REALTIME COMPRESSION FOR HIGH-RESOLUTION CONTENT

Luc Renambot[1]*, Byungil Jeong, Jason Leigh*

University of Illinois at Chicago
Electronic Visualization Laboratory

Figure 1: High-resolution displays for collaborative science (room-size display on the left, desktop-size display on the right)

Introduction

Higher resolution content is collected or generated everyday by domain scientists, such as remote sensing imagery, large scale simulations and computer graphics animations. In the meantime, high-resolution devices are becoming extremely affordable: high-definition (1920 by 1080 pixels) displays and cameras are available for $1000, a very small fraction of their costs a few years ago. Super-high-resolution (SHD) is already being deployed from manufacturers' laboratories to the professional market (*Sony* 4k SXRD projector, *Sharp* 4k LCD panel, *Red* and *Dalsa* 4k cameras, 4k film scanners, ...). Since today's science relies heavily on collaborative exploration involving distributed groups of researchers, collaborative environments must allow the interactive sharing and analysis of such large content. To achieve interactive speed, realtime compression techniques play a key role. Even in this day of exponential increase of networking bandwidth (i.e. deployment of multi-10Gbps networks), compression is required to alleviate the next bottlenecks in modern computer systems: storage speed and memory bandwidth.

In this paper, we introduce the use of software DXT compression for high-resolution content at an interactive speed. DXT compression allows to stream full HD video over a gigabit connection where multiple gigabits were required. Moreover, it enables 4K streaming without the need for an high-end storage system or an expensive codec [7] (dedicated hardware system for encoding and decoding frames). This work will enable the AccessGrid community

---

[1] renambot@uic.edu

to stream video and visualizations at much higher resolutions while minimizing bandwidth requirements.

The contributions of this paper are as follows:

- Compression of HD and SHD in real-time using a commodity PC at a ratio of 6:1, with a good quality,
- A low bandwidth requirement for high-resolution content,
- The integration of DXT compression in the streaming environment SAGE.

DXT Compression

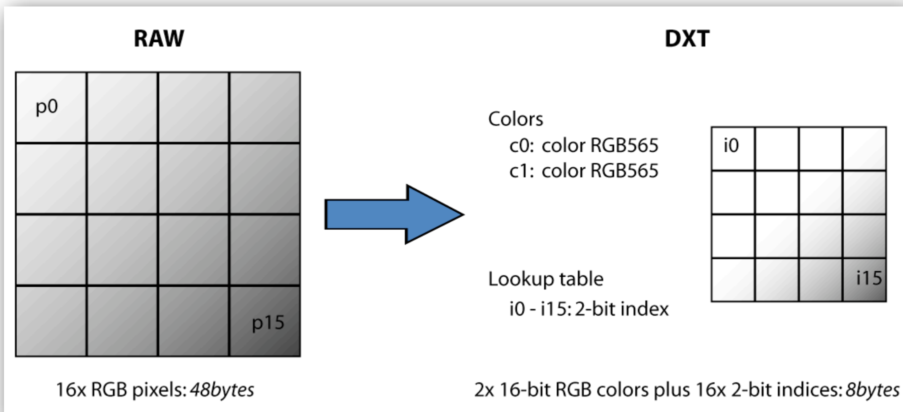

Figure 2: Compression of a 16-pixel block into DXT format.

DXT is a lossy compression technique for texture in 3D graphics [1]. It was designed to reduce the size of textures in video games when video memory was limited. Its main advantages are a fixed compression ratio of 6:1, a wide spread native support on all modern video cards, and a reasonable quality. DXT compression works as follows: it converts a 4x4 block of RGB pixels into 2x 16-bit colors (open choice) and a lookup table of 16x 2-bit color indices. In addition to the two selected colors (a, b), two intermediate colors are derived (1/3a + 2/3b, 2/3a + 1/3b), hence the 2-bit color index giving the choice of four distinct colors. So, the initial data of 48 bytes (16 pixels of 3 bytes each) can represented by 8 bytes. The six-fold data reduction enables very low bandwidth requirements for high-resolution content:

- DXT bandwidth for HD video (1920x1080) is 250 Mb/s at 30 frames per second (fps),
- DXT bandwidth for SHD film (3840x2160) is 800 Mb/s at 24 frames per second.

Up until recently, DXT compression could achieve acceptable quality but was considered slow and was mostly used as an off-line process [2]. In an innovative paper [3], van Waveren revisited the compression algorithm and proposed an assembly-language implementation using Intel multimedia instructions (SSE2). However, the implementation described in the paper is not portable. We then decided to re-implement the compression functions using C/C++ intrinsics functions [4] which are inlined and mapped to native instructions by the compiler. Intel defined a whole set of intrinsics for its multimedia instructions (MMX, SSE, SSE2, ...) which are implemented by all major compiler (Intel, GNU, Microsoft, ...). This allows the code to be ported on wide variety of platforms, in 32bit or 64bit mode, while being equivalent to the low-level non-portable assembly code. Some performance results are shown in Table 1: we show that we can compress HD at 100 frames per second and SHD at 25 frames per second on a recent processor. Higher rates can easily be achieved by using a multi-threaded implementation with an image-space decomposition on a multi-core processor.

