Guidance on the Selection of Central Difference Method Accuracy in Volume Rendering

Kazuhiro Nagai[1] and Paul Rosen[2]

[1] University of Utah
[2] University of South Florida

Abstract. In many applications, such as medical diagnosis, correctness of volume rendered images is very important. The most commonly used method for gradient calculation in these volume renderings is the Central Difference Method (CDM), due to its ease of implementation and fast computation. In this paper, artifacts from using CDM for gradient calculation in volume rendering are studied. Gradients are, in general, calculated by CDM with second-order accuracy, $\mathcal{O}(\Delta x^2)$. We first introduce a simple technique to find the equations for any desired order of CDM. We then compare the $\mathcal{O}(\Delta x^2)$, $\mathcal{O}(\Delta x^4)$, and $\mathcal{O}(\Delta x^6)$ accuracy versions, using the $\mathcal{O}(\Delta x^6)$ version as "ground truth". Our results show that, unsurprisingly, $\mathcal{O}(\Delta x^2)$ has a greater number of errors than $\mathcal{O}(\Delta x^4)$, with some of those errors leading to changes in the appearance of images. In addition, we found that, in our implementation, $\mathcal{O}(\Delta x^2)$ and $\mathcal{O}(\Delta x^4)$ had virtually identical computation time. Finally, we discuss conditions where the higher-order versions may in fact produce less accurate images than the standard $\mathcal{O}(\Delta x^2)$. From these results, we provide guidance to software developers on choosing the appropriate CDM, based upon their use case.

1 Introduction

Direct volume rendering is one of the most powerful and widely used visualization techniques available for understanding 3D data. Users of these visualizations trust the images they see are true to the underlying phenomena being studied. However, artifacts from every stage of the volume rendering pipeline can potentially create inaccurate images. This inaccuracy may have severe consequences, such as failures to diagnose disease [1]. One important volume rendering pipeline stage is the calculation of gradients. These values are critical because the gradient directions are used to light the volume, and the magnitudes of these gradients are used for classification in multidimensional transfer functions.

When studying gradients, there are many potential types of artifacts. The majority of studies have evaluated artifacts from interpolation of gradients and developed new techniques to minimize these errors. Examples include: calculating gradients from analytical derivatives of filtering equations in order to produce fewer artifacts [2]; using interpolations that produce fewer artifacts than the analytical method [3]; and using global illumination in cases where the magnitude of gradient is too small to accurately calculate a direction, such as in homogeneous materials cases [4].

Although a number of advanced methods are available for calculating the gradient field, far and away, the most popular approach is using Central Difference Method (CDM) with second-order accuracy, $\mathcal{O}(\Delta x^2)$. To our knowledge, no existing work has considered artifacts coming from gradient values calculated using CDM[3].

The CDM is based upon Taylor Series expansion that enables approximating the first derivative of a scalar field with some bounded error, in this case $\mathcal{O}(\Delta x^2)$. Although the error within the CDM is well known, little has been done to measure its influence on the quality and correctness of volume rendered images. To address this shortcoming in the literature, we investigate how this albeit small error may impact the output image. We compare the standard CDM approach with more accurate versions of the gradient calculated using the Taylor Series, those with errors of $\mathcal{O}(\Delta x^4)$ and $\mathcal{O}(\Delta x^6)$. In particular, we look at variations in gradient direction and magnitude in a variety of publicly available datasets. Finally, we provide guidance for volume renderer designers, such that they can know when it is appropriate to select an alternative approach.

2 Prior Work

Volume rendering consists of a wide variety of operations, including sampling, filtering, classification, shading, and integration [6]. Each operation introduces artifacts, which may decrease the correctness and/or quality of images. There are various studies of artifacts in each stage of volume rendering.

In sampling, wood-grain artifacts are found with low sampling rate. According to Nyquist-Shannon sampling theorem, the original continuous data can be reconstructed from digital data if sampling rates are twice the highest frequency [7]. However, implementation of Nyquist-Shannon sampling theorem in volume rendering reduces the speed of interaction (i.e. lower frame rate). Adaptive Sampling [8] or stochastic jittering [9] are used in practice.

