New Method for Opacity Correction in Oversampled Volume Ray Casting

Jong Kwan Lee
Department of Computer Science
University of Alabama in Huntsville
Huntsville, AL 35899 USA
jlee@cs.uah.edu

Timothy S. Newman
Department of Computer Science
University of Alabama in Huntsville
Huntsville, AL 35899 USA
tnewman@cs.uah.edu

ABSTRACT

A new opacity correction approach for oversampled volume ray casting is introduced. While the only existing opacity correction method in the literature is based on the assumption of dataset homogeneity, the new opacity correction method introduced in this paper is a faster, generalized voxel-by-voxel approach which does not assume dataset homogeneity. The new opacity correction avoids the dataset homogeneity assumption by introducing a new opacity correction factor for the samples in each voxel. Its performance improvement over the existing opacity correction approach is also exhibited for real volumetric datasets.

Keywords: Volume visualization, Volume ray casting, Oversampling, Opacity Correction

1 INTRODUCTION

Volume visualization has been a powerful aid to knowledge discovery in many application areas. Direct volume rendering (DVR) is one of the classes of volume visualization approaches. DVR involves constructing an image representation for a volumetric dataset without first building any intermediate representation (e.g., a mesh of triangles).

Volume ray casting (VRC) is one of the widely studied and applied DVR techniques (e.g., [Lev88, Mor02, Wei03, Kle05]). Ray casting involves forming an image by passing rays from image locations through the volumetric dataset and integrating light effects on the rays (e.g., integrating light transmissions along the rays). Samples are composited along each ray in a front-to-back or back-to-front manner. Equations (1) and (2) are the discretized formulations of the front-to-back and back-to-front compositions, respectively:

$$I_t = \sum_{i=0}^{n} I_i \times \prod_{j=0}^{i-1} (1-\alpha_j)\,, \tag{1}$$

$$I_t = \sum_{i=0}^{n} I_i \times \prod_{j=i+1}^{n} (1-\alpha_j)\,, \tag{2}$$

where I_t is the final composited intensity for a ray, n is the number of samples for a ray, I_i is the intensity of the i^{th} sample, and α_j is the opacity of the j^{th} sample. The term $(1-\alpha_j)$ in Equations (1) and (2) is the transparency of the j^{th} sample. Front-to-back composition has an advantage of allowing early ray termination since the accumulated transparency from previous samples is kept. The intensity, I_i, of a sample in these formulations can be modeled as:

$$I_i = C_i \times \alpha_i\,, \tag{3}$$

where C_i is the color of the sample.

The number of sample points on each ray can vary as long as it is above the Nyquist sampling frequency to avoid aliasing [Lic98, Swa97]. Moreover, to better-simulate the continuous integration of the light effects in a discrete space, a high sampling rate can be used. However, the result of the ray sample composition should not be negatively impacted when the number of samples on each ray exceeds a unit-sampling rate. That is, when there is oversampling, the opacity must be corrected such that the final composited opacity value is not over-composited. In addition, since the ray intensity composition is not a simple linear composition, as seen in Equations (1) and (2), the opacity correction should be applied in a non-linear way.

Figs. 1 and 2 show the impact of over-composited opacity. In Figs. 1 (a) and 2 (a), isosurface renderings of CT datasets of an engine block and a foot are shown, respectively. Five times oversampled renderings are shown for each of these in Fig. 1 (b) and Fig. 2 (b), respectively. (For these figures, a gradient-based local reflection model was used to determine the sample colors.) In these oversampled renderings, it is harder to see the details of the structures due to the over-composited opacity. For example, the interior structures of the engine block are not visible in Fig. 1 (b) and the bone

