A New Generation Image Processing Algorithm Might Enable Real-Time Visualization of 3D Echocardiographic Data on a Multi-View Auto Stereoscopic LCD

G Saracino, N Greenberg, S Fukuda, T Shiota, JD Thomas

Cleveland Clinic Foundation, Cleveland, OH, USA

Abstract

Visualization of Real-Time 3D Echocardiography (RT3DE) is currently limited by the bi-dimensional nature of conventional monitors. The objective of this study is to implement and evaluate an optimized image-processing algorithm to generate multi-view auto-stereoscopic images. Custom-made software rendered test images (n=5) on a 9-view auto-stereoscopic 3D LCD. We designed and compared two algorithms that used hardware acceleration of graphic adapters. The quality of the final 3D image with the two algorithms was comparable. This study showed that the new-generation algorithm is very efficient in processing complex rendering such as visualization of RT3DE data.

This algorithm might be an enabling factor for real-time visualization of RT3DE on multi-view auto stereoscopic display, which might improve clinical imaging in a host of diseases and use of RT3DE in guidance of interventional procedures.

1. Introduction

Currently available Real-time 3D echocardiography (RT3DE) is strongly limited by the bi-dimensional nature of media (paper, PC monitor) used to visualize 3D datasets. Indeed, to visualize a 3D dataset, one must reformat data to multiplane slices or utilize perspective projections where depth perception is missing [1].

Auto-stereoscopic LCD monitors enable visualization of multiple views of 3D images while preserving depth perception without the need for wearing special glasses[2-3].

Computational power required for generating multi-view images is a limiting factor for online visualization of RT3DE [4].

Last year we reported the feasibility of using auto-stereoscopic LCD to visualize multiple views of RT3DE data while preserving depth perception and our obtained results suggested an improved algorithm for the image processing. The objective of this study is to implement and evaluate a newly designed image-processing algorithm capable of taking full advantage of computational power of commercially available graphic adapters to generate multi-view auto-stereoscopic images.

2. Methods

2.1. Multi-view auto-stereoscopic 3D LCD

In this study we used a 9-view auto-stereoscopic Active Matrix Liquid Crystal Display (LCD) with a native resolution of 1600x1200 pixels.

An auto-stereoscopic display is a special device capable of deploying two different images to each eye of one observer without the use of special glasses.

The separation of images occurs using special lenses that are able to converge specific parts of the screen into distinct focal points. When each eye of the observer is spatially located into two different focal points, the observer experiences a stereoscopic view.

Stereoscopic views can be used to create the perception of depth in images, by feeding each eye with a different perspective view of a scene.

A multi-view auto-stereoscopic display is an even more advanced display capable of rendering several images in as many focal points. This technology can provide the user, who changes his/her vantage point, with a natural change of perspective along with the perception of depth.

In a multi-view auto-stereoscopic display the full resolution of the display underlying the lens has to be partitioned and allocated to each view.

Advances in the technological processes currently used in the construction of Active Matrix LCD enabled the production of high-resolution, very bright displays that are an ideal base for auto-stereoscopic displays.

The full resolution (1600x1200) of the LCD used in this study is partitioned by the lens into nine (9) views each with a resolution of 533x400.

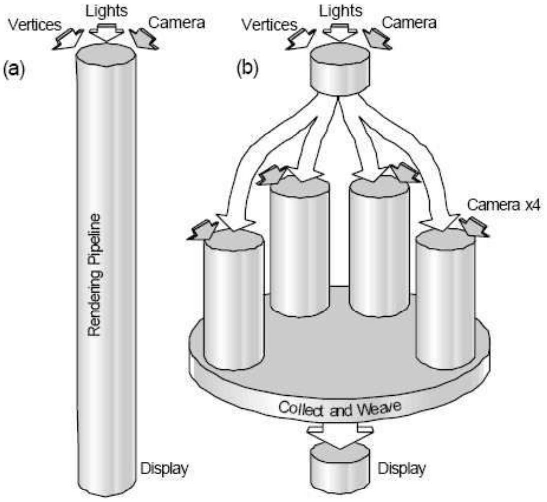

Figure 1 Comparison between conventional image generation (a) and 3D image generation (b). The OpenGL rendering pipeline is repeated for each view. Images are then weaved and shown to the user