The filtering stage interpolates between multiple volume functions, usually through bilinear or trilinear interpolation. Using interpolation functions with C^1-continuity produces better images. Examples include B-spline interpolation [10], Catmull-Rom splines [11], and texture-based convolution.

The classification stage calculates the proper RGBA values by applying a transfer function. The main error created in this stage is determined by whether the transfer function is applied before or after interpolation. Generally speaking, post-interpolation is better than pre-interpolation [12].

In the shading stage, the gradient is the main factor producing artifacts. The main source of error comes from the interpolation of gradients. A common source of problems is the case of low magnitude gradients, where interpolation produces noisy output. For precomputed normalized gradients, interpolation may produce non-normalized gradients. These can be easily renormalized on-the-fly. In addition, the precomputed method may introduce quantization errors by storing the gradients in 8-bits texture channels. Individually, these quantization errors are small. However, when accumulated, these errors may result in wood-grain artifacts.

[3] Usman et al. [5] use CDM with $\mathcal{O}(\Delta x^4)$ as a standard to evaluate their calculations of gradients, but they did not establish a justification for the use of $\mathcal{O}(\Delta x^4)$ instead of $\mathcal{O}(\Delta x^2)$.

There are a variety of methods to calculate gradients, beyond just CDM, such as kernel filter or other techniques mentioned previously. For a variety of reasons, $\mathcal{O}(\Delta x^2)$ CDM is still a common choice to calculate gradients in volume renderers. As far as we know, no one has published research to evaluate artifacts coming from gradients calculated using $\mathcal{O}(\Delta x^2)$ CDM.

One important premise on the study of artifacts in volume rendering is that no single definition of quality exists. Stochastic jittering is a good example. This technique increases the uncertainty by using random noise to erase wood-grain artifacts. The images have high visual quality, but also they may no longer be true to the original data. The result is that many techniques have been published on the premise of "better image quality" without clearly stating whether the basis is aesthetic, correctness, or both.

3 Central Difference Method (CDM)

The finite different stencil associated with CDM is calculated using the Taylor Series of $f(x+\Delta x)$, $f(x)$, and $f(x-\Delta x)$.

$$f(x+\Delta x) = f(x) + f'(x)\Delta x + \frac{f''(x)\Delta x^2}{2!} + \mathcal{O}(\Delta x^3) \tag{1}$$

$$f(x) = f(x) \tag{2}$$

$$f(x-\Delta x) = f(x) - f'(x)\Delta x + \frac{f''(x)\Delta x^2}{2!} + \mathcal{O}(\Delta x^3) \tag{3}$$

In these equations, x represents the current voxel, Δx represents the distance between voxels, $f(x)$ represents the function value, and $f'(x)$ and $f''(x)$ represent the first and second derivatives, respectively. By solving these equations for $f'(x)$, we have found the $\mathcal{O}(\Delta x^2)$ CDM.

$$f'(x) = \frac{f(x+\Delta x) - f(x-\Delta x)}{2} + \mathcal{O}(\Delta x^2) \tag{4}$$

We can also place equations (1), (2), and (3) into a matrix.

$$\begin{bmatrix} f(x+\Delta x) \\ f(x) \\ f(x-\Delta x) \end{bmatrix} = \begin{bmatrix} 1 & \Delta x & \frac{\Delta x^2}{2!} \\ 1 & 0 & 0 \\ 1 & -\Delta x & \frac{\Delta x^2}{2!} \end{bmatrix} \begin{bmatrix} f(x) \\ f'(x) \\ f''(x) \end{bmatrix} \tag{5}$$

To get the CDM equations, the matrix equation can be solved ($A^{-1}B = A^{-1}AX$) and $f'(x)$ selected.

