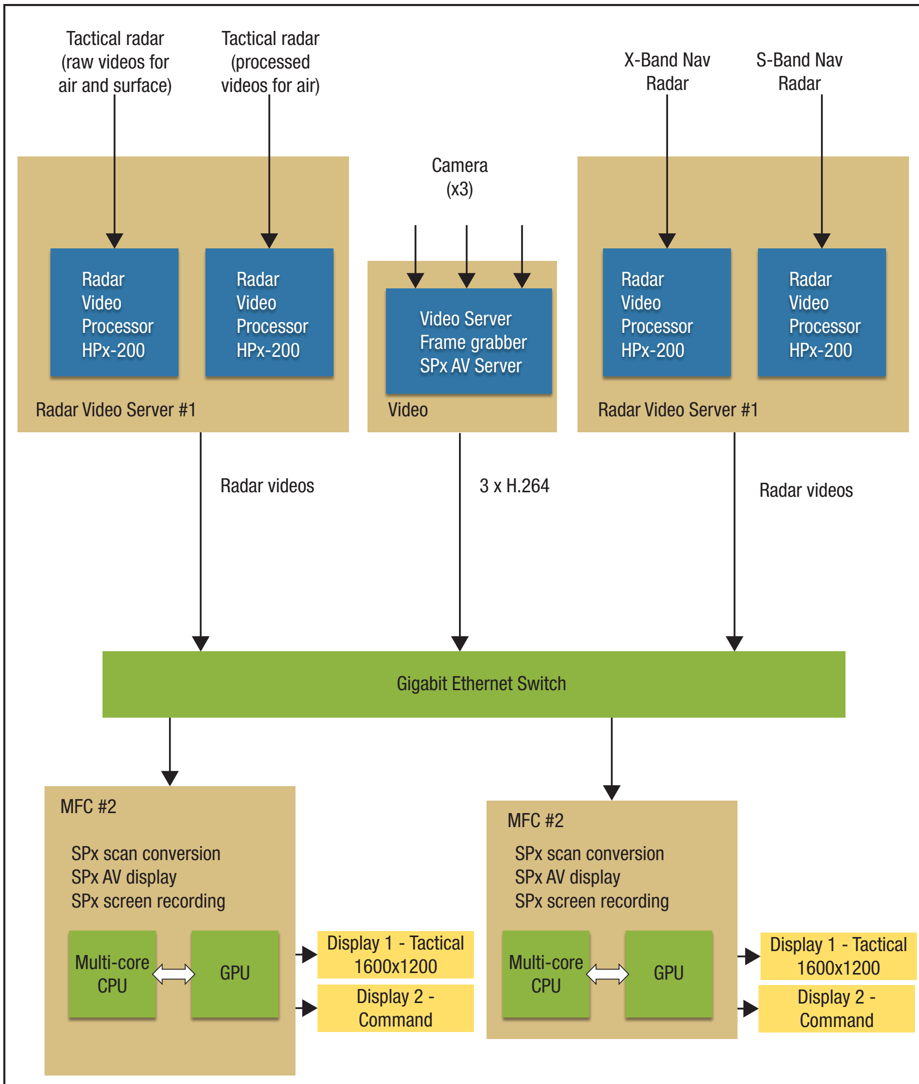

Figure 2

Radar and video servers provide capture and processing of sensor data prior to distribution to multifunction displays for presentation of video with maps and graphics.

Military Display Application

It's useful to consider this challenge in the context of a military display application presenting real-time sensor data, such as radar and camera video, along with maps and graphics. Such a system needs to process a lot of data, while presenting a responsive user-interface to an operator. A multicore processor coupled with a graphics processor, which is capable of serving as a data processor as well as a display processor, allows the solution to be implemented in industry-standard single board computers. This offers the benefits of lower system costs and easier long-term support through standard technology refresh.

The demands of processing and displaying multiple sensor streams, including compressed video, radar imagery and complex maps, will stretch the hardware. And a native software implementation may not achieve the desired performance goals or provide a full range of capabilities. Using CPU and GPU architectures to support real-time, feature-rich displays that permit network streams of compressed radar and video data to be processed and displayed requires careful consideration.

Multi-Role GPU

Beyond its display role, the modern GPU provides features for parallel data computations and dedicated capabilities for video compression and decompression. Most modern GPUs contain modules that are specifically designed for H.264 decompression, and some of these also contain modules for compression. These modules do not perform all the functions of the video processing, only major portions of them such as motion compensation and slice level decoding.

In most cases it means that H.264 decompression can be viewed as largely independent from the display processor functions of the GPU. However, because decompressed video is normally displayed on the screen, the two roles overlap and application software must consider the data processing, video compression and display roles together.