In LCD screens, each pixel is divided into 3 rectangular sub-pixels: one for each color component. The separation into views occurs at a sub-pixel level. This means that the adjacent green, red, blue sub-pixels of one pixel go to different views. For each sub-pixel, the view is determined by the position of the pixel and the construction parameters of the lens. Given the indexes k,l that point to an individual red, green, blue sub-pixel, the following formula can be used to determine the view number N:

$$N = \frac{(k + k_{offset} - 3l \tan\alpha) \bmod X}{X} N_{tot}$$

In the set of parameters used in our 3D LCD (Ntot=9, X=4.5, α=9.4623°,K_{offset}=0) the view number N is always an integer.

In the multi-view configuration, each frame of the 3D image contained nine different views of the scene. Each view was created by calling the same OpenGL drawing primitives but with the projection camera set into a slightly different position (**Figure 1**). All of the views where collected and then combined into one image by following an interleaving pattern that is dependent on the pixel-view mapping function described above (**Figure 2**). We refer to this image preparation process as image weaving process.

Image weaving introduces computational overhead that is added to the time required to create each view. In this study we designed two different algorithms that took advantage hardware acceleration available on the Graphic Adapter to weave views.

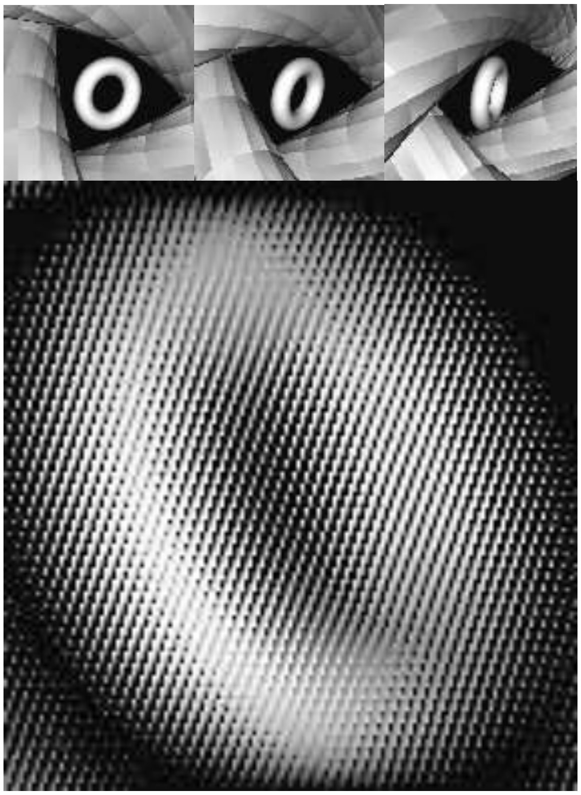

Figure 2 The lower portion shows an example of nine different views weaved into one multi-view image following the interlacing pattern required by our auto-stereoscopic LCD. In the upper part, three views of the same scene before the weaving process.

The first algorithm weaved 9 views at native resolution directly to the frame buffer. This technique accelerated the image generation process using the stencil buffer available since early versions of OpenGL.

In synthesis, each pixel of the stencil buffer is initialized with a status value indicating the view number of each color component. Each time a view is being drawn, an appropriate stencil test controls and allows the actual drawing only in appropriate pixels of the screen. Views are computed at full resolution even if only a small set of pixels will be used in the final rendering. Unused pixels do not generate overhead since they fail the stencil test and geometric calculations are not performed. This approach combines image generation and weaving into one-step process that is entirely performed by the Graphic Processor Unit (GPU).