This approach can be generalized to work for both boundary (non-symmetric) cases and higher-order cases by selecting additional input function values (e.g. $f(x+2\Delta x)$, $f(x-2\Delta x)$, etc.) and expanding the Taylor series further, such that the matrix is square.

4 Experiments

For this study, we used a variety of volume types from The Volume Library website [13]. All experiments were run on a Windows-based laptop with an Intel Core i7 processor and NVIDIA GTX 880m GPU. The experiments were performed using a homegrown volume renderer with the configuration shown in Table 1.

Table 1: Configuration of Volume Renderer

Sampling Rate:	Greater than twice per voxel
Filtering:	Trilinear
Classification:	Pre-interpolation
Shading:	Pre-computed gradients renormalized in fragment shader
Integration:	8-bits used to store each fragment value.

These conditions are relatively common ones. We do place some considerations towards minimizing artifacts from other pipeline stages. For example, the sampling rate is so high that trilinear interpolations will not show wood-gain artifacts. Additionally, using pre-interpolation for classification avoids generating high frequency artifacts. Finally, although gradients are precomputed, interpolated gradients are normalized in the fragment shader.

For our results, we make an assumption that the $\mathscr{O}(\Delta x^6)$ gradient is our ground truth (i.e. no errors). We conducted several experiments to determine the influence of various levels of accuracy for gradients, focusing on magnitude and direction separately. This is because they tend to be used in different stages of the pipeline (i.e. magnitude in classification, direction in lighting), rarely together.

Throughout, the absolute error (d_m) and relative error (d_m^r) in magnitude is calculated by:

$$d_m = |A| - |B|, \tag{6}$$

$$d_m^r = \frac{|A| - |B|}{|A|}, \tag{7}$$

where A is the gradient of the $\mathscr{O}(\Delta x^6)$ voxel and B is the gradient of the same voxel with either $\mathscr{O}(\Delta x^2)$ or $\mathscr{O}(\Delta x^4)$.

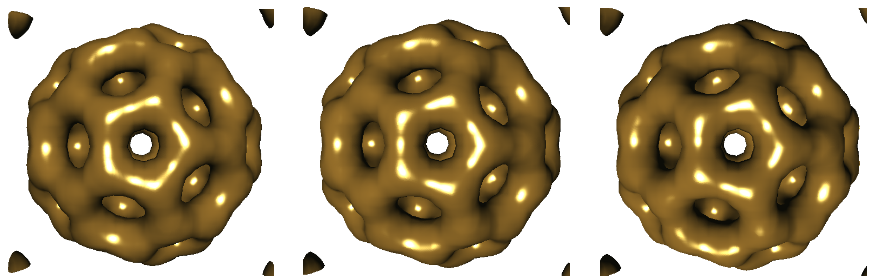
The absolute error in the direction of gradients (d_d) is calculated by:

$$d_d = 1 - \frac{A \cdot B}{|A||B|}. \tag{8}$$

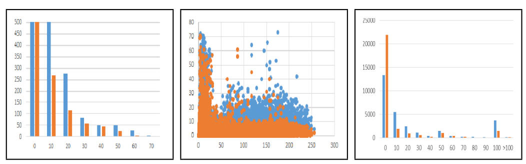
Since gradient direction is a normalized vector, relative error for gradient direction cannot be calculate.

5 Results

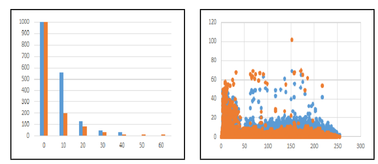
We tested all of pvm files we could obtain from The Volume Library website [13], with the exception of 16-bit XMasTree.pvm and 8-bit Porsche.pvm, which we were unable to load. In total, we tested over 30 datasets. Due to space limitations, we discuss only a few interesting cases here.