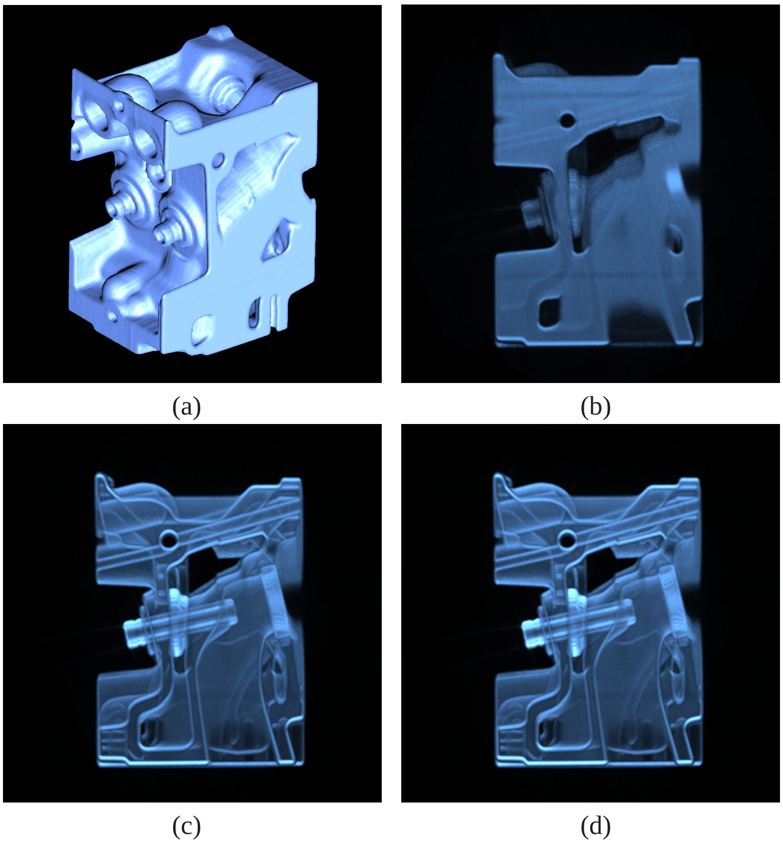


Figure 1: Renderings (Engine, 256×256×256 CT) from (a) Marching Cubes isosurfacing and (b-d) oversampled (5 times) volume ray casting: (b) without opacity correction, (c) with Lacroute/Lichtenbelt et al.'s opacity correction, and (d) with new opacity correction.

structures are not clear in Fig. 2 (b). Thus, when oversampling, an opacity correction is needed to correct such artifacts.

In this paper, we describe a new method to address the issue of over-compositing of opacity in oversampled volume ray casting.

2 RELATED WORK

In this section, related work is discussed. Lichtenbelt et al. [Lic98] have discussed the problem of over-composited opacity and described an opacity correction formula based on the assumption that the datasets are homogeneous (i.e., all sample values are the same). (However, most real volumetric datasets are not homogeneous.) When the dataset is homogeneous and front-to-back composition is used, the method to correct opacity described by Lichtenbelt et al. uses the rationale that the unit-sampled intensity in a voxel should be equal to the oversampled intensity in the voxel. Thus,

$$\begin{aligned} C\times\alpha = C\times\alpha' &+ \{C\times\alpha'(1-\alpha')\} \\ &+ \left\{C\times\alpha'(1-\alpha')^2\right\} + \dots \\ &+ \left\{C\times\alpha'(1-\alpha')^{N-1}\right\}, \end{aligned} \tag{4}$$

where C is the color, α is the original opacity determined by an opacity transfer function, α' is the corrected opacity, and N is the oversampling factor. Based on this expression, Lichtenbelt et al. [Lic98] derived the opacity correction formula:

$$\alpha' = 1 - \sqrt[N]{1-\alpha}\,. \tag{5}$$

This correction is applied on a voxel-by-voxel basis (i.e., a separate correction is computed for each sample in each voxel). However, Equation (5)'s opacity adjustment will only be appropriate when the assumption of dataset homogeneity holds.

The opacity correction described in Lichtenbelt et al. [Lic98] was motivated by work by Lacroute [Lac95], who presented an opacity correction formula in terms of sample spacing for the shear-warp factorization-based rendering. Lacroute's formula is equivalent to Equation (5). Others [Pfi05, Sch03, Wei00] have also used or discussed use of this opacity correction formula. Schulze et al. [Sch03] have suggested that an opacity correction factor be computed using the formula in Equation 5 with the corrected opacity used to correct the color as follows:

$$C_i' = C_i \times \frac{\alpha_i'}{\alpha_i}\,, \tag{6}$$

where C_i' is the corrected color. Schulze et al. applied their corrections to undersampled ray casting.

Although Lacroute [Lac95] and Lichtenbelt et al. [Lic98] have noted that the opacity correction in Equation 5 can be implemented by a lookup table for α' as a function of α, discretizing the opacity can result in loss of information. Thus, in the experiments we report later, a lookup table for α' is not used.

2.1 Toward the New Approach

The new opacity correction method introduced here is an efficient approach; it can be computed much faster than the existing approach. It is also able to produce renderings of comparable quality to those of the prior (i.e., Lacroute/Lichtenbelt et al.) approach. Moreover, it is not based on the assumption of homogeneity of the data. We note that it is also possible to employ a lookup table for the new opacity correction method, but we do not use a lookup table for it for the reason we mentioned above.

