

Integrating Autostereoscopic Multi-View Lenticular Displays in Minimally Invasive Angiography

Daniel Ruijters

X-Ray Predevelopment, Philips Healthcare, Best, the Netherlands
danny.ruijters@philips.com.

Abstract. In order to gain maximal insight in the vascular morphology during minimal invasive treatment of vascular pathologies, high resolution 3D Rotational Angiography (3DRA) data is acquired and visualized peri-operatively. The 3D data provides good understanding of the vasculature, however in case of complex structures (*e.g.* liver and neurovasculature, and arteriovenous malformations) there is a distinct added value in showing a stereoscopic visualization to the clinician. In order to provide such a stereoscopic image, a 42" autostereoscopic nine-view lenticular screen was mounted in the Operating Room (OR). To optimally assist the intervention, the stereoscopic visualization of the vasculature can be presented from the same viewing incidence as the real-time X-ray image, which is used to guide the intra-vascular devices, such as catheters and stents. The GPU-based implementation offers interactive manipulation throughout.

1 Introduction

Autostereoscopic displays allow a stereoscopic view of a 3D scene, without the use of any external aid, such as goggles. The additional depth impression that a stereoscopic image offers, allows a natural interpretation of 3D data. Principally there are two methods for conveying a stereoscopic image: time multiplexing and spatial multiplexing of two or more views. Though two views are enough to create the impression of depth (after all, we have only two eyes), offering more views has the advantage that the viewer is not restricted to a fixed sweet spot, since there is a range of positions where the viewer will be presented with a stereoscopic visualization. As a consequence, multiple viewers can look at the same stereoscopic screen, without wearing goggles. Furthermore it is possible to 'look around' an object, when moving within the stereoscopic range, which aids the depth perception. The multi-view lenticular display uses a sheet of lenses to spatially multiplex the views [1], and typically offers four to fifteen spatially sequential images.

The Graphics Processing Unit (GPU) is a powerful parallel processor on today's off-the-shelf graphics cards. It is especially capable in performing Single Instruction Multiple Data (SIMD) operations on large amounts of data. In this

article a method for rendering a 3D scene for display on lenticular screens is described, benefitting from the vast processing power of modern graphics hardware. The presented approach allows dynamical adjustment of the resolution of the rendered images, in order to guarantee a minimal frame rate. In this way interactive frame rates can be reached, enabling to fully profit from lenticular displays in a clinical environment.

2 State of the art

In 1838 Sir Charles Wheatstone developed a device, called the stereoscope, which allowed the left and the right eye to be presented with a different image, in order to create a impression of depth. The development of *autostereoscopic* display devices, presenting stereoscopic images without the use of glasses, goggles or other viewing aids has seen an increasing interest since the 1990s [2–4].

The advancement of large high resolution LCD grids, with sufficient brightness and contrast, has brought high quality multi-view autostereoscopic lenticular displays within reach [5]. The range of viewing positions, allowing the perception of a stereoscopic image, is mainly determined by the number of views, offered by the display.

Several publications describe how the GPU can be employed to extract the data stream for the lenticular display from a 3D scene in an effective manner. Kooima *et al.* [6] and Kratz *et al.* [7] present a two-pass GPU based algorithm for a two-view head-tracked stereoscopic display. First the views for the left and the right eye are rendered, and in the subsequent pass they are interweaved. Domonkos *et al.* [8] describe a two-pass approach, dedicated for iso-surface rendering. In the first pass they perform the geometry calculations on the pixel-shader for every individual pixel, and in the second pass the shading is performed. Hübner and Pajarola [9] describe a GPU-based single-pass multi-view volume rendering, varying the direction of the casted rays depending on their location on the lenticular screen.

The previous GPU-based approaches were dedicated render methods, working on the native resolution of the lenticular LCD grid. The approach in this paper decouples the rendering resolution from the native LCD grid resolution, allowing lower resolutions, when higher frame rates are demanded.