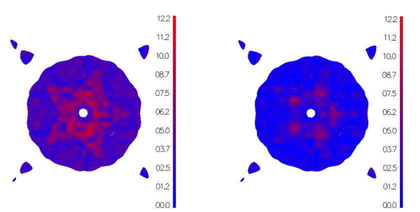
(a) Shown with $\mathcal{O}(\Delta x^2)$ (left), $\mathcal{O}(\Delta x^4)$ (middle), and $\mathcal{O}(\Delta x^6)$ (right), the first difference is in the shape of the specularity. Secondarily, the $\mathcal{O}(\Delta x^4)$ and $\mathcal{O}(\Delta x^6)$ have darker illumination in the holes of the bucky ball.



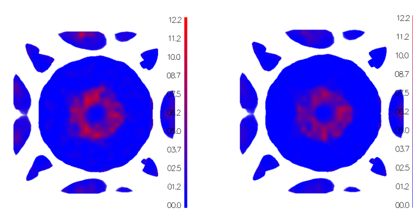
(b) Plots of the error in gradient magnitude for each voxel with $\mathcal{O}(\Delta x^2)$ (blue) and $\mathcal{O}(\Delta x^4)$ (orange). The histogram (left) shows the number of voxels with at different absolute error levels. The center chart plots the volume function value (horizontally) against the absolute error (vertically). The right histogram plots relative error (%) horizontally. Errors are calculated using Equation (6) and (7).



(c) Plots of the error in gradient direction for each voxel with $\mathcal{O}(\Delta x^2)$ (blue) and $\mathcal{O}(\Delta x^4)$ (orange). The histogram (left) shows the number of voxels with at different error levels. The right chart plots the volume function value (horizontally) against the error (vertically). Errors are calculated using Equation (8).



(d) Images showing error in the gradient magnitudes with $\mathcal{O}(\Delta x^2)$ (left) and $\mathcal{O}(\Delta x^4)$ (right).



(e) Images showing error in the gradient directions with $\mathcal{O}(\Delta x^2)$ (left) and $\mathcal{O}(\Delta x^4)$ (right).

Fig. 1: Experimental results using Bucky.pvm.

5.1 Errors in Gradient Magnitude and Direction

Our results show that the gradients of each voxel calculated from $\mathcal{O}(\Delta x^2)$ have greater error than those calculated using $\mathcal{O}(\Delta x^4)$. The comparison of the magnitude and direction can be seen in Fig. 1. In particularly, differences can be seen in the results of lighting in Fig. 1a.

Gradient Magnitude Fig. 1b shows the magnitude error for each voxel using $\mathcal{O}(\Delta x^2)$ and $\mathcal{O}(\Delta x^4)$. The plots shows $\mathcal{O}(\Delta x^2)$ has greater error than $\mathcal{O}(\Delta x^4)$. For example, 6% of voxels have 0.78% error in $\mathcal{O}(\Delta x^2)$. On the other hand, using $\mathcal{O}(\Delta x^4)$, only 1% of voxels have the same error (see Fig. 1b, center).

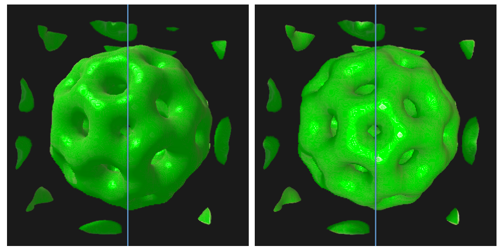


Fig. 2: The left half of each image is calculated by $\mathcal{O}(\Delta x^2)$, while the right uses $\mathcal{O}(\Delta x^4)$. The lighting in the left image is similar, but in the right image, the right half shows specular spots, which are not seen in the left half.

Fig. 1d shows the distributions of error on the image. Of particular note is the the red-colored pentagon shape on the surface of the bucky ball in $\mathcal{O}(\Delta x^2)$. In $\mathcal{O}(\Delta x^4)$, small red spots around the center hole of the bucky ball are observed.

Gradient Direction Fig. 1c shows the error in gradient direction $\mathcal{O}(\Delta x^2)$ and $\mathcal{O}(\Delta x^4)$, both showing greater error in $\mathcal{O}(\Delta x^2)$ than $\mathcal{O}(\Delta x^4)$.