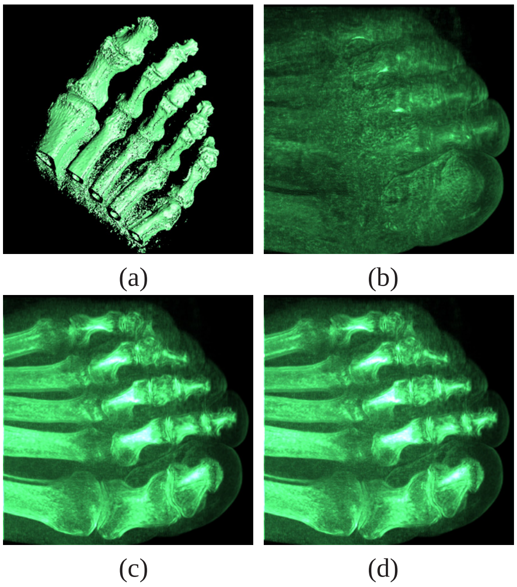

(a) (b) (c) (d)

Figure 2: Renderings (Foot, 256×256×256 CT) from (a) Marching Cubes isosurfacing and (b-d) oversampled (5 times) volume ray casting: (b) without opacity correction, (c) with prior (Lacroute/Lichtenbelt et al.'s) opacity correction, and (d) with new opacity correction.

3 NEW OPACITY CORRECTION APPROACH

The new opacity correction method we introduce here is derived from a generalized form of an opacity-based derivation similar to Equation (5)'s derivation. Like the prior method, it is a voxel-by-voxel approach, although it does not assume homogeneity of the dataset.

3.1 Application Conditions

In the work here, we have used two color transfer functions. Both transfer functions used trilinear interpolation to interpolate samples that were not on the grid points.

The first is for gray-scaled application:

$$C_i = \frac{D_i}{M}, \qquad (7)$$

where M is the maximum sample data value in the dataset and D_i and C_i are the sample data value and color, respectively, for the i^{th} sample in a cell. For generic application, especially for 8-bit data, the simple linear transfer function in Equation (7) is suitable.

The second transfer function is for color-scaled application:

$$C_i = local\_reflection\_model\,(G_i), \qquad (8)$$

where G_i is the gradient of the i^{th} sample and C_i is the color that is associated with the i^{th} sample in a cell. The local reflection model utilizes ambient, diffuse, and specular light components and uses the sample gradient as the surface normal. For the sample gradient, we employed two different approximation schemes (which produce visually similar results): (1) using trilinear interpolation of the linear central-difference gradients at the 8 grid points in the voxel and (2) using linear central differences at the sample points. (Figs. 1, 5, and 6 use scheme (1); Fig. 2 uses scheme (2).) For the opacity transfer function, we have used a simple linear transfer function similar to Equation (7) (i.e., $\alpha_i = D_i/M$, where α_i is the opacity for the i^{th} sample in a cell).

3.2 New Opacity Correction

In this subsection, the derivation of the new opacity correction formulation is shown. Then, we show an approximation that allows fast computation of the new opacity.

For a unit-sampling versus an oversampling in one voxel, our model assumes the unit-sampled composited transparency and the oversampled composited transparency should be equivalent for the voxel. For example, for the case of two times oversampling, the composited transparency within a voxel should be:

$$(1-\alpha_{u1}) = (1-p\,\alpha_{o1}) \times (1-p\,\alpha_{o2}), \qquad (9)$$

where α_{ui} is the unit-sampled opacity for the i^{th} sample in the voxel, α_{oi} is the oversampled opacity of the i^{th} sample in the voxel (i.e., $\alpha_{u1} = \alpha_{o1}$), and p is the new opacity correction factor. As seen in Equation (9), our opacity correction does not depend on dataset homogeneity (i.e., when there are different α_i values for different samples). Then, Equation (9) can be rearranged as follows:

$$F : (\alpha_{o1}\,\alpha_{o2})\,p^2 - (\alpha_{o1}+\alpha_{o2})\,p + \alpha_{o1} = 0\,. \qquad (10)$$

$$
\begin{aligned}
&(-1)^0 \left(\prod_{s=1}^{N} \alpha_{os} \right) p^N \\
&+(-1)^1 \left\{ \sum_{t=1}^{N} \left(\prod_{s=1,s\neq t}^{N} \alpha_{os} \right) \right\} p^{N-1} \\
&+(-1)^2 \left\{ \sum_{u=1}^{N-1} \sum_{t=u+1}^{N} \left(\prod_{s=1,s\neq t,u}^{N} \alpha_{os} \right) \right\} p^{N-2} \\
&+(-1)^3 \left\{ \sum_{v=1}^{N-2} \sum_{u=v+1}^{N-1} \sum_{t=u+1}^{N} \left(\prod_{s=1,s\neq t,u,v}^{N} \alpha_{os} \right) \right\} p^{N-3} \\
&\vdots \\
&+(-1)^N \alpha_{o1} p^0 = 0.
\end{aligned}
\tag{11}
$$

In the new method, the expression in Equation (10) is solved for p. Similar formulas can be obtained for oversamplings of more than two times. Those formulas can be generalized, as we show above in Equation (11), for large N, where N is the oversampling factor.

