SCI INSTITUTE
TECHNICAL REPORT

Depth of Field Effects for Interactive Direct Volume Rendering

Mathias Schott[*,†], *Pascal Grosset*[*,†], *Tobias Martin*[†], *Vincent Pegoraro*[‡], *Charles Hansen*[*,†]
[*]SCI Institute, University of Utah, USA
[†]School of Computing, University of Utah, USA
[‡]Saarland University, Germany

UUSCI-2010-005

Scientific Computing and Imaging Institute
University of Utah
Salt Lake City, UT 84112 USA

December 7, 2010

Abstract:

In this paper, a method for interactive direct volume rendering is proposed for computing depth of field effects, which previously were shown to aid observers in depth and size perception of synthetically generated images. The presented technique extends those benefits to volume rendering visualizations of 3D scalar fields from CT/MRI scanners or numerical simulations. It is based on incremental filtering and as such does not depend on any precomputation, thus allowing interactive explorations of volumetric data sets via on-the-fly editing of the shading model parameters or (multi-dimensional) transfer functions.

Depth of Field Effects for Interactive Direct Volume Rendering

Mathias Schott[1,2], Pascal Grosset[1,2], Tobias Martin[2], Vincent Pegoraro[3], Charles Hansen[1,2]

[1]SCI Institute, University of Utah, USA, [2]School of Computing, University of Utah, USA,
[3]Saarland University, Germany

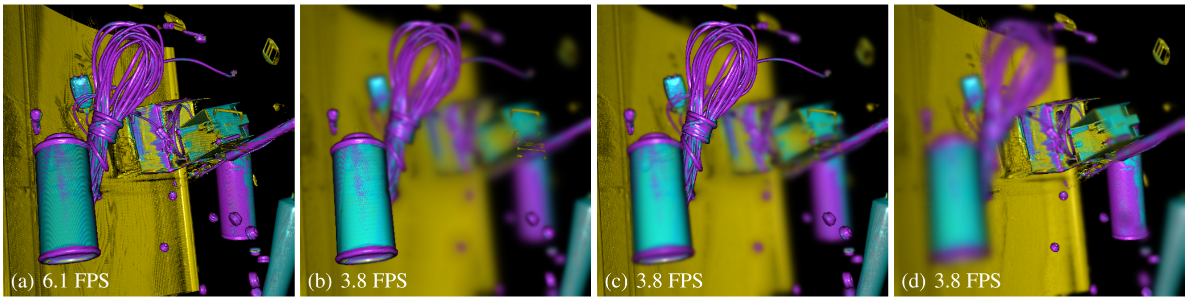

Figure 1: *The backpack data set with* $512 \times 512 \times 373$ *voxels rendered a) with Phong shading only; and b) depth of field with* $R = 0.60$ *focused on the spray can in the foreground, c) the wires behind it and d) the boxes with the other spray can in the back ground. In each image 1469 slices were taken and gradients were estimated on the fly.*

Abstract

In this paper, a method for interactive direct volume rendering is proposed for computing depth of field effects, which previously were shown to aid observers in depth and size perception of synthetically generated images. The presented technique extends those benefits to volume rendering visualizations of 3D scalar fields from CT/MRI scanners or numerical simulations. It is based on incremental filtering and as such does not depend on any precomputation, thus allowing interactive explorations of volumetric data sets via on-the-fly editing of the shading model parameters or (multi-dimensional) transfer functions.

Categories and Subject Descriptors (according to ACM CCS): Computer Graphics [I.3.7]: Three-Dimensional Graphics and Realism— Subjects: Color, shading, shadowing, and texture

1. Introduction

Volumetric rendering is widely used to visualize 3D scalar fields from CT/MRI scanners and numerical simulation datasets. An important aspect of volumetric rendering is the ability to provide perceptual cues to aid in understanding of structures contained in the data.

Recently, a perceptual model [HCOB10] has been proposed that explains how the human visual system uses the depth of field introduced by the lens of the eye to aid inferring absolute distances. This model allows one to manipulate the conveyed scale of synthetically generated images, and as such can be used to improve the effectiveness of scientific visualizations, by emphasizing different scales.

Depth of field effects have been researched in the fields of realistic image synthesis where stochastic sampling is often used to generate high quality images, at the cost of considerable computation time. In contrast, there exists a large body of interactive techniques that allow the approximation of depth of field effects for scenes of polygonal geometry, whereas depth of field effects for volumetric rendering have not received a similar attention.