3 The multi-view lenticular display

The multi-view lenticular display device consists a sheet of cylindrical lenses (lenticulars) placed on top of an LCD in such a way that the LCD image plane is located at the focal plane of the lenses [10]. Therefore, LCD pixels located at different positions underneath the lenticulars fill the lenses when viewed from different directions, see figure 1. Provided that these pixels are loaded with suitable stereo information, a 3D stereo effect is obtained, in which the left and right

eye see different, but matching information. The screen we used offered nine distinct angular views, but the described method is applicable to any number of views.

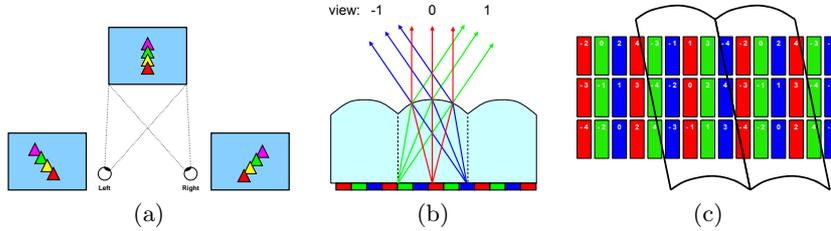


Fig. 1. (a) Stereoscopic image perception. Two different views are combined by the brain, in order to perceive depth. (b) The light of the sub-pixels under the lenticular lenses is directed into different directions [11]. (c) The cylindrical lenses depict every sub-pixel in a different view. The numbers in the sub-pixels indicate in which view they are visible.

The fact that the different LCD pixels are assigned to different views (spatial multiplex), leads to a lower resolution per view than the resolution of the LCD grid [12]. In order to multiplex the views horizontally (our eyes are placed horizontally), the lenticular cylindrical lenses should be placed vertically, parallel to the LCD pixel grid columns. However, this would reduce the resolution only over the horizontal axis, which is undesirable. Therefore, the lens cylinders are slanted at a small angle, distributing the the reduction of resolution over the horizontal and vertical axis [1]. The resulting assignment of a set of LCD pixels, is illustrated in figure 1c. Note that the red, green and blue color channels of a single pixel are depicted in different views. Every view offers a full color image, though.

4 Method

We use a two pass algorithm: First the individual views from the different foci positions are separately rendered to an orthogonal grid. In the second pass, the final output signal has to be resampled from the views to a non-orthogonal grid in the compositing phase (see figure 2). The processing power of the GPU is harvested for both passes. In order to maintain an acceptable frame rate, the resolution of the views can be changed dynamically.

The frustums that result from the different focal spots, are illustrated in figure 3. The viewing directions of the frustums are not parallel to the normal of the screen, except for the center one. Therefore the corresponding frustums are asymmetric [13]. A world coordinate (x, y, z) that is perspectively projected, using

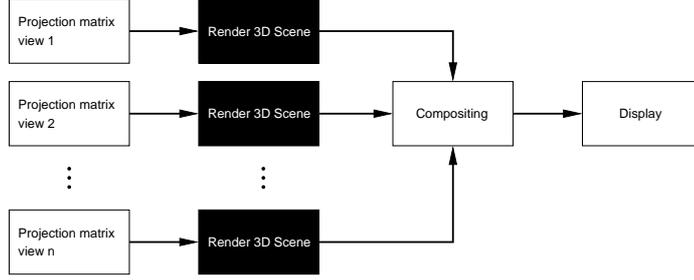


Fig. 2. The process of rendering for the lenticular display. Optionally, the rendering of the N individual views can be done in parallel.