For the general case, our approach's opacity correction factor, p ($0 \leq p \leq 1$), can be determined for each voxel by solving Equation (11) for p for the voxel; we need to solve Equation (11) for p for each voxel.

For example, the opacity correction expression F when $N = 3$ is:

$$
\begin{aligned}
F(p) = {} & (\alpha_{o1}\alpha_{o2}\alpha_{o3})\, p^3 \\
& -(\alpha_{o1}\alpha_{o2} + \alpha_{o2}\alpha_{o3} + \alpha_{o1}\alpha_{o3})\, p^2 \\
& +(\alpha_{o1} + \alpha_{o2} + \alpha_{o3})\, p \\
& -\alpha_{o1}\,,
\end{aligned}
\tag{12}
$$

where $F(p) = 0$ for the value of p that produces the proper correction.

However, Equation (11) can contain high-order polynomials when the oversampling rate is high. To avoid complex computations for solving the high-order polynomials (i.e., $N \geq 3$), we approximate them by fitting 2^{nd} degree polynomials and then solving for p. This approximating expression has the form:

$$\hat{F}(p) = A \times p^2 + B \times p + C\,, \tag{13}$$

with $\hat{F}(p) = 0$ for the p that satisfies the condition. In particular, we have used $p_0 = 0.0$, $p_1 = 0.5$, and $p_2 = 1.0$ for the 2^{nd} degree polynomial fittings (i.e., the fitted polynomial is forced to pass through these three points). For example, at $p_1 = 0.5$, we have $\hat{F}(0.5) = A \times (0.5)^2 + B \times (0.5) + C = F(0.5)$.

The evaluations of Equation (13) produce the system:

$$
\begin{pmatrix} p_0^2 & p_0 & 1 \\ p_1^2 & p_1 & 1 \\ p_2^2 & p_2 & 1 \end{pmatrix}
\begin{pmatrix} A \\ B \\ C \end{pmatrix}
=
\begin{pmatrix} F(p_0) \\ F(p_1) \\ F(p_2) \end{pmatrix},
\tag{14}
$$

where $F(p_i)$ are from Equation (11). Thus, the solution (A, B, C) is:

$$
\begin{pmatrix} A \\ B \\ C \end{pmatrix}
=
\begin{pmatrix} p_0^2 & p_0 & 1 \\ p_1^2 & p_1 & 1 \\ p_2^2 & p_2 & 1 \end{pmatrix}^{-1}
\begin{pmatrix} F(p_0) \\ F(p_1) \\ F(p_2) \end{pmatrix}.
\tag{15}
$$

Once (A, B, C) have been determined, Equation (13) can be solved for the best p using standard quadratic equation solution methods. This process of determining p must be performed for each voxel (i.e., a unique p will be determined for each voxel), although the matrix of p_0, p_1, p_2 values will not change. Thus, a final form of our new opacity correction is as follows:

$$
\begin{aligned}
p &= f(\hat{\alpha})\,, \\
\alpha' &= p \times \alpha\,,
\end{aligned}
\tag{16}
$$

where the function f solves a second degree polynomial given a set of samples, $\hat{\alpha} = \{\alpha_0, \alpha_1, ..., \alpha_{N-1}\}$, in the voxel and α' is the corrected opacity.

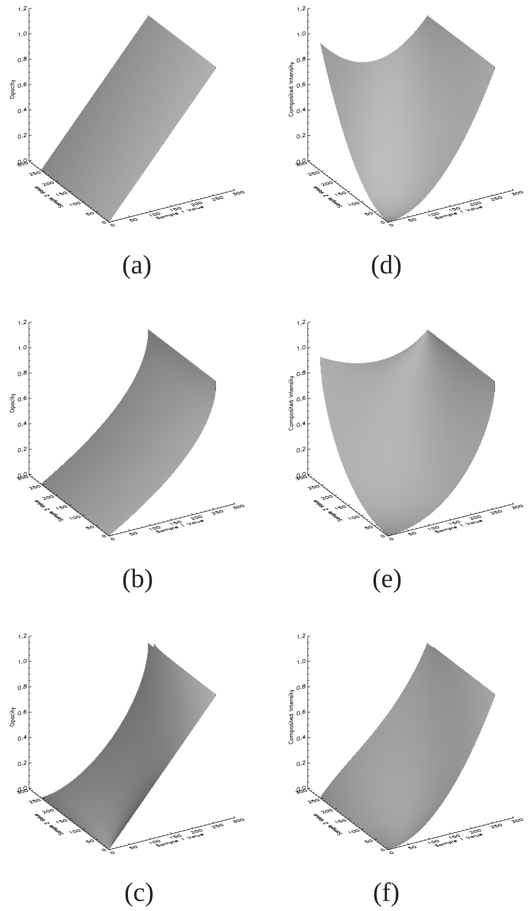
A lookup table can be used in implementing the new opacity correction. However, our reports here are not based on lookup table use.