The second algorithm, instead, exploits previously obtained results. In our previous study, we found that the rendering time per pixel per view was almost invariant when the number of views greater than 3. We hypothesized that the rendering algorithm would be faster if it generated 9 views at a lower resolution and than weaved them at the native resolution using the GPU. In the second algorithm image generation and weaving occur without the need to transfer partial result from video memory into system memory. To ensure that lowering the resolution of the views did not negatively impact the overall quality of final image partial image rendering was performed at a resolution slightly higher that the resolution allocated to each view (512x512 pixels vs. 533x400 pixels).

2.2. Test image visualization

To assess performance of the two multi-view rendering algorithms, we measured the frame rate achieved when a commercially available workstation (Dual Xeon 3.6 GHz, Hyper-threading enabled, 2 Gb Ram) rendered five test images (n=5) on our 9-view auto-stereoscopic LCD (Philips, Eindhoven, Holland). The resolution of the resulting multi-view image was set to 1600x1200 pixels but each test image used a different degree of complexity that is quantified by the number of triangles rendered and the percent of screen that it covered (**Table 1**).

Table 1 Test images had 3D scene with increasing complexity quantified by the number of triangles rendered and the percent of total pixels utilized on the screen.

	Triangles rendered	Screen covered (Percent)
Test Image 1	250.000	25
Test Image 2	400.000	41
Test Image 3	550.000	62
Test Image 4	700.000	67
Test Image 5	950.000	83

3. Results

Using our first algorithm, the frame rate of test images ranged from 36 to 127 fps and the average frame rate was 63 fps. Using our newly designed algorithm the rendering speed ranged from 80 to 89 fps and the average was 84 fps (**Figure 3**).

As it was expected, the frame rate decreased with increasing complexity of the scene rendered. When the complexity of the scene rendered was sufficiently high our new algorithm was faster than the previous generation (Test images 2 through 5, 47.25 fps vs. 82.75 fps, p <0.01) (**Figure 4**).

In the test image with lowest complexity the first algorithm was faster than the second (127 fps vs 89 fps).

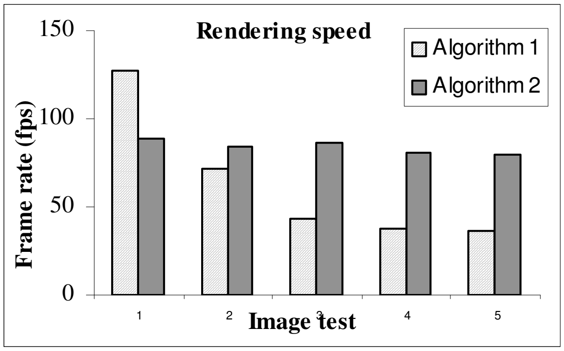

Figure 3 The obtained rendering speed of the two algorithms for each test image.

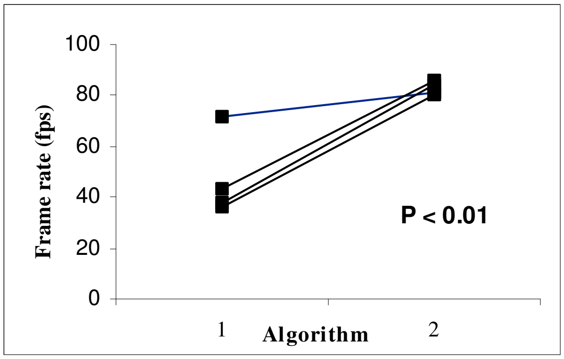

Figure 4 Comparison of the rendering speed in the test images in test image from 2 through 5

4. Discussion and conclusions

This study attempted to implement and evaluate an optimized algorithm for novel visualization of multi-view images. We showed that when rendering complex images on multi-view auto-stereoscopic LCD, it is faster to create and collect lower-resolution view and weave them into a full-resolution image rather than to weave 9 full-resolution views directly into frame buffer. The original contribution lays partly in having identified a method to use hardware acceleration available to perform the image weaving process while maintaining partial results entirely into video memory.