In this paper, we present a method for interactive direct volume rendering that allows the computation of depth of field effects for volumetric data sets with both solid and transparent features. The proposed method is based on incremental filtering, which has been successfully used in the past to approximate integration for computing advanced scattering and occlusion effects [KPH*03, SPH*09, vPBV10].

Depth of field effects can be easily integrated into a slice-based direct volume rendering system using an incremental filtering approach, which does not require any precomputation, thus allowing interactive examinations of volumetric data sets via on-the-fly editing of the shading model parameters or (multi-dimensional) transfer functions, enabling the classification of structures in the data set as solid or semi-transparent. The presented technique also considers features in front of the focal plane correctly, which are difficult to handle for alternative depth of field methods based on solid surfaces.

This paper is structured as follows: Section 2 discusses methods related to depth of field and related Focus+Context techniques. Section 3 discusses depth of field and outlines an integration into a slice-based volume renderer. Results are presented and discussed in Section 4, followed by conclusions and future work in Section 5.

2. Related Work

Depth of field (DOF) is common in photography and is often simulated in ray tracing to enhance realism. In visualization, it has been mainly used to improve depth perception by blurring objects that would appear out of focus when viewed with a physical camera or to guide users to salient regions by blurring less relevant regions in Focus+Context visualizations.

Kosara *et al.* [KMH01] proposed to blur out objects which are not currently relevant in a scene by applying a box filter using texture mapping hardware. From the user study [KMH*02] they subsequently carried out, they found that participants were able to locate objects based in those visualizations similar compared to using color alone or both together, indicating that blurring can convey further information useful for the comprehension of a specific visualization.

Other Focus+Context techniques are also often used in volume rendering. Viola *et al.* [VKG04] investigated altering the opacity to create cut-away views where opacity and color is modulated to attract the user's gaze to the point of interest. Wang *et al.* [WZMK05] used the optical properties of a lens to magnify the region of interest to see it in greater detail, while deemphasizing the surrounding region which gets slightly distorted and blurry as a result. Along the same lines, Wang *et al.* [WWLM10] use a grid based approach to enlarge the region of interest at the expense of the other regions.

Ropinski *et al.* [RSH06] evaluated DOF among a number of other techniques to help depth perception in angiography where they found it to be useful to improve depth perception. Ciuffreda *et al.* [CWV07] present an empirically based, conceptual model of human blur perception and its impact on the blur-based depth-ordering of objects. Held *et al.* [HCOB10] present a probabilistic model of blur, introduced by the lens of the human eye, which can be used by a viewer together with further relative depth information to give cues about absolute distances in a scene. Their model was validated by a user study where subjects had to estimate absolute distances of objects in synthetic polygonal scenes. Our proposed depth of field method can be used achieve similar effects for visualizations of volumetric data sets.

Since depth of field has been introduced in ray tracing [Coo84, NSG90], many approximations to this technique have been proposed for computing depth of field effects interactively, especially considering opaque geometry. Many of these approaches are image-space techniques which blur an initially generated in-focus image. Potmesil and Chakravarty [PC81] use linear filtering in a post-processing stage to adaptively blur images according to their distance from the focal plane. This technique is fast, since it does not attempt to minimize depth discontinuities or color bleeding. A detailed survey of further depth of field techniques has been compiled by Barsky *et al.* [BK08].

Barsky *et al.* [BHK*a, BHK*b] improve this technique by rendering scenes consisting of opaque geometry into several layers and blurring them individually, which Kraus *et al.* implemented on the GPU [Kra07]. Our proposed method similarly performs blurring on the layers inherent to the slice traversal of a direct volume rendering system. The incremental nature of the blurring reduces aliasing artifacts due to undersampling.

Depth of field has also been investigated for volume rendering by Crassin *et al.* [CNL*09, CNLE09] to render large out-of-core data sets. They compute an object space sparse voxel tree and use it to render massive data sets via a view-dependent level-of-detail approach which can also be used to approximate depth of field effects by performing mipmap level lookups into their data structure. They note that animation is an issue with their approach, indicating that their preprocessing is rather expensive, though details are not given. The method proposed in this paper does not depend on precomputation and as such allows on-the-fly replacement of the volume data, e.g. by paging in volumes of a time series.