4 EXPERIMENTAL RESULTS

In this section, we consider the behavior (including some limitations) of the new opacity correction and compare the performances of the existing opacity correction and the new opacity correction.

4.1 Analysis Within A Voxel

First, we consider the within-voxel behavior of the Lacroute/Lichtenbelt et al.'s opacity correction and of our new opacity correction.



(a) (d)

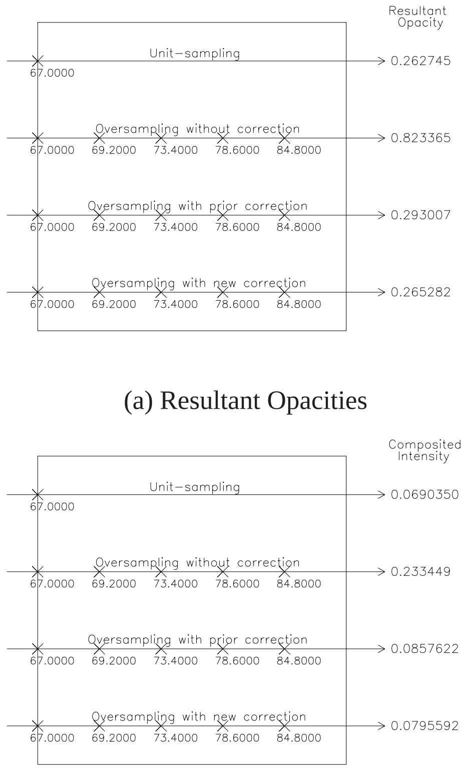
(b) (e)

(c) (f)

Figure 3: Oversampling's (2 times) resultant opacities in (a)-(c): (a) without correction, (b) with prior (Lacroute/Lichtenbelt et al.'s) correction, and (c) with new correction. Corresponding composited intensity surfaces shown in (d)-(f): (d) without correction, (e) with prior correction, and (f) with new correction. All surfaces exhibit results over the universe of possible combinations of sample values for 8-bit data.

4.1.1 Synthetic Data-Testing All Combinations

The characteristics of different opacity corrections can be observed using synthetic data to compare the resultant opacity surfaces and composited intensity surfaces. Here, we consider two times oversampling for the cases of (1) compositing without opacity correction, (2) compositing with the prior (Lacroute/Lichtenbelt et al.'s) opacity correction, and (3) compositing with the new opacity correction. The resultant opacity surfaces and composited intensity surfaces for all possible combinations of sample values of 8-bit data are shown in Fig. 3. Parts (a)-(c) show the resultant opacities for these three cases. Corresponding composited intensity surfaces (i.e., composited intensities for all combinations of two samples) for Figs. 3 (a)-(c) are shown in Figs. 3 (d)-(f). While the resultant opacity surface without correction in Fig. 3 (a) is linear, the resultant opacity surface with the prior opacity correction and with the new correction in Figs. 3 (b) and (c), respectively, show non-linear variations for different sample values. In addition, the resultant opacity surface in Fig. 3 (c) for the new method shows a more complicated, non-linear variation in its shape. There are more apparent differences between the composited intensity surfaces in Fig. 3 (e) for the prior opacity correction and Fig. 3 (f) for the new correction. For the case of two times oversampling, the homogeneity assumption of the prior opacity correction does not allow the correction to vary suitably in the cases where values are the least homogeneous (e.g., the combination of one small and one large value). Specifically, when the first sample has a low value and the second sample has a high value, the second sample should have very little impact on the intensity composition within a voxel. Fig. 3 (f) shows that



(a) Resultant Opacities

(b) Composited Intensities

Figure 4: Example comparison of resultant/composited opacities and intensities for a voxel: Rays within a voxel for unit-sampling, oversampling without correction, oversampling with Lacroute/Lichtenbelt et al.'s correction, oversampling with new correction from top to bottom, respectively. (All oversamplings: 5 times)

