Stereoscopic Displays and Applications XV
http://www.Stereoscopic.org

Michal Hušák, Chris Ward, "Universal System for Stereoscopic Video Visualization on WIN32 Platform", presented at Stereoscopic Displays and Applications XV, San Jose Convention Center, San Jose, California, USA

To be published as *Proceedings of SPIE* Vol. #5291A Stereoscopic Displays and Virtual Reality Systems XI, Woods, Bolas, Merritt, Benton, Editors.

Conference Chairs:
Andrew J. Woods, Ctr. for Marine Science and Technology/Curtin Univ. of Technology (Australia);
John O. Merritt, The Merritt Group;
Stephen A. Benton, MIT Media Lab.

DepthQ
Universal System for Stereoscopic Video Visualization on WIN32 Platform

Michal Hušák, Chris Ward*
Lightspeed Design Group, 1611 116th NE Suite 112, Bellevue, WA 98004, U.S.A.

ABSTRACT

We have developed software for flexible and cost effective high-resolution stereoscopic video playback from an off-the-shelf Windows compatible computer. The software utilizes the highly flexible input format created through compatibility with the Microsoft DirectShow standard. Video processing speeds are based on selected compression method usage in combination with hardware accelerated OpenGL data post processing. The key features of the software are: support for multiple input and output formats, on the fly format conversion, up to HDTV (currently 1280x720) per eye resolution, ability to preview data from stereoscopic cameras, and adjustable stereoscopic data corrections.

Keywords: Stereoscopic, video, player, DirectShow, OpenGL

1. INTRODUCTION

The first media used for stereoscopic motion picture storage in history was standard film based material. Today this method is still successfully used in IMAX and other special venue theatres. Analog video playback of electronically recorded stereoscopic data, prior to the computer, was very inconvenient, inflexible, and limited in resolution.

Recent developments in computer technologies, particularly increased CPU speed, improvements in memory access architectures, and graphic card developments have totally changed the situation. On a standard PC it is now possible to not only operate at HDTV (per eye) movie resolutions, but to even make on the fly conversions between formats and apply several stereoscopic data corrections. This article will describe how we are utilizing recent advances in graphic card hardware and software technologies to develop universal stereo-video playback software – DepthQ.

2. VIDEO STORAGE FORMATS

2.1 Stereoscopic image coding

The software is independent of the source data's method of stereo image encoding. We can distinguish between several stereoscopic source movie formats. Some of the formats are created by the stereo video recording device used: interlaced format (dual camera with multiplexor, camera with NuView attachment), beam splitter or Stereo Imaging System format. Other common formats are the result of source data reprocessing: above/below format, above/below format optimized for synchronization doubling, left/right format. There exist only a very few norms for stereo-video data standardization. The following articles describe some basic standards for interlaced stereo-video coding and progressive NTSC stereo video coding [1,2].

Another possibility is to store stereo-video data in separate left and right data or movie files. This type of separated storage makes poor sense and creates unnecessary difficulties with data access and decoding synchronization. A separated method is not the best method going forward in a digital architecture.

From the view of very fast hardware accelerated image-processing requests, the most appropriate format for stereo-video playback is a format with clearly separated left and right images. The Interlaced format does not fit this category – almost all operations within a PC require separation into left and right image from the individual lines. Another suitable feature for fast processing: one eye data exactly in one pixel block, no rotation or translation required, full resolution per eye. Only the above/below format with double height and full per eye resolution satisfies this needs. From the coding point of view this is the most suitable format for stereo-video data storage.

Our DepthQ code can handle all of the mentioned formats in both above-below and right-left eye variants. Only one format has a slight negative influence on performance, the interlaced format, as it requires line reordering. We can reprocess all the other formats to any output form on the fly in de-facto zero time by methods described in Section 4. See fig. 1. for supported input format overview. We did not in DepthQ give any format a priority as a "standard", but the format most often used by us is above/below with full per eye resolution and left eye image encoded in the top area.

2.2 Compression method

For stereo-video playback we can use compressed or un-compressed movie data. The selection of the best compression solution to use depends on: the required output image quality, output resolution, speed of playback, and available storage space. Lets discuss the maximum movie quality and resolution as the main criteria. We have in principle two key bottlenecks that influence the compression method used:

1) CPU speed available for the decompression operation and its related memory access speed.

2) Maximum data flow from a storage mechanism (usually single hard drive or disk array) to the PC memory.