The method proposed by Ropinski *et al.* [RSH06] computes a depth buffer based on opacities accumulated along rays while marching through a volume. The depth values then get blurred across all those pixels with a depth above a certain threshold, thus approximating out-of-focus objects behind the focal plane. The depth of field technique proposed in this paper additionally considers the blurring of foreground objects and trivially handles transparent features.

3. Depth of Field

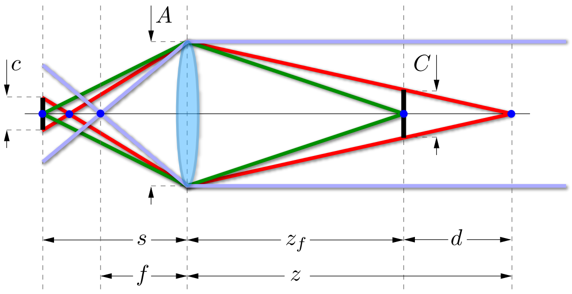

Figure 2: *Illustration of the basic geometric setup for the depth of field scene with the lens and rays for calculating the view space circle of confusion C of an unfocused object at distance z, when the lens is focused at the distance z_f.*

The depth of field is the *in-focus* range of a scene acquired by a camera with sufficient sharpness, in contrast to the *out-of-focus* regions of a scene, which appear gradually blurred in an image. The *circle of confusion*, which can be derived using basic geometry as illustrated in Figure 2, describes the shape and size of the blur spot.

It can be computed in view space C (Equation 1) and image space c (Equation 2) using the focused distance z_f of the lens, its aperture A, and the distance z of an object with respect to the lens:

$$C(z) = A\,\frac{|z_f - z|}{z} \tag{1}$$

$$c(z) = C(z)\,m \tag{2}$$

$$m = \frac{f}{z_f - f} \tag{3}$$

The projection distance s is a function of the lens specific focal length f and the focused distance z_f. Together, they determine the magnification m of the projected image on the image plane.

Different lenses have different depths of field, depending on their geometrical and material properties, such as aperture, focal length and the distance they are focused on. These have been successfully used by photo and film artists to create depth of field effects to guide a viewers gaze onto interesting parts of a scene.

Those techniques have also been used to improve viewer immersion of synthetically generated images, often with focus on real-time entertainment or offline movie rendering. Sections 3.1 and 3.2 discuss how depth of field effects can be integrated into a slice-based direct volume renderer to allow visualization of volumetric data sets.

3.1. Integration into a Slice-Based Volume Renderer

Depth of field effects can be interactively computed by modifying the traversal of the slices and their compositing of a slice-based volume renderer [WVW94] which renders proxy geometry, typically slices created by intersecting view aligned planes with the bounding box of the volume on graphics hardware. A fragment program then computes color and opacity values to be composited with the image of previous slices already stored in the frame buffer.

The CPU-computed proxy geometry, namely view-aligned slices, are first partitioned into those in front and those behind the focal plane of the lens, as illustrated in Figure 3. Each set is then traversed individually and composited into a separate buffer, both being subsequently blended together.

The slices in front of the focal plane are projected into an intermediate buffer and traversed in front-to-back order during which the blurring introduced by the depth of field is approximated by averaging and compositing multiple samples from the previous intermediate buffer. The current and next intermediate buffers are swapped and the algorithm proceeds to the next slice.

Similarly, the slices behind the focal slice are traversed in back-to-front order and composited into the eye buffer during which multiple samples are also taken and averaged from the previous eye buffer, subsequently followed by swapping the current and next eye buffers before proceeding with the next slice. The process of averaging samples from the eye and intermediate buffers is detailed in Section 3.2.

The intermediate buffer is then blended on top of the eye buffer using the over operator, and the result then gets copied into the frame buffer of the visible window, converting the final values from floating-point format to the fixed-point format usually used for displaying images on the screen. If required, tone-mapping can be applied here.

3.2. Incremental Filtering for Computing Depth of Field

Blurring across the circle of confusion, as described in Equation 1 and illustrated in Figure 3, is done by averaging samples from the previous eye/intermediate buffer in a neighborhood around the current fragment when updating the current eye/intermediate buffer.

The proposed method is integrated into an existing volume rendering system, which uses the perspective projection matrix P to compute the image space circle of confusion, instead of directly evaluating Equations 2 and 3.