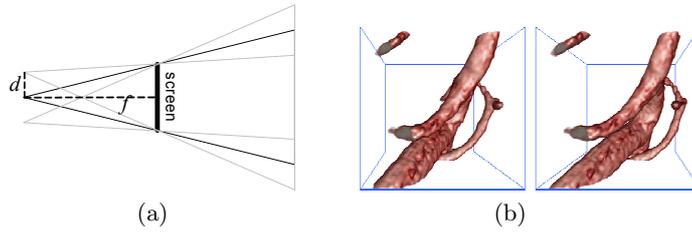


Fig. 3. (a) The frustums resulting from three different view points. (b) The same scene rendered from the extreme left and right viewpoints.

such an asymmetric frustum, leads to the following view port coordinate $v(x, y)$:

$$v(x, y) = \left(\frac{(x - n \cdot d) \cdot f}{f - z} + n \cdot d, \frac{y \cdot f}{f - z} \right) \quad (1)$$

Whereby f denotes the focal distance, n the view number and d the distance between the view cameras.

After the projection matrix has been established, using equation 1, the scene has to be rendered for that particular view. All views are stored in a single texture, which is called *texture1*. In OpenGL, the views can be placed next to each other in horizontal direction, using the `glViewport` command. The location of a pixel in view n in *texture1* can be found as follows:

$$\mathbf{t} = \left(\frac{1}{2} + \frac{n}{N} + \frac{2p_x - 1}{2N}, p_y \right) \quad (2)$$

whereby \mathbf{t} denotes the normalized texture coordinate, \mathbf{p} the normalized pixel coordinate within the view, and N the total number of views. The view index n is here assumed to be in the range $[-\frac{1}{2}(N-1), \frac{1}{2}(N-1)]$, as is used in figure 1c.

To composite the final image, which will be displayed on the lenticular screen, the red, green and blue component of each pixel has to be sampled from a different view. The view number stays fixed all the time for each sub-pixel,

therefore this information is pre-calculated once, and then put in a static texture map, called *texture0*.

In the compositing phase, all the pixels in the output image are parsed by a GPU program. For each normalized pixel coordinate \mathbf{p} in the output image, *texture0* will deliver the view numbers n that have to be sampled for the red, green and blue components. The respective views are then sampled in *texture1* according equation 2 using bi-linear interpolation, delivering the appropriate pixel value. A performance characterization of volume rendering to the lenticular screen according to the presented approach is given in table 1.

Volume description	Frames per second	Views per second
3DRX foot, $256^2 \cdot 200$ voxels (25 MB)	52.4	471
CT head, $512^2 \cdot 256$ voxels (128 MB)	19.2	173
3DRA neuro-vascular (sparse), 512^3 voxels (256 MB)	21.3	192
4D cardiac CT, 3 phases of $512^2 \cdot 333$ voxels (totally 510 MB)	13.6	123

Table 1. In order to characterize the performance of GPU-accelerated volume rendering [14] and compositing on the 9 view lenticular screen, several data sets were rendered at the maximum resolution to an output window of 800^2 pixels. All measurements were obtained, using a 2.33 GHz Pentium 4 system, with 2 GB RAM memory, and a nVidia QuadroFX 3500 with 256 MB on board memory as graphics card.

5 Clinical setup

We applied the presented approach to visualize intra-operatively acquired 3D data sets on a Philips 42" lenticular screen, which was mounted in the operation room (OR). The orientation of the depicted 3D data set can follow in realtime the orientation of an X-ray C-arc system (see figure 4), which means that on the lenticular display the 3D data set is visualized from the same viewing angle, as the viewing incidence on the patient in the real-time X-ray image. This approach allows to reduce the X-ray radiation, since the physician can choose the optimal orientation to acquire X-ray images without actually radiating. Further it improves the interpretation of the live projective 2D X-ray image, which is presented on a separate display, since the 3D data on the stereoscopic screen (which is in the same orientation) gives a proper depth impression through the stereovision of the lenticular screen. The fact that the clinician is not limited to single sweet spot, makes the multi-view display particularly suitable for this environment, since the clinical intervention demands that an operator can be positioned freely in the range close to the patient.