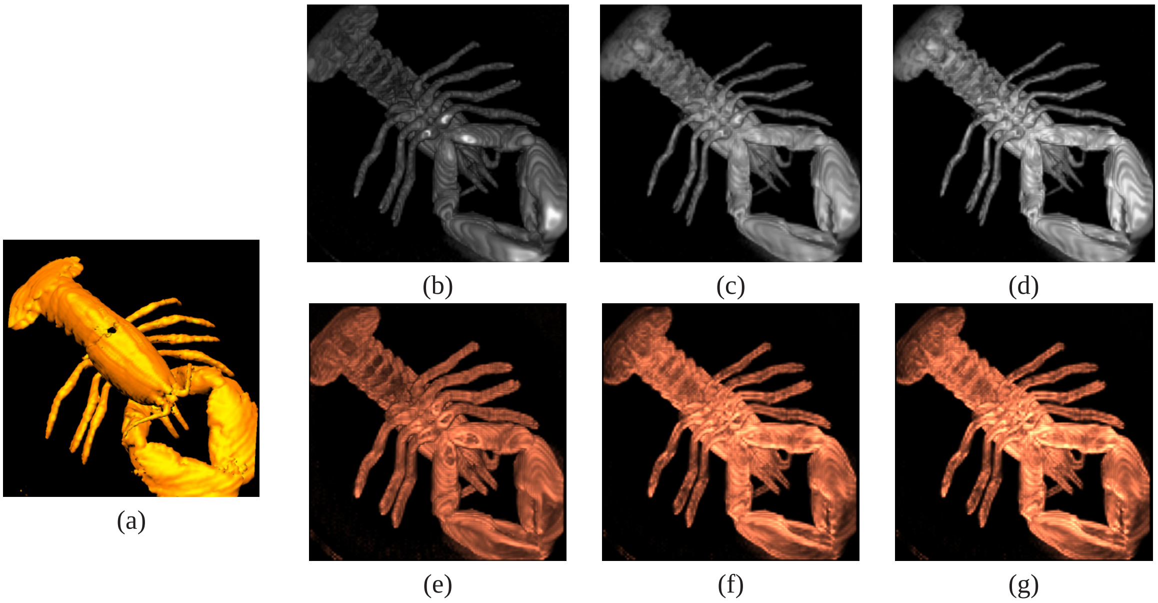


Figure 5: Renderings (lobster, 120×120×34 CT) from (a) Marching Cubes isosurfacing and (b-g) oversampled (5 times) volume ray casting, (top: using color transfer function in Equation (7), bottom: using color transfer function in Equation (8)), (b, e) without opacity correction, (c, f) with Lacroute/Lichtenbelt et al.'s opacity correction, and (d, g) with new opacity correction.

our new opacity correction produces intensity composition in which the second sample has little impact on the intensity composition when the first sample has a low value and the second sample has a high value. However, as shown in Fig. 3 (e), the prior opacity correction over-composited the second sample value which resulted in a higher composited intensity for the same case. Moreover, when the second sample has a zero value, the resultant opacities within a voxel should be linearly changed as the first sample values vary linearly. Fig. 3 (c) shows that our new opacity correction produces more linear resultant opacities when the first sample values vary linearly and the second samples have zero values. (However, Lacroute/Lichtenbelt et al.'s correction produces non-linear resultant opacities for the same cases in Fig. 3 (b).)

4.1.2 Real Data Tests

We have examined the behaviors of different corrections by comparing the resultant opacities and composited intensities within a voxel using the sample values extracted from real volumetric datasets.

Fig. 4 shows a typical example of opacity and intensity compositions in which we can observe the behaviors of different oversamplings within a voxel. In the figures, the rays, from top to bottom, are for unit-sampling, oversampling without correction, oversampling with Lacroute/Lichtenbelt et al.'s correction, and oversampling with the new correction. The $'\times'$ marks represent the sample points on each ray. The sample value of the unit-sampling was 67.0 and the sample values of the oversampling from left to right were 67.0, 69.2, 73.4, 78.6, and 84.8. The resultant opacities and composited intensities using these sample values are shown in the figure. In this example, the uncorrected oversampling produced over-composited intensity which was caused by the over-composited opacities.

Based on empirical tests on several real datasets (enumeration omitted here due to space limits), we conclude the following three behaviors of different corrections within a voxel: (1) uncorrected oversampling always over-composites the opacities and the intensities and (2) both Lacroute/Lichtenbelt et al.'s opacity correction and our new opacity correction limit the false over-composition of opacities and intensities.

4.2 Differences in Renderings

Next, we compare the quality of the renderings produced by the new and the existing opacity corrections. We consider real datasets using visual comparison. We also quantitatively compare the accuracy of different opacity corrections versus a gold standard. Since we do not know the analytical distribution of values for the real datasets, our gold standard rendering considers an analytical function. (Since the unit-sampled rendering is only a discrete approximation (i.e., Riemann sum approximation) to a continuous light integral, the unit-sampling renderings are not suitable to be used as the gold standard of the opacity-corrected oversamplings.) Thus, we have used the Marschner and Lobb analyti-