We do not reference a stereoscopically optimized compression algorithm, because no such compression algorithm is available on the market in a standardized form. The common existing 2D video targeted algorithms can be used effectively. These compression algorithm's can be generally ordered according to their respective CPU usage requirements in the following order (from highest CPU usage):

A) DivX, XVID, Microsoft VM9
B) MPEG2, MPEG1
C) DV, MJPEG
D) Proprietary speed optimized algorithms
E) Non-compressed data

The order of the algorithms as regards total data flow requirements from the hard drive (bottleneck 2) is in exactly the opposite order.

DepthQ can use almost all common compression algorithms. The access to these algorithms is granted through Microsoft DirectShow API. DirectShow also gives backward compatibility with the Video for Windows standard as well. It depends on the user's requirements which format should be used. The proprietary speed optimized algorithm used by DepthQ for the best high resolution playback is balanced between the CPU load and disk data flow bottleneck to give the best result (but with very large movie files). For small movie files at relative acceptable resolution (full NTSC or PAL per eye) mpeg2 or DivX can do the job.

2.3 Stereoscopic DVD format

A specific category of input is the stereoscopic DVD. Such DVDs are nowadays produced especially by the Slingshot company as a conversion of IMAX movies. The stereoscopic information on such media is encoded in the form of a interlaced MPEG2 stream to be compatible with direct playback on TV. Specific characteristic of such data are encoding of the MPEG2 stream inside the DVD format framework, support for multiple titles and eventually CSS algorithm based protection against data manipulation.

We have solved direct playback of stereoscopic DVD in DepthQ by making it compatible with existing PC DVD software decoders. DepthQ can internally use third-party decoder for the menu navigation, CSS descrambling and other DVD related issues. DepthQ takes care about conversion of the interlaced data into any supported output format. However, although the technical requirements for DVD decoding are clear, licensing issues for a decoder with the exact requirements for 3D DVDs are ongoing, as the market remains small for 3D DVD playback.

2.4 Still image formats

The most common format for still images storage is the JPS format suggested by Canopus company. It is a standard JPG format with stereo image encoded in the form of cross eye orientation. In principle it is possible to use for stereoscopic image encoding the same formats as for the movies and any commons still image compression.

Microsoft DirectShow still image engine can handle JPG, TGA, GIF, and BMP images. DepthQ can utilize the Microsoft supported decompression for stereoscopic image visualization including programmable slide show (JPS could be handled as JPG data). It is possible to use third-party still image DirectShow component as well. The LEAD Technologies components was tested for JPEG 2000, PNG and other still image format handling. The most promising format for the future is probably the usage of JPS standard in combination with JPEG 2000 compression.

3. OUTPUT FORMATS

3.1 Page flipping

The page flipped format for stereoscopic video output is used in combination with CRT monitors or specialized DLP projectors and shutter glasses. The support of this output depends very much on the graphic card used. In principle there exist 2 APIs which can produce page flipping output: OpenGL and DirectDraw (as a sub part of Microsoft DirectX). The future of standardized page flipping support under DirectDraw is not clear. Microsoft has removed the stereoscopic specification from their DirectX SDK. nVidia partially supports DirectDraw based page flipping in a proprietary limited way (works only in full screen mode).

We have decided to support the hardware page flipping output in DepthQ at this time as it is the most standardized and well tested way – in the form of OpenGL support. OpenGL page flipping works both in full-screen mode and in in-window mode. It is supported not only on nVida graphic cards (Quadro series), but on other professional graphic cards from ATI and 3D Labs production as well.

3.2 Dual output for passive projection

The passive polarized projection require output for 2 projection devices. Such output could be created by hardware reprocessing of page flipped output (by CyViz like devices) or by the help of dual output graphic cards. On dual output graphic cards the redirection could be achieved in two ways: utilizing the standard dual screen desktop geometry (supported by most graphic card drivers) or using a special method for sending video data to only one output connector.

In DepthQ we support both methods. DepthQ can create on the output left right coded stereo image or above/below images. After zooming such images to full dual desktop size it is possible to obtain on each graphic card connector only the proper eye signal. The OpenGL method for creation of the output images eliminates problems with overlay surfaces usage (works only for one monitor) or RGB DirectDraw surfaces usage (performance trouble). This modes is e.g. useful in combination with Wildcat graphic cards.

Other possibility for dual output creation is the utilization of nVidia Quadro graphic card special features. The dual output Quadro cards can optionally redirect page flipping mode into different output in a special stereoscopic cloned mode. On the latest Quadro graphic cards (2000 FX and 3000 FX) the output in this mode is even perfectly genlocked.

In addition to standard redirection to 2 outputs DepthQ supports a special redirection mode optimized for projection from LCD projectors. This mode is based on idea described in following article[3]. The green components from each eye picture are swapped making utilization of light pre-polarization in LCD projectors possible.