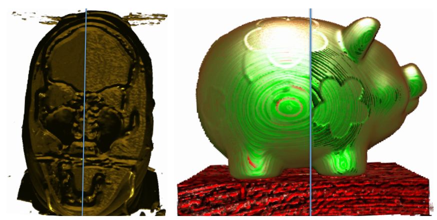


Fig. 3: Images of VisMale.pvm (left) and Pig.pvm (right) where the left half of each is calculated using the $\mathcal{O}(\Delta x^2)$ CDM and the right half uses $\mathcal{O}(\Delta x^4)$ CDM. VisMale.pvm demonstrates what happens with homogenous materials, while Pig.pvm demonstrates what can happen with high frequency edges.

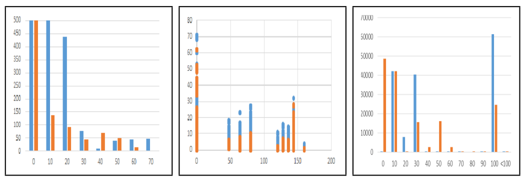
Fig. 1e displays the distribution of errors in the gradient direction with $\mathcal{O}(\Delta x^2)$ (left) the $\mathcal{O}(\Delta x^4)$ and $\mathcal{O}(\Delta x^4)$ (right). While there is some difference between $\mathcal{O}(\Delta x^2)$ and $\mathcal{O}(\Delta x^4)$, the difference is not as obvious as it was for the gradient magnitude.

Fig. 2 shows rendered images of Bucky.pvm. The left half of each use $\mathcal{O}(\Delta x^2)$ and the right uses $\mathcal{O}(\Delta x^4)$. In the right image, the white specular highlights are observed. These are only visible under certain lighting directions and not visible under $\mathcal{O}(\Delta x^2)$. This represents a small, but nonetheless important, difference in images produced with less accurate gradients.

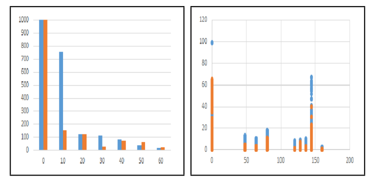
5.2 Homogeneous Materials

Materials that are more or less homogeneous (i.e. they have consistent values) are known to be problematic for CDM. Essentially, if all of the voxels surrounding me have approximately the same value, my gradient magnitude is zero and direction is underconstrained. In the $\mathcal{O}(\Delta x^2)$ CDM, the stencil size is quite small, limited to 6 surrounding voxels. By using the $\mathcal{O}(\Delta x^4)$ or higher CDM, the stencil footprint increases, and the voxel gradients becomes more stable.

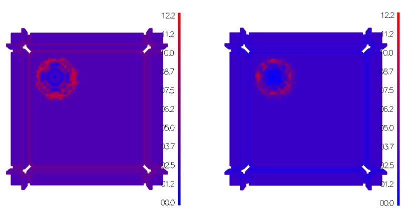
(a) Images using $\mathscr{O}(\Delta x^2)$, $\mathscr{O}(\Delta x^4)$, and $\mathscr{O}(\Delta x^6)$, respectively. The images show greater ringing artifacts for CDMs that should produce more accurate results (i.e. $\mathscr{O}(\Delta x^6)$).



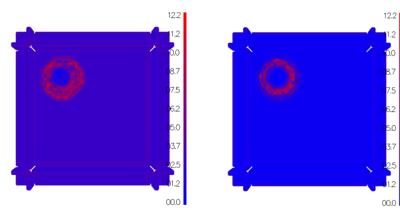
(b) Plots of the error in gradient magnitude for each voxel with $\mathscr{O}(\Delta x^2)$ (blue) and $\mathscr{O}(\Delta x^4)$ (orange). The histogram (left) shows the number of voxels with at different absolute error levels. The center chart plots the volume function value (horizontally) against the absolute error (vertically). The right histogram plots relative error (%) horizontally. Errors are calculated using Equation (6) and (7).