The ability to visualize and manipulate the 3D data interactively is of great importance in the analysis and interpretation of the data. Interactivity, in this



Fig. 4. Left: the 42" lenticular screen (top right display) in the operating room. Middle: the X-ray C-arc system in a clinical intervention. Right: the degrees of rotational freedom of the C-arc geometry.

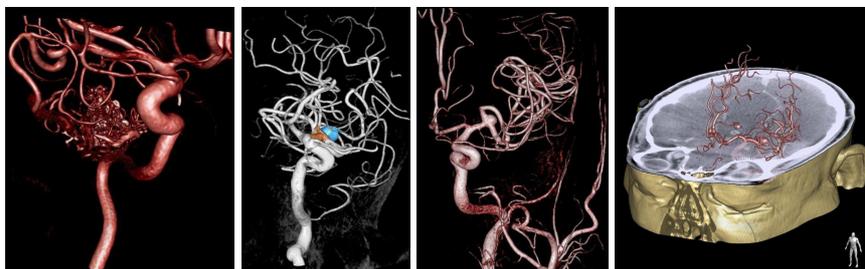


Fig. 5. Volume rendered vascular 3DRA images. From left to right: brain arteriovenous malformations, virtual stenting and aneurysm (blue), another neuro vessel tree with aneurysm, and vasculature mixed with soft-tissue data.

context, means that the frame rates of the visualization are sufficient to provide direct feedback during user manipulation (such as rotating the scene). When the visualization's frame rate is too low manipulation becomes very cumbersome. Five frames per second are often used as a required minimum frame rate.

Especially the cerebral vasculature consists of many curved vessels, see figure 5. From a single X-ray image it is impossible to interpret the in-plane curvature (the curvature in the z -direction of the image). But even looking at a 3D rendered image on a 2D plane (*i.e.* conventional monitor), it is often very difficult to estimate the in-plane curvature without rotating the vessel tree. Rotating a 3D scene is not a problem when sitting behind a desktop computer, but during a clinical intervention the performing clinician is primarily occupied with the medical procedure, and interaction with sterile computer input devices (which are available in the OR) is an additional task, demanding focus. The stereoscopic image allows to interpret the 3D shape, including the in-plane curvature, in a single glance without any additional input interaction, and therefore reduces the mental stress on the clinician during the intervention.

5.1 Follow C-arc

When the Follow C-arc mode is active, the viewing incidence of the 3DRA data set is matched in real-time with the current rotation and angulation of the C-arc geometry (see figure 4). This allows the clinician to use the view on the 3DRA data set to predict an optimal working position for the C-arc without actually using X-ray radiation.

In order to perform this, the transformation of the coordinate space of the 3DRA data to the coordinate space of the C-arc X-ray detector has to be established. Let the origin of the coordinate system of the 3DRA data be positioned at the center of the data set, and let the x -axis correspond to the short side of the table, the y -axis to the long side of the table, and the z -axis point from the floor to the ceiling. The X-ray C-arc system can rotate over three axis (see figure 4b). The rotation of the detector coordinate system, with respect to the 3DRA data can be expressed as:

$$M = R_x \cdot R_y \cdot R_z = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{pmatrix} \cdot \begin{pmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{pmatrix} \cdot \begin{pmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (3)$$

Note that the order of the matrix multiplications is given by the mechanics of the C-arc system. The C-arc system's iso-center serves as origin for the rotation matrices. The rotation of 3DRA volume to the detector coordinate system corresponds to the inverse of matrix M , which is equal to its transposed, since rotation matrices are orthogonal.

The counterpart of the follow C-arc function is called 3D APC (Automatic Positioning Control), which allows the C-arc to be moved to a viewing incidence corresponding to the orientation of the 3D rendering of the 3DRA data set. Desired working positions can be planned and stored before or during the intervention and later recalled during the intervention when needed.