3.3 Other supported output formats

There exist several other format suitable for stereoscopic visualization or projection: anaglyphs, interlaced pattern for line blanking, above/below for synchronization doubling, input for Stereo Imaging Systems projector attachment, vertical interlaced pattern for auto-stereoscopic monitors. DepthQ can create all of them on the fly in hardware accelerated way. See fig. 2 for details. Conversion into any new format, including e.g. very complicated color pattern for auto-stereoscopic monitors on the fly could be easy implemented on request.

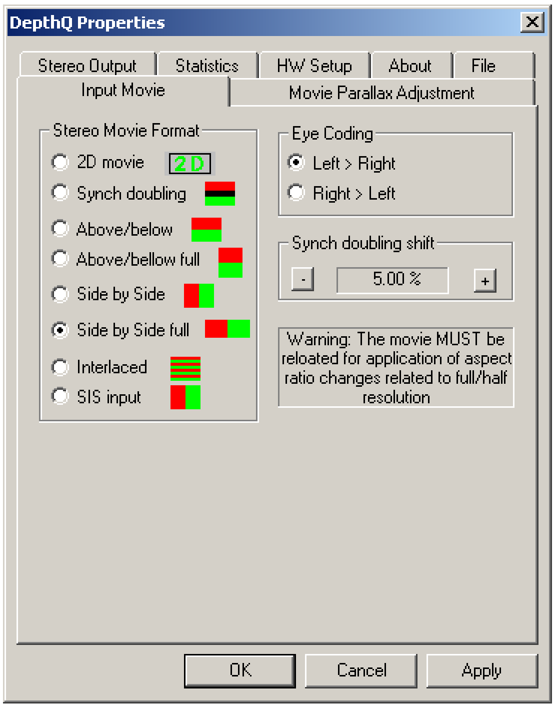

Figure 1. DepthQ Input Movie control panel demonstrating supported input formats.

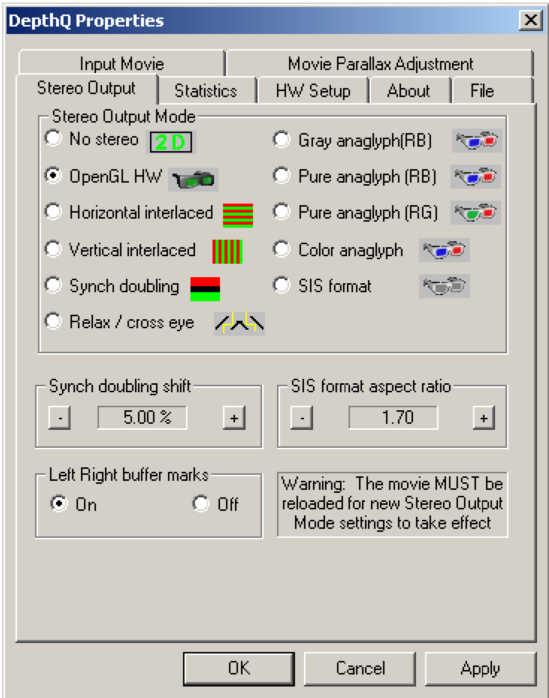

Figure 2. DepthQ Stereo Output Mode control panel demonstrating supported input formats.

4. REQUEST FOR HI RESOLUTION AND HW RELATED ISSUS

One of the most important requirement for the hi quality stereoscopic playback is support for as high-resolution as possible in combination with on the fly data reprocessing. The required current target per eye resolution which should be handled from one PC is determined by the best digital projection device available on the market – it is approximately 1920x1080 requiring 1920x2160 above/below coded stereoscopic movie. Such resolution require about 400 MB/s data flow for non compress pixels in RGBA format at 24 fps. Under the assumption that acceptable compression is 1:5 we need 80MB/s source data flow. Data processing could be in additionally complicated by image post processing on the fly. Let us have a look on the related performance problems.

4.1 CPU related issues

A very interesting information about CPU usage for video data processing could be found in[4] . The article discuss the speed of a typical video decompression operation – expansion of YV12 coded data into an YUY2 coded image. The key results mentioned in the article are summarized in fig. 3.