(c) Plots of the error in gradient direction for each voxel with $\mathscr{O}(\Delta x^2)$ (blue) and $\mathscr{O}(\Delta x^4)$ (orange). The histogram (left) shows the number of voxels with at different error levels. The right chart plots the volume function value (horizontally) against the error (vertically). Errors are calculated using Equation (8).



(d) Measurement of error in gradient magnitude for $\mathscr{O}(\Delta x^2)$ (left) and $\mathscr{O}(\Delta x^4)$ (right).



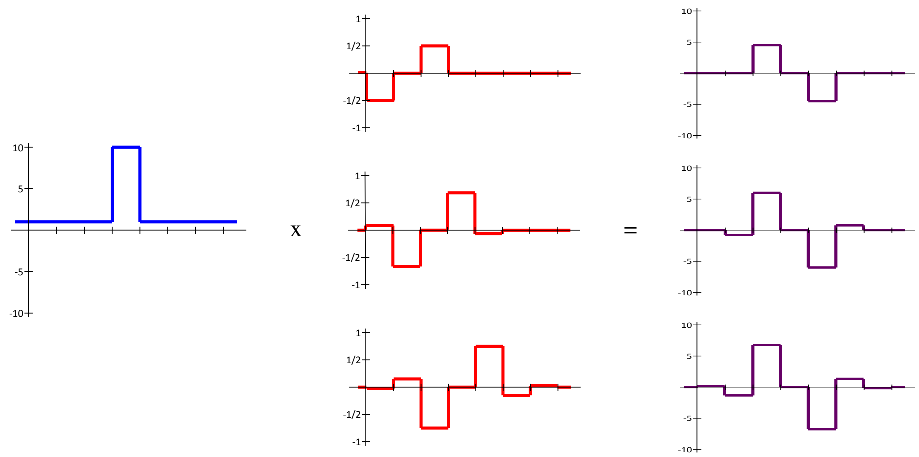
(e) Measurement of error in gradient direction for $\mathscr{O}(\Delta x^2)$ (left) and $\mathscr{O}(\Delta x^4)$ (right).

Fig. 4: Experimental results using Cross.pvm

Fig. 3 (left) is an example of this. The left half of the image shows no texture in the brain with $\mathscr{O}(\Delta x^2)$. The right half, with $\mathscr{O}(\Delta x^4)$ begins to show some of the texture detail. Its important to recognize, however, that although the gradient is better constrained, that does not mean the gradient is true to the underlying data. The real problem here is that the data are either truly homogeneous or undersampled.

5.3 High Frequency Materials and Ringing Artifacts

Although theoretically more accurate, higher-order CDM does not always lead to an accurate. High frequency surfaces, such as those seen in Fig. 3 (right) or Fig. 4a, are just such failure cases. Here, the large sampling area of the $\mathscr{O}(\Delta x^4)$ and $\mathscr{O}(\Delta x^6)$ CDMs have



(a) Input Function (b) Stencil (c) Output Function

Fig. 5: A simple rectangular input function (left), representing a high-frequency feature, has $\mathcal{O}(\Delta x^2)$ (top), $\mathcal{O}(\Delta x^4)$ (middle), and $\mathcal{O}(\Delta x^6)$ (bottom) stencils applied (center). The resulting output (right) shows larger spread for the higher accuracy stencils. Since these output represent the gradient in one direction, the larger footprints lead to the larger ringing for those stencils.

led to a ringing artifacts. Figure 5 demonstrates how this occurs. The larger footprint of the higher order stencils results in capturing the feature in the gradient calculation at further distance.