6 Conclusions

In this article an approach was described, bringing stereoscopic visualizations in the operating room, in order to support the clinician during minimal invasive treatment of vascular pathologies using X-ray angiography. We used a 42" nine-view lenticular screen to offer an image that could be easily interpreted by the clinicians. The orientation of the 3DRA data on the lenticular display followed the viewing incidence of the X-ray C-arm in real-time, in order to provide maximum insight in the anatomy that is depicted on a corresponding X-ray image.

Due to the GPU-acceleration, together with the adaptive adjustment of the intermediate view resolution, interactive frame rates can be reached, which allows intuitive manipulation of the rendered scene. Since both the 3D rendering and

the compositing take place on the graphics hardware, the requirements for the other components of the PC system are rather modest. Thus the realization of the proposed high performance system can be very cost effective.

First feedback from the clinicians indicates that there certainly is a perspective for clinical added value. Orthopedic surgery is suggested as another application area that could benefit from the multi-view stereoscopic display. For better integrated usage, the display should be mounted on the same ceiling suspension with the other (2D) displays. Also the integration of the live fluoroscopy image in the 3D scene would be highly appreciated. The fact that the clinicians do not need to wear any additional glasses, and are not limited to a sweet spot, as well as the fact that large data sets can be manipulated interactively, make this method very suitable for a clinical interventional environment.

References

1. van Berkel, C., Parker, D.W., Franklin, A.R.: Multiview 3D LCD. In: Proc. SPIE - Stereoscopic Displays and Virtual Reality Systems III. Volume 2653. (April 1996) 32–39
2. Halle, M.: Autostereoscopic displays and computer graphics. *ACM Computer Graphics* **31**(2) (May 1997) 58–62
3. Dodgson, N.A.: Autostereoscopic 3D displays. *Computer* **38**(8) (2005) 31–36
4. Onural, L., Sikora, T., Ostermann, J., Smolic, A., Civanlar, M.R., Watson, J.: An assessment of 3DTV technologies. In: Proc. NAB Broadcast Engineering. (2006) 456–467
5. Vetro, A., Matusik, W., Pfister, H., Xin, J.: Coding approaches for end-to-end 3D TV systems. In: Picture Coding Symposium (PCS). (December 2004)
6. Kooima, R.L., Peterka, T., Girado, J.L., Ge, J., Sandin, D.J., DeFanti, T.A.: A GPU sub-pixel algorithm for autostereoscopic virtual reality. In: IEEE Virtual Reality Conference. (Mar 2007) 131–137
7. Kratz, A., Hadwiger, M., Fuhrmann, A., Splechna, R., Bühler, K.: GPU-based high-quality volume rendering for virtual environments. In: Proc. AMI-ARCS. (Sep 2006)
8. Domonkos, B., Egri, A., Fóris, T., Juhász, T., Szirmay-Kalos, L.: Isosurface ray-casting for autostereoscopic displays. In: Proc. WSCG. (January 2007) 31–38
9. Hübner, T., Pajarola, R.: Single-pass multi-view volume rendering. In: Proc. IADIS International Computer Graphics and Visualization. (July 2007)
10. van Berkel, C.: Image preparation for 3D-LCD. In: Proc. SPIE - Stereoscopic Displays and Virtual Reality Systems VI. Volume 3639. (May 1999) 84–91
11. Braspenning, R., Brouwer, E., de Haan, G.: Visual quality assessment of lenticular based 3D displays. In: Proc. 13th European Signal Processing Conference (EUSIPCO). (September 2005)
12. Dodgson, N.A.: Autostereo displays: 3D without glasses. In: EID: Electronic Information Displays. (1997)
13. Maupu, D., Horn, M.H.V., Weeks, S., Bullit, E.: 3D stereo interactive medical visualization. *IEEE Computer Graphics and Applications* **25**(5) (Sept.-Oct. 2005) 67–71
14. Ruijters, D., Vilanova, A.: Optimizing GPU volume rendering. *Journal of WSCG* **14**(1-3) (January 2006) 9–16