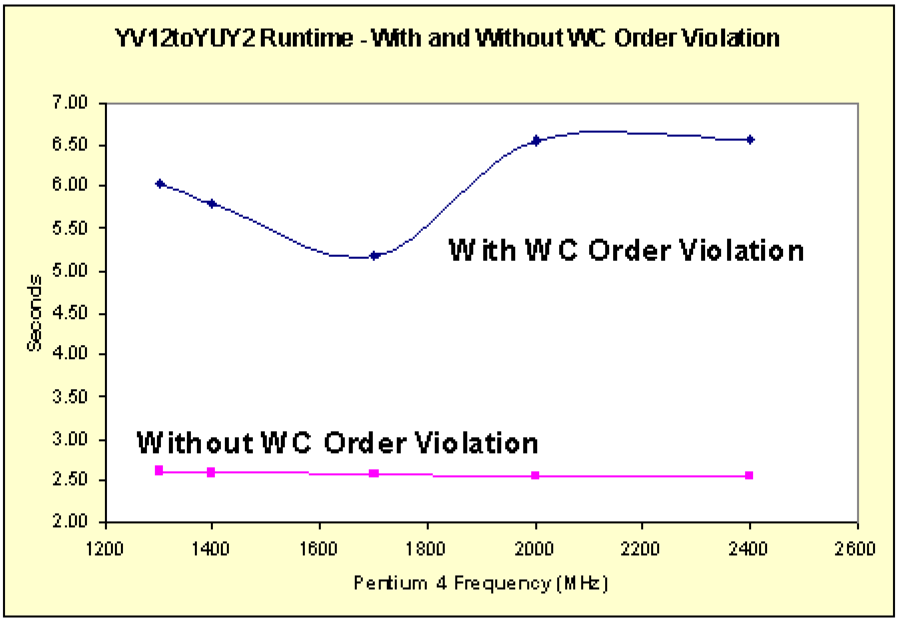

Figure 3. Speed dependence of a YV12->YUY2 routine in dependence on memory access method and CPU frequency.

The figure gives 2 key pieces of information. The speed of typical video processing operation does not greatly depend on the CPU frequency. Algorithms operating in the same time only with one memory block and optimized for SSE2 instructions could be significantly faster than other algorithm (the Without Write-Combining Violation method). The conclusion is: CPU speed improvement without other technologic improvement cannot help us at all, because the CPU will be during video processing doing nothing but waiting for data. Another conclusion: decompression algorithms optimized for latest non-standard instructions (MMX,SSE,SSE2) are absolutely necessary.

4.2 Disk access related issues

Even for compressed data we need a very hi bandwidth data flow close to 80MBs. The speed of buffered read from a normal IDE disk is about 30MB/s. So the problem could be with some reserve solved with 4 IDE disk based disk array and changed to memory data transfer problem. For low bandwidth data a simple 2 IDE disk array gives enough power with reserve.

4.3 Memory access related issues

The key bottleneck for video processing on PC is not CPU speed or graphic card speed nor disk array speed. The bottleneck is the speed of the CPU and graphic card communications with memory. This speed is influenced by following factors:

1) Used memory architecture

2) Front buss (FSB) frequency

3) AGP communication with graphic card

The speed of AGP 4x bus looks sufficient and AGP8x bus usage gives only negligible improvement. On the other side our experiments have shown that the speed of video data processing is almost linearly dependent on the FSB and the memory architecture frequency. The best result could be obtained from motherboard using 800MHz FSB in combination with dual channel DDR memories working on 800 MHz or 533MHz FSB working with RDRAM at 1066 MHz. With speed/disk data flow balanced algorithms we were able to handle up to 1280x720 per eye resolution (movie encoded at 30 fps). Full PAL or NTSC per eye resolution in mpeg2 compression could be handled by PC with 533MHz FSB and 800 MHz RDRAM or comparable fast DDR equivalents.

4.4 Graphic card related issues

It is clear that we need a hardware-accelerated method for all the operation with the decompressed video data. Under WIN32 platform we have following possibilities: DirectDraw surfaces, DirectDraw Overlay surfaces, Direct3D textures, OpenGL textures. DirectDraw surfaces operations are slow. Overlay surfaces operations are fast, but they are non-functional for dual output. Neither DirectDraw surfaces nor Overlay surfaces support acceleration for complicated operations (e.g. image rotation). Direct3D textures support complicated operation, but they do not support in window HW page flipping. DepthQ use for all operations OpenGL textures.

The memory access bottleneck indicates the following “way to speed”: to transfer the decompressed video image inside graphic card video memory as fast as possible, to do as much as possible operations in video memory by the hardware accelerated commands. We can speed up the transfer to video memory in several ways:

1) Reduce the data transfer amount by transferring partially compressed data

2) Use special HW accelerated methods of data transfer

We can reduce the amount of data transfer from video de-compressor to graphic card by transferring in compressed color space. Such transfer usually does not have influence on data quality, because the compression algorithm uses reduced color space anyway. The examples of such reduced format are YV12 or YUY2. Expansion to the output RGB format could be done by graphic card HW. DepthQ can operate now with YUY2 data in HW accelerated way.