Our methods relieve the stress put on the memory-bus by data transfer of partial results back and forth between Video Memory and System Memory. This resulted in higher rendering speed not only because the main processor could operate on system memory while the Graphic Processor was operating on video memory, but also because we could take advantage of the greater memory bandwidth available on the video memory. In fact, video memory in our system operated faster than system memory (DDR3 at 1000 MHz vs. DDR2 at 800

MHz).

In addition, we further enhanced the design of our algorithm by using the consideration, made in our previous study, that processing time per pixel per view was almost constant when number of views generated were greater than three. We reduced the number of pixels processes by lowering the resolution of partial views, but to preserve the quality of final image we kept the resolution of each rendered view higher than the resolution allocated on the LCD panel to one view (512x512 pixels vs. 533x400 pixels). Our results showed that this expedient was particularly advantageous when rendering complex scenes that used a large portion of pixels of the screen.

On the other hand, it did not improve the performance when rendering simplest test scene. The phenomena can be easily explained considering that the second algorithm has to transfer and process pixels of partial view even if they were merely blank. The first algorithm, instead, did not per perform any operation at all for blank pixels.

The new algorithm exploits a more efficient use of the geometric pipeline but overall performs better if the gain in the geometric pipeline of the graphic card justifies the constant overhead of weaving all the pixels of views.

That leads to the discussion of which algorithm to choose in a given situation. Results from this study suggest employing the first algorithm if the user has to draw a scene made of lines, dots or small particles, and the second algorithm if the scene is more complex and uses a large portion of the screen. It is important to note that even in the worst case scenario the second algorithm was able to achieve a high frame rate (89 fps) and its performance stayed balanced under load (> 80 fps).

The results suggest that new generation algorithm is well balanced and better suited for complex rendering of data from Real Time 3D Echocardiography.

Our new generation algorithm promises to be an enabling factor to achieve the online visualization of RT3DE on 3DLCD in real-time and to unlock the full potential of this novel display in the clinical environment. The availability of online 3D visualization techniques could positively impact effectiveness of 3D imaging modalities in guiding interventional procedures such as pericardial effusion, percutaneous mitral valve repair, and transseptal puncture. Further investigation is needed.

Acknowledgements

Support for this study is partially provided by a grant funded by Philips 3D Solution, Eindhoven.

References

[1] G Saracino, NL Greenberg, T Shiota, C Corsi, C Lamberti, JD Thomas. Fast interactive real-time volume rendering of real-time three-dimensional echocardiography: an implementation for low-end computers. Computers in Cardiology 2002;29:613-616.

[2] C van Berkel, D W Parker and A R Franklin. Multiview 3D-LCD. Proc SPIE 1996;2653:32-39.

[3] S Ichinose, N Testutani & M Ishibashi. Full-Color Stereoscopic Video Pickup and Display Technique without Special Glasses. SID 1989;89:188-191.

[4] N Davies, M McCormick and L Yang. Three-dimensional imaging systems: a new development. Appl Optics 1988;27(21):4520-4528.

[5] D E Sheat, G R Chamberlin, P Gentry, J S Leggatt and D J McCartney. 3-D Imaging Systems for Telecommunications Applications. Proc SPIE 1992;1669:186-192.

[6] A Woods, T Docherty and R Koch. Image Distortions in Stereoscopic Video Systems. Proc SPIE 1993;1915:36-38.

[7] G S B Street. Method and apparatus for use in producing autostereoscopic images; Eur Pat 1983.

[8] Y Yeh and L D Silverstein. Limits of Fusion and Depth Judgement in Stereoscopic Colour Displays. Human Factors 1990;32(1):45-60.

[9] C van Berkel, A R Franklin and J R Mansell. Design and Applications of Multiview 3D-LCD. Proc SID Euro Display 1996;96:109-112.

[10] W L Martens. Physiological approach to optimal stereographic game programming: a technical guide. Proc SPIE;2653:261-270.

Address for correspondence

James D. Thomas, MD
Department of Cardiovascular Medicine
Cleveland Clinic Foundation
Desk F15
9500 Euclid Avenue
Cleveland, OH 44195