Implementation	2k Resolution	4k Resolution
squish “C/C++” implementation	2 fps	0.7 fps
new “C/C++” implementation	10 fps	3fps
SSE2 on Pentium4 3.0Ghz	48 fps	12 fps
SSE2 on Xeon Woodcrest 3.0Ghz	100 fps	25 fps

Table 1: Compression speed in frames per second, for a single-threaded algorithm.

The reduced size of the compressed file makes it possible to playback HD and SHD content without the need for a high-end storage system as required until now. From a common 4-drive RAID-0 system, we were able to play HD content at more than 60 frames per second (fps) and SHD at 35 fps, as shown in Figure 3. Even with a single hard-drive, we were able to play SHD at 20 fps.

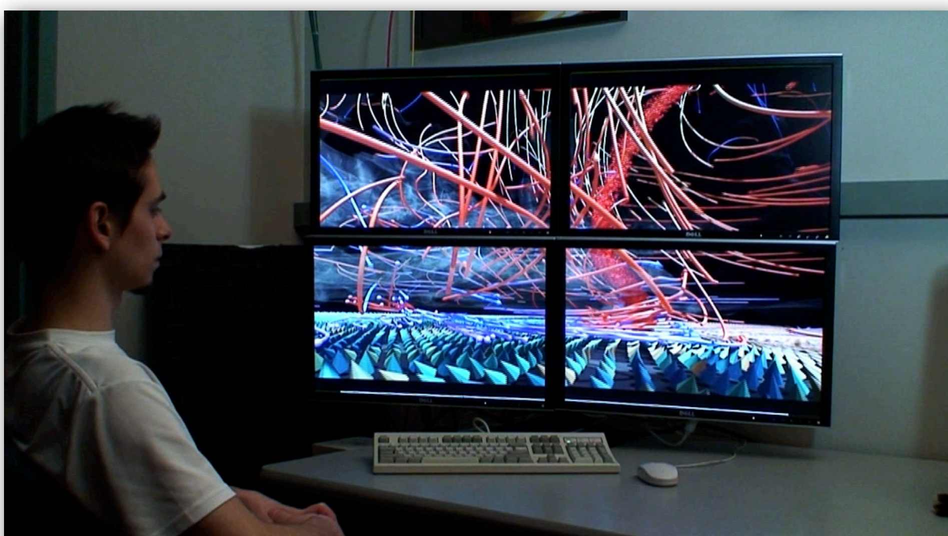

Figure 3: 4K computer-generated animation played on a 4x-HD development system

Such a performance also enables live HD video streams to be compressed in realtime. For instance, it’s possible to capture low-latency video using a consumer HDMI capture card (Blackmagic Intensity HDMI, PCI-express 1x) connected to a consumer HD camera with a

HDMI port (such cameras by Sony, Panasonic, ...). Since DXT is an intra-frame compression technique, we achieve here a low-latency streaming system for video conferencing.

We are integrating of the DXT pixel format into the SAGE environment [5]. It allows SAGE applications to provide directly compressed image buffers for streaming and remote display. Early results seem very promising: from a remote machine hosting a single hard-drive, we were able to stream and display HD content at 50 fps and SHD content at 20 fps.

To quantify the image quality loss introduced by the DXT compression (i.e. color accuracy reduction), we evaluated the error between the source image and the compressed image by a Root Mean Square (RMS) function. We applied this technique to a variety of SHD content (computer graphics, film scan, video). On average, the realtime DXT compression shows an error similar to a 75% JPEG compression (100% being the highest quality). In a few extreme cases (images with large color gradient), the compression is similar to a 25% JPEG compression.

Conclusion

In conclusion, we have shown that realtime compression of HD and SHD content is possible on modern processors using multimedia instructions. We are integrating the DXT compression into our SAGE streaming environment, that will allow interactive and collaborative environments to handle multiple HD video cameras and very high-resolution content. This work enables some convergence between the AG community and projects such as OptIPuter (www.optiputer.net) and Cinegrid (www.cinegrid.org) to stream video and visualizations at much higher resolutions while minimizing bandwidth requirements. Our software is open-source and can be accessed on our website [6].

References

1. OpenGL DXT texture compression, http://oss.sgi.com/projects//ogl-sample/registry/EXT/texture_compression_dxt1.txt
2. S. Brown, squish DXT Compression Library, http://www.sjbrown.co.uk/?code=squish
3. J.M.P. van Waveren "Real-Time DXT Compression" May 20th 2006 © 2006, Id Software, Inc., http://www.intel.com/cd/ids/developer/asmo-na/eng/324337.htm
4. Intel, How to Use Intrinsics, http://www.intel.com/cd/ids/developer/asmo-na/eng/59645.htm
5. Jeong, B., Renambot, L., Jagodic, R., Singh, R., Aguilera, J., Johnson, A., Leigh, J. *"High-Performance Dynamic Graphics Streaming for Scalable Adaptive Graphics Environment"*, Proceedings of SC06, Tampa, FL, November 2006
6. Fast DXT: a real-time DXT compression library, http://www.evl.uic.edu/cavern/fastdxt/
7. Yamaguchi, T., Nomura, M., Shirakawa, K., Fujii, T. *"SHD movie distribution system using image container with 4096x2160 pixel resolution and 36 bit color"*, IEEE International Symposium on Circuits and Systems. 23-26 May 2005, Page(s):5918 - 5921 Vol. 6.