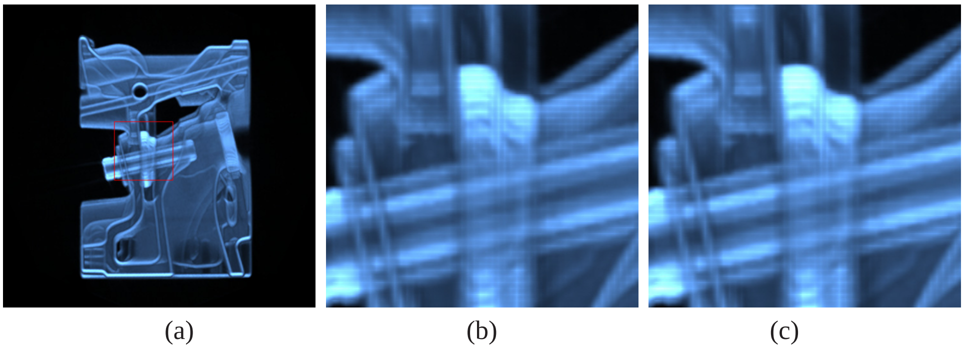


Figure 6: Comparison of opacity correction results. (a) Unit-sampled rendering with a region highlighted, (b) Lacroute/Lichtenbelt et al.'s correction for 5 times oversampling in the highlighted region, (c) new opacity correction for 5 times oversampling in the highlighted region.

cal function [Mar94] as the basis of a gold standard to compare the oversampled renderings.

First we discuss the real data results. We have applied the new and the prior (Lacroute [Lac95]/Lichtenbelt et al. [Lic98]) opacity corrections to 40 real volumetric datasets. Some example renderings are shown in Figs. 1, 2, and 5. These figures show the renderings from Marching Cubes isosurfacing (i.e., (a) in each figure) and from oversampled VRC without opacity correction, with the prior opacity correction, and with our new opacity correction for three of the real datasets. For the VRC renderings, all of the figures used shading based on the color transfer function in Equation (8) except the figures in Figs. 5 (b)-(d), which used the gray-scaled color transfer function in Equation (7). All the oversampled renderings without opacity correction (i.e., Fig. 1 (b), Fig. 2 (b), and Figs. 5 (b) and (f)) suffered from over-composited opacity. The oversampling without any opacity correction tends to become darker or brighter when the samples from the front slices have low values or high values, respectively. The prior opacity correction and our new opacity correction tend to produce rendering results (i.e., Figs. 1 (c) and (d), Figs. 2 (c) and (d), and Figs. 5 (c) and (d), and (g) and (f)) that are similar. Fig. 6 illustrates how close the oversampled renderings are using the prior opacity correction and the new correction. Fig. 6 (b) and (c) show the zoomed-in renderings (of the region marked by a red box in Fig. 6 (a)) with the prior opacity correction and with the new correction. As seen in the figures, the difference in the oversampled renderings is very small.

Fig. 7 shows the differences of ray casting renderings for the Marschner and Lobb function. Fig. 7 (a) shows the rendering that has been analytically computed, Fig. 7 (b)-(d) show the five times oversampled rendering without correction, with Lacroute/Lichtenbelt et al.'s correction, and with the new correction, respectively. The average pixel-by-pixel difference between Fig. 7 (a) and Fig. 7 (b)-(d) were 0.1200, 0.0895, and 0.0894, respectively. Thus, the results may be considered to be comparable.

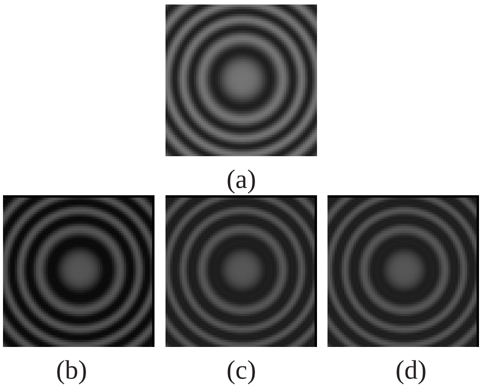


Figure 7: Ray cast renderings (Marschner-Lobb function, 64×64×64) for (a) analytical integration and (b-g) oversamplings (5 times): (b) without opacity correction, (c) with existing opacity correction, and (d) with new opacity correction.

4.3 Processing Times

While the oversampled renderings of both opacity corrections were similar when using the color transfer function in Equation (7), the processing time with the new opacity correction was faster than the time for the Lacroute/Lichtenbelt et al.'s correction. One source of the speedup is that the inverse matrix of p_0, p_1, and p_2 in Equation (15) can be computed once and be continuously re-used. In addition, each opacity correction factor p is re-used N times within a voxel. The computational speed advantage of the approach proposed here is the chief reason to use the new approach when using the color transfer function in Equation (7).