Standard OpenGL implementation is not very well optimized for pixel data transfer. From technical reason OpenGL standard trades stability to speed in this direction. This field is open for non-standard OpenGL feature usage.

Once we have the image data in graphic card video memory as a texture it is possible to do any operation at a speed significantly higher in comparison to equivalent operation done by CPU. In addition all of this operation are done by GPU in parallel to the CPU work de-facto in zero time. We use OpenGL texture operation in DepthQ e.g. for following tasks: Left-Right eye picture separation from the source image, image stretching and rotation, image shifting for vertical and horizontal parallax correction on the fly, aspect ratio related operations, color manipulation operations e.t.c. ….

5. DIRECT PREVIEW FROM CAMERA

The Microsoft DirectShow API offers hi level of abstractions from the source movie data. A code written in a way respecting all DirectShow standards could be as a result used not only for recorded movie processing but it can be combined with a real time data source as well – stereoscopic video camera or cameras.

5.1 single camera signal processing

The most easy task is connection to one camera. This camera signal could be both recorded on the fly and converted to all supported stereoscopic output format for preview purpose. We had in practice tested such solutions for a DV camera with NuView attachment and also for a camera with a beam splitter attachment. On the fly preview of stereoscopic data during recording is very useful for the camera alignment and stereoscopic scene arrangement checking.

5.2 dual camera data processing

This is a bit more complicated problem. It was necessary to develop an additional DirectShow component (DepthQ Video Mixer) to make on the fly preview from 2 camera possible. The Mixer takes frames from 2 sources (2 video camera or 2 movie files) and combines them into above/below format. The problem with input data synchronization is solved by combining together frames “most close in the time”. The performance problems related to communication with dual camera and data recompression makes recording in such mode possible only for limited resolutions. Preview only works O.K. for full PAL/NTSC per eye resolutions. We had tested in practice such setup for dual DV camera , dual Orange Micro USB2 camera as well as for several USB1 Web cam.

6. DEPTHQ COMPONENTS

DepthQ stereoscopic playback system consist now from 4 key software components. The main component is a DepthQ DirectShow stereoscopic video rendering COM object. This component is written in combination of C++ and assembler. DepthQ Video Renderer takes care about all the HW accelerated stereoscopic video operations.

The second main component is a movie playback user interface – DepthQ Player. DepthQ Player is written in Delphi. It takes care about DirectShow object management , movie playback control (including remote control) and user interactions. Some of the GUI interface futures: output aspect ratio control, support for play list, adjustable output area adjustment, remote control features (serial, TCP/IP, IR), possible GUI customization and skinning …

Third component is a camera communication code – DepthQ Cap cooperating with the DepthQ Video Renderer. This code makes on the fly stereoscopic preview and recording from 1 camera with DirectShow compatible drivers possible.

The last component is DepthQ Video Mixer – DirectShow COM object taking care about mixing data from 2 video sources.

CONCLUSIONS

With the combined power of the modern CPU and GPU based graphic cards it is possible to play from an off-the-shelf PC, stereoscopic video data at HDTV resolutions. These high-resolution video streams may then be manipulated in real-time, creating completely new creative and technical opportunities never before possible within the standard analog or mainstream digital video playback paradigms. For the future the key hardware problem is faster memory architecture, which directly influences all video operations. The key software task for the future is to maximally redirect the video operation from CPU and standard memory to GPU and video memory.

REFERENCES

1. A. Woods , T. Docherty, R. Koch, “3D Video Standards Conversion”, Stereoscopic Display and Virtual Reality Systems VII, Proc. of SPIE Vol. 2653A, pp. 210-218, 1996
2. J. Goodman, “Development of the 960p stereoscopic video format”, Stereoscopic Display and Virtual Reality Systems X , Proc. of SPIE Vol. 5006, pp 187-194, 2003
3. V. A. Elkhov, Y.N. Ovechkis, “Light loss reduction of LCD polarized stereoscopic projection”, Stereoscopic Display and Virtual Reality Systems X , Proc. of SPIE Vol. 5006, pp. 45-48, 2003
4. E. L. Palmer, “How to get faster video rendering on the Intel Pentium 4 processor”, Intel Developer Services www, Intel Corporation, 2002

* chris.ward@lightspeedDesign.com; phone: 425.637.2818 ; fax: 425.696.0115 ; www.LightspeedDesign.com