Fig. 4b shows the error in the gradient magnitude for $\mathcal{O}(\Delta x^2)$ and $\mathcal{O}(\Delta x^4)$ CDM versions. Both plots show the error in the gradient magnitude of $\mathcal{O}(\Delta x^2)$ is larger than that of $\mathcal{O}(\Delta x^4)$—the same result as Bucky.pvm. However, the assumption that the $\mathcal{O}(\Delta x^6)$ CDM is the ground truth lead us to this false conclusion. Fig. 4d shows the distribution of errors in image space. As we can see, the errors are occurring in regions where the the gradients of the high frequency surface voxels are not parallel. This is obvious in the round dent, but it also occurs in the seam where the perpendicular surfaces come together.

Fig. 4c shows the error for gradient directions with $\mathcal{O}(\Delta x^2)$ and $\mathcal{O}(\Delta x^4)$ CDM on each voxel. Both plots show $\mathcal{O}(\Delta x^2)$ has greater error than $\mathcal{O}(\Delta x^4)$. Fig. 4e displays the distributions of errors in the direction of gradients with $\mathcal{O}(\Delta x^2)$ (left) and $\mathcal{O}(\Delta x^4)$ (right). The $\mathcal{O}(\Delta x^2)$ shows darker than $\mathcal{O}(\Delta x^4)$. However, this $\mathcal{O}(\Delta x^2)$ does not have the red lines, which we could see in the error for the gradient magnitude.

The bottom of the right images of Fig. 4a shows the bottom of the right side of Cross.pvm image. $\mathcal{O}(\Delta x^4)$ and $\mathcal{O}(\Delta x^6)$ show extra lines that do not appear in $\mathcal{O}(\Delta x^2)$. This ringing artifact is well studied [14]. In this case, the higher order version contains more artifacts than its lower order counterpart.

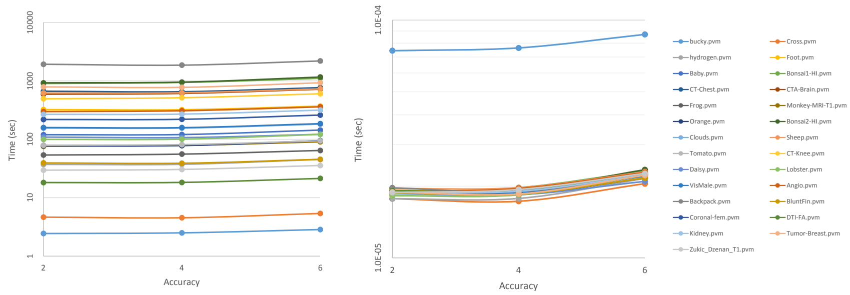


Fig. 6: The time vs. $\mathscr{O}(\Delta x^2)$, $\mathscr{O}(\Delta x^4)$ and $\mathscr{O}(\Delta x^6)$. The left shows the time to calculate 100 iterations. The right is the same result per voxel, normalized by dividing the time by the number of voxels.

5.4 Times to compute CDM with $\mathscr{O}(\Delta x^2)$, $\mathscr{O}(\Delta x^4)$, and $\mathscr{O}(\Delta x^6)$

The time to compute gradients for all datasets are presented in Fig. 6. In most cases, the time to calculate $\mathscr{O}(\Delta x^4)$ is about the same as the $\mathscr{O}(\Delta x^2)$. In fact, some of $\mathscr{O}(\Delta x^4)$ cases are less than $\mathscr{O}(\Delta x^2)$. The time to calculate $\mathscr{O}(\Delta x^6)$ is only about 20% more than $\mathscr{O}(\Delta x^2)$. We speculate that this small variation has something to do with memory locality, since stencil operations tend to be memory bound. This is due to the relatively small number of arithmetic operations performed relative to the large number of memory operations in inner product calculations. For these types of memory bound operations, cache performance is king. Larger stencils result in greater numbers of conflict misses.

6 Discussion & Conclusions

With these results, we give some brief guidance on selecting what type of CDM to use in the real-world.

My CDM choice does matter. Although many images were similar under multiple orders of CDM, there were small, possibly important differences, depending upon your usage scenario. When scientific data is used for decision making, it is critical to keep the images true to the data.