The view space circle of confusion C for a given slice with view space z-component z is projected from view space into

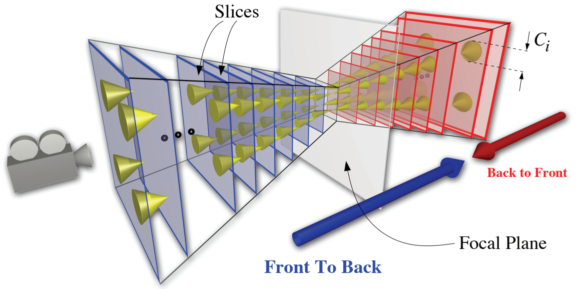

Figure 3: *Geometric setup of computing depth of field effects by traversing the slices of a direct volume rendering system in two separate passes; those in front of the focal plane in front-to-back order and those behind the focal plane in back-to-front order. The bases of the yellow cones denote the regions from which samples are taken during the incremental filtering. Slices are scaled in screen space to guarantee that the filtering is done properly with respect to slice boundaries, due to linearly increasing circles of confusion C_i, which are 0 at the slice closest to the focal distance z_f.*

clip space $\vec{C}_c$ by the projection matrix P, and transformed by the matrix T from the clip space range of $[-1,\ 1]^2$ to the texture coordinate space range of $[0,\ 1]^2$, to yield the texture space circle of confusion $\vec{C}_t$, shown in Equation 5:

$$\vec{C}_c \;=\; \begin{pmatrix} x_t \\ y_t \\ z_t \\ w_t \end{pmatrix} = T \cdot P \cdot \begin{pmatrix} C \\ C \\ z \\ 1 \end{pmatrix} \qquad (4)$$

$$\vec{C}_t \;=\; \begin{pmatrix} \frac{x_t}{w_t} \\ \frac{y_t}{w_t} \end{pmatrix} \qquad (5)$$

The texture space circle of confusion $\vec{C}_t$ is passed to the fragment shader where it is used to compute texture coordinate offsets for sampling the previous eye/intermediate buffer. A set of N sample offsets $\vec{p}_i$ is generated on the ellipse with extent C_t and added to the projected texture coordinate $\vec{f}_t$ of the currently processed fragment, yielding the set of texture space sample positions $\vec{t}_i$, as Equation 6 shows:

$$\vec{t}_i = \vec{f}_t + \vec{p}_i \qquad (6)$$

Sample offsets $\vec{p}_i$ were initially created using a Poisson distribution, but experimentation suggested a regular grid of user specified resolution (e.g. $N = 4$, for a 2×2 grid) as a reasonable compromise between image quality and performance.

The previous eye/intermediate buffer is then sampled at all sample positions $\vec{t}_i$ and those are accumulated and divided by the total number of samples N. This average is then taken as the *destination* parameter in the front-to-back compositing equation for the current intermediate buffer, or the back-to-front composition equation for the current eye buffer respectively. The *source* parameter is the result of evaluating the 2D scalar/gradient magnitude transfer function and a shading model, such as the commonly used emission-absorption model or the Phong surface shading model.

In typical slice-based volume rendering systems, proxy geometry is generated by intersecting slicing planes against the bounding box of the data set, however the increase of C as a function of d requires scaling the slice in screen space by C to guarantee that filtering is handling slice boundaries correctly. Figure 3 illustrates the required proxy geometry, namely two frustums joined together with their top surfaces at the focal plane, their taper being a function of the circle of confusion C.

However many data sets have interesting features surrounded by large homogeneous regions, which are often culled by the transfer function, thus hiding any potential filtering artifacts close to the slice boundaries, therefore it is sufficient to intersect the bounding box, which can be scaled by a constant factor, if required.

4. Results and Discussion

The methods was implemented using OpenGL and Cg running on a mid-range NVIDIA GeForce GTX 460 GPU with 1 GB of video memory. The images were rendered at a 512×512 resolution into 16-bit precision floating-point buffers with an inter-slice distance of $d = 0.001$ to capture high-frequency details introduced by multi-dimensional transfer functions and unless noted otherwise, depth of field effects were rendered by taking 2×2 samples when blurring during slice traversal.