The level of support for video decompression using GPUs varies significantly between different graphics cards, so the best available method for decompression must be detected by the application at runtime to give the best performance. Not using the full functionality of a GPU for video decompression, when one is available, will normally lead to very high CPU usage.

Data processing using OpenCL or CUDA allows the general purpose processing features of the GPU to be utilized. However, because the data copying to a GPU still takes a measurable amount of time, it is very important that the data processing runs asynchronously to the application code, otherwise no benefit is gained. Tasks are assigned for the GPU to run, and the application code then continues to run other parts of the application until the GPU tasks complete.

GPU Memory Limitations

One area where all of the GPU roles are limited is memory usage. Although the memory bandwidth to a GPU is typically very high, the amount of memory available on the GPU is generally significantly less than the amount of system memory. Additionally, the speed when

TECH RECON

Figure 3

The Royal Navy Type 45 Destroyers utilize radar video compression but only after full verification of the system performance with compression disabled.

reading data from the GPU to system memory is much slower than moving data to the GPU. Video decompression often requires a significant amount of GPU memory due to the high number of frames that any single frame may depend upon. Similarly, when using the GPU as a data processor, the amount of memory required can be high when performing a large amount of processing.

That high memory usage is exacerbated by the need to reduce the amount of copying from GPU memory to host memory, which means it is often advantageous to keep intermediate results on the GPU using up GPU memory. To reduce the amount of memory used, and the amount of data transferred by video compression and by general purpose data processing, care must be taken in choosing the output location for video decompression and by designing the algorithms that minimize memory usage and memory bandwidth.

GPU video decompression often supports outputting the decompressed data directly into a texture that can be rendered by an accelerated rendering API (OpenGL or DirectX) without copying the data back to system memory. Similarly, if the result of the data processing is to be displayed on the screen, then the result can be moved to a texture without copying the data to system memory. Ensuring that the video data stays within the GPU memory is critical. (See online edition of this article for further discussion of radar compression issues and general architecture description of a Radar and Video Display solution.)

Radar and Video Display Solution

For a naval upgrade program for the Norwegian Coast Guard, Exelis in Chesapeake, VA, USA needed to interface to three radar signals and deliver the real-time data into two multifunction consoles, where it would be displayed with maps and tracks on board the icebreaker and offshore patrol vessel KV Svalbard (Figure 1). Three video cameras provided surveillance imagery that would be distributed on the same network to the consoles, where it would be presented on a multi-head display with the radar. A system block diagram is shown in Figure 2.

The radar and video servers capture, process and stream the sensor data onto a common Gbit network. In the case of the camera video, the capture was integrated with an H.264 encoder allowing streams of compressed video to be presented to the client consoles. The multifunction clients are based on standard PC technology and provide a high-resolution dual-head display that shows scan converted radar, maps, target overlaps and camera video data. The client is a Linux application that provides a situational awareness display to an operator, who interacts with the system during a mission.

The system has to update a dual-head, high-resolution screen with real-time radar and video imagery, while maintaining a responsive user interface for normal operator interaction. The solution is based on Cambridge Pixel's SPx software for radar and video processing. The software solution allocates tasks to the CPU and the GPU and carefully manages the flow of data to ensure that the real-time radar and video is displayed in real time.

The decompression of the H.264 video is performed using core functions in the GPU to assist with the processing. After decompression the video remains in GPU memory where it is processed with a GPU data processing task that handles adjustments to the image brightness, contrast and color conversions. Graphical data is then overlaid on the video image before it is copied from GPU memory to the display window.

For the radar, one core of the CPU handles the complex scan conversion calculations and the radar image is then transferred into GPU memory so that a GPU processing module can handle the mixing of the radar layer with map and overlays. The composed image is then copied to the display window. This update process occurs every 20 ms to ensure that the display updates with the sweep of the radar. The internal processes associated with these CPU and GPU activities are shown in Figure 3. A key feature of the SPx software is that the division of processing between the CPU and the GPU is hidden from the main application software.

However, whereas a general-purpose programming language that attempts to distribute processing between the CPU and GPU has no understanding of the high-level data flows, and therefore can't easily get a good balance, the SPx software works at a higher level with knowledge of radar and video to place functional blocks in the best location. It can then effectively manage the transfer of data through processing onto the screen.

In summary, it's a higher-level partitioning of the processing with knowledge of the type of data being manipulated, but the results provide a highly integrated solution based around commodity processing and graphics hardware, with corresponding benefits for lifetime support.

Cambridge Pixel
Royston, UK
+44 (0) 1763 852749
[www.cambridgepixel.com],