Proceedings of the Asia-Pacific Advanced Network 2011 v. 32, p. 139-142.

Deploying GPU-based Real-time DXT compression for Networked Visual Sharing

Namgon Lucas Kim and JongWon Kim*

Networked Computing Systems Lab., School of Information and Communications, Gwangju Institute of Science and Technology (GIST), Gwangju, Republic of Korea

E-Mails: {namgon, jongwon}@nm.gist.ac.kr

* Tel.: +82-62-715-2219; Fax: +82-62-715-3155
http://dx.doi.org/10.7125/APAN.32.17 ISSN 2227-3026

Abstract: The networked visual sharing application in multi-party collaboration environment needs compression of video streams due to network bandwidth limitation. For interactive real-time sharing, real-time compression of high-quality video as well as audio echo cancellation are required, which commonly depend on the availability of high-cost hard-to-setup specialized compression and echo-cancellation hardware. In this paper, by leveraging the computing power of GPU-accelerated PC (personal computer), we discuss how to support the software-only real-time compression of HD (high-definition) video streams. The chosen lightweight scheme, DXT (i.e., S3 Texture Compression), is highly matched with GPU-accelerated texture compression. By implementing GPU-accelerated DXT compression, based on CUDA (Compute Unified Device Architecture) parallel computing, and by deploying a software-based echo controller together, we can enable a low-cost solution for efficient networked visual sharing in collaboration environment.

Keywords: DXT video compression, CUDA parallel computing, echo cancellation, real-time collaboration environment, and networked visual sharing.

1. Introduction

Efficient capture, process, and transport of various types of media (i.e., video, audio, and graphics) are important for interactive real-time media sharing in multi-party collaboration environment. To satisfy the real-time requirement and the network bandwidth limitation (e.g.,

typically 1 Gbps), we should compress HD (high-definition, 1920x1080@30fps, 1.5Gbps) video streams with high-cost hard-to-setup specialized compression hardware.

In this paper, by leveraging the computing power of GPU-accelerated PC (personal computer), we discuss how to support the software-only real-time compression of HD (high-definition) video streams. The chosen lightweight scheme, DXT (i.e., S3 Texture Compression), is highly matched with GPU-accelerated texture compression [1]. The fixed-rate compression of DXT, coupled with the simple memory access, makes it suited for compressing textures in GPU-accelerated 3D graphics. DXT, more specifically DXT1, performs lossy texture compression at a fixed ratio (from 4:1 to 8:1) per 4x4-pixel block. Note that most graphic cards support GPU-accelerated decompression of DXT so that DXT-compressed texture is directly rendering by GPU itself. However, DXT compression (i.e., encoding) of HD video, in a low-specification PC, usually takes longer than real-time. In [1], Waveren proposed real-time DXT compression algorithms, implemented by Intel MMX and SSE2 extensions. FastDXT [2] is also implemented by C/C++ intrinsic of SSE2 based on Waveren's algorithm because SSE2 assembly is difficult to be applied for general purpose. With NVIDIA's CUDA (Compute Unified Device Architecture) [3, 4], which enables GPU-accelerated parallel computing, an open-source DXT compressor is included as one of texture tools [5]. It is however focused on off-line high-quality texture compression, which is not useful for interactive real-time scenarios.

In this paper, by implementing GPU-accelerated DXT compression, based on CUDA parallel computing, and by deploying a software-based echo controller together, we show how to enable a low-cost solution for efficient networked visual sharing in multi-party collaboration environment. Note that, in order to avoid the disturbing echo sounds in multi-party collaboration environment, we should install expensive echo cancellation equipments in all participating sites and jointly configure all of them for proper operation. EchoDamp is a software-based echo controller with multi-channel audio mixer, which is cheap and easy to use [6].

2. Design and Implementation

We describe our proposed deployment, to a collaboration site, of GPU-based real-time DXT compression and software-based echo cancellation as shown in Figure 1. Also, a SAGE (Scalable Adaptive Graphics Environment)-driven NeTD (Networked Tiled Display) is adopted to show ultra-high-resolution visualization [7, 8]. The SAGE-driven NeTD is capable of processing (assisted by GPU) and interactively (e.g., with resizing and moving) displaying DXT-compressed video stream. For echo cancellation, we apply the EchoDamp software-based echo cancellation, which consists of echo controller, audio processor (tied with sound card), microphone(s), and speaker(s), and pre-amplifier (provides inter-connections). It is known that this software-based EchoDamp can remove echo with the delay limit of 200~300ms. Also, for

simplicity, video and audio streams are separately delivered, which actually requires special attentions at the receiver site to synchronize separately-arrived video and audio streams.