During experimentation, it was observed that it was rather difficult and impractical for a user to manipulate lens parameters, such as aperture A, focal length f and focal distance z_f directly, in order to achieve a specific visualization. As such a more intuitive way was derived, based on the observation that the circle of confusion is proportional to the difference $d = |z_f - z|$ between the object distance z and the focal distance z_f, which describes depth of field *qualitatively*, whereas the lens properties describe it in a *quantitative* way. The results presented were created with a simplified model to specify the depth of field parameters, as outlined in Equation 7.

$$C(d) \;=\; Rd \qquad (7)$$

$$A(z_f) \;=\; 2Rz_f \qquad (8)$$

The camera always focuses on a user specified point in model space, and as such, the focal distance z_f is changing in view space when the user changes the camera position or orientation. A scalar R then determines the rate with which the circle of confusion increases as a function of the distance d. The lens aperture A, shown in Equation 8 is then a function of z_f and R.

This frees the user from the burden to refocus z_f when the camera is moved, e.g. when zooming into a specific region of the data set, and also allows quick shifts of focus, e.g. from the front of the volume to the back, irrespective of where the camera is located. Specifying R directly instead of A keeps the region in focus at a constant range, independent of the focus distance.

The backpack CT data set is shown in Figure 1, where the camera is focusing on objects at different depths, such as the spray can in the foreground, the wires behind, and the boxes and the other spray can in the background, as shown in Figures 1(b) to 1(d). The various degrees of blur induced by the depth of field approximation makes it easier to intuitively reason about their spatial arrangement. Noticeable is also the increased ability to locate the small circular objects distributed throughout the scene, compared to Figure1(a).

Figure 4 shows the Richtmyer-Meshkov fluid data set rendered with Phong shading in Figure 4(a). The high frequencies of the features contained in this fluid simulation data set makes it relatively hard to make out distinct features, it is difficult to get an intuition how those bubbles are arranged with respect to each other. Figure 4(b) shows the same scene, but this time with the camera focused on a bubble in the foreground. Figure 4(c) then emphasizes the prominent yellow bubble in the center with the surrounding features gradually becoming out-of-focus. Moving the focus plane behind the yellow bubble, as shown in Figure 4(d), guides the gaze of the observer towards the other yellow bubble at the boundary of the data set, shown in the far back. The interactive depth of field approximation is able to point out features at different distances, while maintaining a global impression of the whole data set.

Figure 5 shows the aneurysm MRI data set rendered with without depth of field in Figure 5(a) and with depth of field effects with focus on the enlarged blood vessel. The circle of confusion rate of change is increased from $R = 0.55$ in Figure 5(b) to $R = 6.03$ in Figure 5(c), causing features relatively far away from the focus plane to be blurred more, thus expanding their screen space coverage. Features closer towards the focus plane are less affected, especially when their respective image space circles of confusion are smaller than the pixel size of the blurred buffers. Increasing the number of samples taken during blurring changes the shape of the blur subtly (Figure 5(c) and Figure 5(d)), especially for regions that are strongly defocused, at the price of a considerable performance drop.

Figure 6 shows the abdomen of the Visible Human CT data set where the bones and parts of the internal organs were classified using the transfer function. The lack of depth of field, as shown in Figure 6(a), yields significant visual clutter, which is drastically reduced in Figure 6(b) with the camera focused on the spine. The blurred internal organs and the semi-transparent skin layer support the comprehension of the relative arrangement of the features contained in the data set.

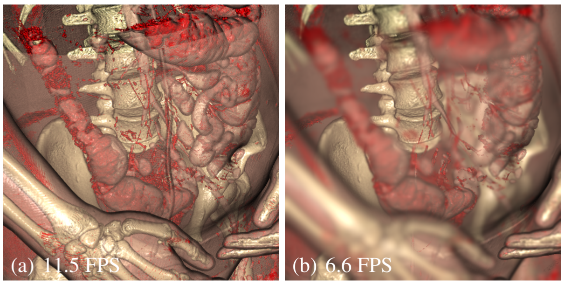

Figure 6: *The abdomen part of the visible human data set with* $512 \times 512 \times 512$ *voxels rendered with a) Phong shading and b) a depth of field effect with a circle of confusion rate of change of* $R = 0.41$*. The focus is on the spine, the skin is rendered semi-transparently to give additional context. For this image, 853 slices were taken and gradients were computed on the fly.*

5. Conclusion and Future Work

In this paper, a method has been presented which adds depth of field effects to direct volume rendering, which were shown to aid in the inference of depth cues. The method does not rely on pre-computation and therefore allows interactive manipulation of camera position and transfer function, for both solid and transparent features.