Higher-accuracy is (usually) better because it theoretically produces images true to their underlying data (i.e. with lower error). However, one has to be cautious when blindly applying either a low or high order CDM as the practice does not always live up to theory. The choice may in fact depend upon the data.

Higher-accuracy costs more time, but it does not cost nearly as much as one might expect. If the algorithm is well designed to take advantage of memory locality, a higher order calculation may only suffer a small onetime overhead, in the case of static datasets. If looking to improve performance, focus on aspects that impact the performance of rendering per frame, such as sampling rate.

$\mathcal{O}(\Delta x^4)$ *represents a good compromise* between fast calculation, accuracy, and small ringing artifacts. While renderers should really be adapting to their data, we have settled on the $\mathcal{O}(\Delta x^4)$ CDM as a better default than the commonly selected $\mathcal{O}(\Delta x^2)$ version. This finding supports the decision of Usman et al. [5] to use CDM with $\mathcal{O}(\Delta x^4)$ to evaluate their calculations of gradients.

In conclusion, when it comes to putting together a quality volume renderer, many design decisions must be made. It is clear from our results that, while not the most critical element, the choice of gradient calculation mechanism can have an important impact on the final rendered image. As such we recommend more thought be put into the choice, and, at the very least, volume renderer designers should chose $\mathcal{O}(\Delta x^4)$ as their default method.

References

1. Lundstrm, C., Ljung, P., Persson, A., Ynnerman, A.: Uncertainty visualization in medical volume rendering using probabilistic animation. IEEE Trans. Vis. Comput. Graph. **13** (2007) 1648–1655
2. Bentum, M.J., Lichtenbelt, B.B.A., Malzbender, T.: Frequency analysis of gradient estimators in volume rendering. IEEE Transactions on Visualization and Computer Graphics **2** (1996) 242–254
3. Hossain, Z., Alim, U.R., Möller, T.: Toward high-quality gradient estimation on regular lattices. IEEE Trans. Vis. Comput. Graph **17** (2011) 426–439
4. 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** (2003) 150–162
5. Alim, U., Moller, T., Condat, L.: Gradient estimation revitalized. IEEE Transactions on Visualization and Computer Graphics **16** (2010) 1495–1504
6. Engel, K., Hadwiger, M., Kniss, J.M., Lefohn, A.E., Salama, C.R., Weiskopf, D.: Real-time volume graphics. In: ACM SIGGRAPH 2004 Course Notes. SIGGRAPH '04, New York, NY, USA, ACM (2004)
7. Nyquist, H.: Certain topics in telegraph transmission theory. Transactions of the AIEE **47** (1928) 617–644
8. Kraus, M., Strengert, M., Klein, T., Ertl, T.: Adaptive sampling in three dimensions for volume rendering on gpus. In Hong, S.H., Ma, K.L., eds.: APVIS, IEEE (2007) 113–120
9. Cook, R.L.: Stochastic sampling in computer graphics. ACM Trans. Graph. **5** (1986) 51–72
10. Mihajlovi, eljka Goluban, A.a.M.: Frequency domain analysis of b-spline interpolation. (1999)
11. Möller, T., Machiraju, R., Mueller, K., Yagel, R.: Classification and local error estimation of interpolation and derivative filters for volume rendering. In: 1996 Volume Visualization Symposium, IEEE (1996) 71–78
12. Engel, K., Kraus, M., Ertl, T.: High-quality pre-integrated volume rendering using hardware-accelerated pixel shading (2014)
13. Roettger, S.: The Volume Library. http://lgdv.cs.fau.de/External/vollib/ (2015) [Online; accessed 19-March-2015].
14. Marschner, S.R., Lobb, R.J.: An evaluation of reconstruction filters for volume rendering. In Bergeron, R.D., Kaufman, A.E., eds.: Proceedings of the Conference on Visualization, Los Alamitos, CA, USA, IEEE Computer Society Press (1994) 100–107