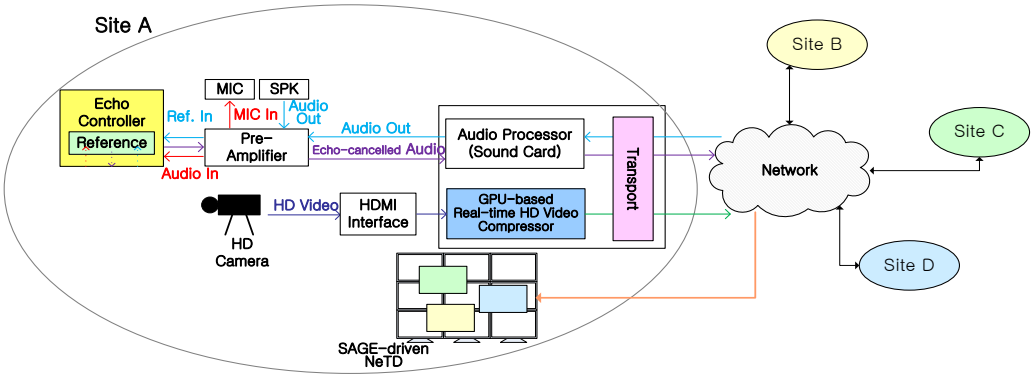

Figure 1. A deployment example for networked media sharing for multi-party

The DXT compression module is composed with color converter and DXT compressor. Each frame of input video (captured by HDMI interface) is color converted (by using CUDA parallel programming) from YUV to RGB format. The DXT compressor then encodes each frame by using CUDA parallel computing. Each GPU thread DXT-compresses each 4x4-pixel block, as shown in Figure 2. We apply FastDXT algorithm [2], as shown in Figure 2.

Figure 2. Procedure for DXT compression of a 4x4 pixel-block.

3. Experiment

The evaluation environment is mainly composed with two PC nodes with Intel CPU (single-core Xeon Nocona 3.6Ghz, Core2quad Q9300 2.5Ghz), all equipped with NVIDIA GTS 450 GPU (192 cores, 1.6 GHz, 1GB RAM). Also, for HD video capturing, we use a HDMI-interfaced camera (SONY HDR-CX150) and a HDMI interface card (Blackmagic Intensity Pro). Finally the SAGE-driven node is equipped by tiling 8 SAMSUNG Networked displays.

We compare the experimented performance of DXT compression as shown in Table 1. Note that we use CUDA toolkit 4.0 for implementation. In case of CPU-only compression, the powerful quad-core PC can handle the real-time compression and transport while the single-core PC is not reaching the 30fps target (despite the hyper-threading with two threads). The GPU-accelerated DXT compression can satisfy the real-time compression and transport with lots of remaining horsepower. However, note that since we do not optimize the processing delay yet, current prototype is still suffering from relatively longer (than 120ms) delay.

Table 1. Comparison of DXT compression performance.

Machine	O/S	Compression	FPS	B/W (Mbps)	CPU usage
Dell Precision 670 (single core)	Ubuntu 10.04.3 x64 (64bit)	CPU-based	23~24	200 ~220	95~99%
		GPU-based	30	260 ~ 270	38~40%
Dell Precision T3400 (quad core)	Ubuntu 10.04.3 x64 (64bit)	CPU-based	30	260 ~ 270	80~85%
		GPU-based	30	260 ~ 270	20~24%

4. Conclusions

We propose, by leveraging the computing power of GPU-accelerated PC (personal computer), the software-only real-time DXT compression of HD (high-definition) video streams. The realized prototype implementation based on CUDA parallel computing enables a low-cost solution for efficient networked visual sharing in multi-party collaboration environment.

Acknowledgements

- This work was supported by the Basic Research Project through a grant provided by GIST in 2011.
- This work was also achieved by utilizing PLSI supercomputing resources of Korea Institute of Science and Technology Information.

References

1. J.M.P. van Waveren, *Real-time DXT compression*. Id Software Inc., May 2006.
2. L. Renambot; B. Jeong; J. Leigh. Real-time compression for high-resolution content. *Proc. Access Grid Retreat 2007*, Chicago, Illinois, USA, May 2007.
3. NVIDIA CUDA. http://developer.nvidia.com/object/cuda.html.
4. J. Sanders; E. Kandrot. *CUDA by Example: An Introduction to General-Purpose GPU Programming*. 1st ed. Addison-Wesley Professional, July 2010.
5. nvidia-texture-tools, http://code.google.com/p/nvidia-texture-tools.
6. EchoDamp, http://www.echodamp.com.
7. SAGE, http://www.sagecommons.org.
8. B. Jeong; J. Leigh; A. Johnson; L. Renambot; M. Brown; R. Jagodic; S. Nam; H. Hur. Ultrascale collaborative visualization using a display-rich global cyber-infrastructure. *IEEE Computer Graphics and Applications*, May/June 2010; vol. 30, no. 3, pp. 71-83.