Incorporating the proposed method into an existing slice-based direct volume rendering system required minimal modifications and allowed the rendering of depth of field effects with frames rate reduced by a factor less than 50%. Performance can potentially be increased by changing the number of samples during incrementally filtering as a function of the texel space circle of confusion.

In the future, we would like to extend the proposed technique to handle tilt shifting effects and chromatic aberration, which change the circle of confusion as a function of a tilted focal plane and the wavelength of light respectively.

It would also be interesting to quantify the perceptual gains of depth of field effects and their interplay with different local and global shading models in the context of direct volume rendering and visualization.

References

[BHK*a] Barsky B., Horn D., Klein S., Pang J., Yu M.: Camera models and optical systems used in computer graphics: Part i, object-based techniques. In *Computational Science and Its Applications - ICCSA 2003*, Lecture Notes in Computer Science. 2

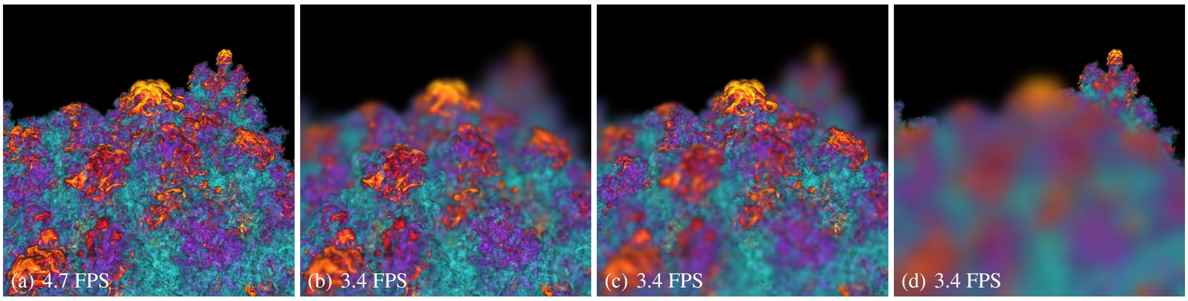

Figure 4: *A* $1024 \times 1024 \times 384$ *voxel subset of the Richtmyer-Meshkov data set rendered with a) Phong shading and subsequently DOF with* $R = 1.32$ *and focus on b) a bubble in the front c) the center bubble d) the far yellow bubble. In each image 1350 slices were taken and gradients were computed on the fly due to memory constraints.*

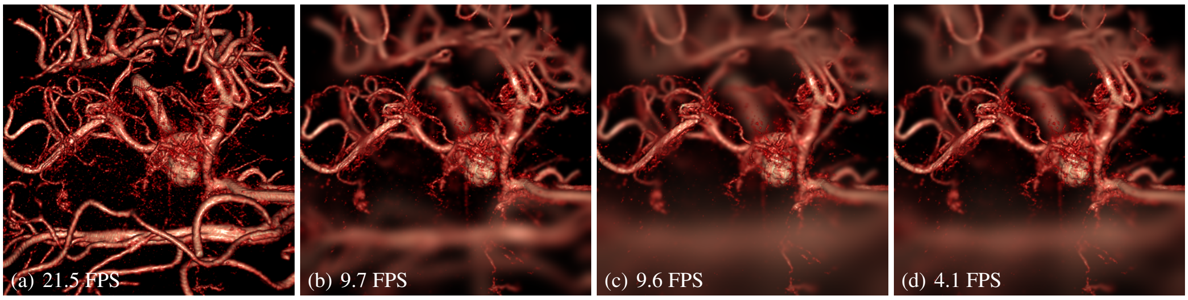

Figure 5: *The aneurysm data set with* $256 \times 256 \times 256$ *voxels rendered with Phong shading and, a) no depth of field, b) depth of field with focus on the damaged blood vessel and circle of confusion rate of change* $R = 0.55$ *and c) focused at the same plane but a stronger depth of field effect by setting* $R = 6.03$*, d) which is similar to the previous image, but a sample grid resolution of* 4×4 *compared to the* 2×2 *grid used in the previous images. For each image, 655 slices were taken and gradients were precomputed and fetched from a 3D texture during rendering.*

[BHK*b] Barsky B., Horn D., Klein S., Pang J., Yu M.: Camera models and optical systems used in computer graphics: Part ii, image-based techniques. In *Computational Science and Its Applications - ICCSA 2003*, Lecture Notes in Computer Science. 2