Some shortcomings of the new correction are as follows: (1) the inverse matrix of p_0, p_1, and p_2 in Equation (15) has to be re-computed when the sampling rate changes, and (2) when a lookup table is used, the dimension of the lookup table depends on the sampling rate and one lookup table for every sampling rate has to be pre-computed.

Table 1 shows three measures (i.e., maximum, average, and minimum) of the new opacity correction's speedup over the Lichtenbelt et al.'s correction for the 40 real datasets when oversampling 5 times. For the speedup of the opacity correction itself (shown in Table 1's upper row), all non-correction processing steps (e.g., interpolation of sample values and ray compositions) are ignored; only the processing time for the opacity correction itself is measured. On average, the new opacity correction has a speedup of about 12.36 when considering only the correction itself. In Table 1's bottom row, the overall speedup for the new opacity-corrected VRC rendering (versus VRC with correction by the Lacroute/Lichtenbelt et al.'s method) is shown; on average, the new approach allows an overall speedup in VRC of about 1.85. We have found that the speedup tends to get higher as the oversampling rate increases.

$\times$ 5	Max.	Avg.	Min.
Opacity Correction	14.724	12.359	6.785
Overall	1.997	1.846	1.767

Table 1: Speedup of new correction versus Lacroute/Lichtenbelt et al.'s correction for 40 datasets, for 5 times oversampling.

4.4 Approximation Error

The use of approximation in the new approach's quadratic curve fitting in Equation (13) is a source of error. We performed one experiment to evaluate the error. It involved measuring the absolute differences between the true opacity correction factor and the approximated opacity correction factor using the (120x120x34) lobster dataset shown in Fig. 5. The average fitting error for five times oversampling for one ray-casting on the dataset was 0.0031 and the standard deviation was 0.012. In addition, only 0.09% of all the fitting errors exceeded 0.1 in magnitude.

5 CONCLUSION

In this paper, a new opacity correction method for oversampled volume ray casting was described. The new correction method is a generalized voxel-by-voxel approach which avoids the assumption of dataset homogeneity. The new correction limits the false increase in opacity from over-composition of the opacity and it is faster than the only existing opacity correction. The new method can be applied about 12 times faster than the existing approach which yields an overall speedup in VRC of about 1.85.

In the future, we intend to work on further improvements to the speed and accuracy of opacity correction.

ACKNOWLEDGMENT

The work reported here was strengthened by the suggestions of the reviewers.

REFERENCES

[Kle05] Klein, T., Stegmaier, M., Stegmaier, S., and Ertl, T., Exploiting Frame-to-frame Coherence for Accelerating High-quality Volume Raycasting on Graphics Hardware, Proc., Visualization '05, Minneapolis, pp. 223-230, 2005.

[Lac95] Lacroute, P., Fast Volume Rendering Using a Shear-Warp Factorization of the Viewing Transformation, Doctoral Dissertation (Technique Report CSL-95-678), Stanford University, 1995.

[Lev88] Levoy, M., Display of Surfaces from Volume Data, IEEE Computer Graphics and Applications, Vol. 5 (3), pp. 29-37, 1988.

[Lic98] Lichtenbelt, B., Crane, R., and Naqvi, S., Introduction to Volume Rendering, Prentice Hall, Upper Saddle River, NJ, 1998.

[Mar94] Marschner, S., and Lobb, R., An Evaluation of Reconstruction Filters for Volume Renderings, Proc., Visualization '94, Washington, DC, pp. 100-107, 1994.

[Mor02] Mora, B., Jessel, J.-P., and Caubet, R., A New Object-order Ray-casting Algorithm, Proc., Visualization '02, Boston, pp. 203-210, 2002.

[Pfi05] Pfister, H., Hardware-Accelerated Volume Rendering, The Visualization Handbook ed. by C. Hansen and C. Johnson, Elsevier, New York, pp. 229-258, 2005.

[Sch03] Schulze, J.P., Kraus, M., Lang, U., and Ertl, T., Integrating Pre-Integration into the Shear-Warp Algorithm, Proc., Third Int'l Workshop on Vol. Graphics, Tokyo, pp. 109-118, 2003.

[Swa97] Swan II, J.E., Mueller, K., Moller, T., Shareef, N., Crawfis, R., and Yagel, R., An Anti-Aliasing Technique for Splatting, Proc., Visualization '97, pp. 197-204, 1997.

[Wei03] Weiler, M., Kraus, M., Merz, M., and Ertl, T., Hardware-based Ray Casting for Tetrahedral Meshes, Proc., Visualization '03, Seattle, pp. 333-340, 2003.

[Wei00] Weiler, M., Westermann, R., Hansen, C., Zimmerman, K., and Ertl, T., Level-Of-Detail Volume Rendering via 3D Textures, Proc., 2000 IEEE Symp. on Vol. Visualization, Salt Lake City, pp. 7-13, 2000.