[BK08] Barsky B. A., Kosloff T. J.: Algorithms for rendering depth of field effects in computer graphics. In *Proceedings of the 12th WSEAS international conference on Computers* (Stevens Point, Wisconsin, USA, 2008), World Scientific and Engineering Academy and Society (WSEAS), pp. 999–1010. 2

[CNL*09] Crassin C., Neyret F., Lefebvre S., Sainz M., Eisemann E.: Beyond triangles: gigavoxels effects in video games. In *SIGGRAPH 2009: Talks* (2009), SIGGRAPH '09, ACM, pp. 78:1–78:1. 2

[CNLE09] Crassin C., Neyret F., Lefebvre S., Eisemann E.: Gigavoxels: ray-guided streaming for efficient and detailed voxel rendering. In *Proceedings of the 2009 symposium on Interactive 3D graphics and games* (2009), I3D '09, ACM, pp. 15–22. 2

[Coo84] Cook R. L.: Computer Graphics Volume 18, Number 3 July 1984. *Computer 18*, 3 (1984), 137–145. 2

[CWV07] Ciuffreda K. J., Wang B., Vasudevan B.: Conceptual model of human blur perception. *Vision Research 47*, 9 (2007), 1245 – 1252. 2

[HCOB10] Held R. T., Cooper E. A., O'Brien J. F., Banks M. S.: Using blur to affect perceived distance and size. *ACM Transactions on Graphics 29*, 2 (Mar. 2010), 1–16. 1, 2

[KMH01] Kosara R., Miksch S., Hauser H.: Semantic depth of field. In *INFOVIS* (2001), pp. 97–104. 2

[KMH*02] Kosara R., Miksch S., Hauser H., Schrammel J., Giller V., Tscheligi M.: Useful properties of semantic depth of field for better f+c visualization. In *Proceedings of the symposium on Data Visualisation 2002* (Aire-la-Ville, Switzerland, Switzerland, 2002), VISSYM '02, Eurographics Association, pp. 205–210. 2

[KPH*03] Kniss J., Premoze S., Hansen C., Shirley P., McPherson A.: A model for volume lighting and modeling. *IEEE Transactions on Visualization and Computer Graphics 9*, 2 (2003), 150–162. 2

[Kra07] Kraus M.: Depth-of-field rendering by pyramidal image processing. *Computer Graphics Forum 26* (September 2007), 645–654(10). 2

[NSG90] Novins K. L., Sillion F. X., Greenberg D. P.: An efficient method for volume rendering using perspective projection. *SIGGRAPH Comput. Graph. 24* (November 1990), 95–102. 2

[PC81] Potmesil M., Chakravarty I.: A lens and aperture camera model for synthetic image generation. *SIGGRAPH Comput. Graph. 15* (August 1981), 297–305. 2

[RSH06] Ropinski T., Steinicke F., Hinrichs K.: Visually supporting depth perception in angiography imaging. In *Smart Graphics* (2006), Springer, pp. 93–104. 2

[SPH*09] Schott M., Pegoraro V., Hansen C., Boulanger K., Bouatouch K.: A directional occlusion shading model for interactive direct volume rendering. In *Computer Graphics Forum (Proceedings of Eurographics/IEEE VGTC Symposium on Visualization 2009)* (2009), vol. 28. 2

[VKG04] Viola I., Kanitsar A., Groller M. E.: Importance-driven volume rendering. In *Proceedings of the conference on Visualization '04* (Washington, DC, USA, 2004), VIS '04, IEEE Computer Society, pp. 139–146. 2

[vPBV10] Šoltészová V., Patel D., Bruckner S., Viola I.: A multidirectional occlusion shading model for direct volume rendering. *Computer Graphics Forum 29*, 3 (june 2010), 883–891. 2

[WVW94] Wilson O., VanGelder A., Wilhelms J.: *Direct volume rendering via 3D textures*. Tech. rep., University of California at Santa Cruz, Santa Cruz, CA, USA, 1994. 3

[WWLM10] Wang Y.-S., Wang C., Lee T.-Y., Ma K.-L.: Feature-preserving volume data reduction and focus+context visualization. *IEEE Transactions on Visualization and Computer Graphics 99*, RapidPosts (2010). 2

[WZMK05] Wang L., Zhao Y., Mueller K., Kaufman A.: The magic volume lens: an interactive focus+context technique for volume rendering. In *Visualization, 2005. VIS 05. IEEE* (2005), pp. 367 – 374. 2