SIGRAD 2013, pp. 55-60
T. Ropinski and J. Unger (Editors)

Towards Data Centric Sampling for Volume Rendering

Stefan Lindholm[1], Daniel Jönsson[1], Hans Knutsson[2] and Anders Ynnerman[1]

[1]Scientific Visualization Group, Linköping University, Sweden
[2]IMT, Linköping University, Sweden

Abstract

We present a new method for sampling the volume rendering integral in volume raycasting where samples are correlated based on transfer function content and data set values. This has two major advantages. First, visual artifacts stemming from structured noise, such as wood grain, can be reduced. Second, we will show that the volume data does no longer need to be available during the rendering phase; a surface representation is used instead, which opens up ample opportunities for rendering of large data. We will show that the proposed sampling method gives higher quality renderings with fewer samples when compared to regular sampling in the spatial domain.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Bitmap and framebuffer operations I.3.6 [Computer Graphics]: Methodology and Techniques—Graphics data structures and data types

1. Introduction

Volume rendering is an essential and widespread technique used to explore volumetric data. Its use in areas such as medicine and engineering puts a demand on accurate visualizations while still allowing interactive exploration of the data. The majority of the techniques related to volume rendering use some form of emission/absorption model of light transport. The traditional computational approach involves a discrete Riemann sum, where the volume is sampled along viewing rays (emanating from the screen out through the scene), resulting in an iterative solution. It is of great importance that this model can be evaluated in an accurate and fast manner. An important aspect of the traditional approach is that the sample positions of neighboring rays are selected in the spatial domain with next to zero correlation in attribute space between neighboring rays. This paper investigates an alternate way to select samples to accurately and efficiently evaluate an emission/absorption model.

We propose a scheme that correlates samples in attribute space across all rays in the image. The effect is that each sample represents a given *change in function value* as opposed to a certain spatial distance. Our scheme creates correlated samples for all rays simultaneously (an operation of object space complexity) while still allowing the emission/absorption model to be evaluated independently for each ray (an operation of image space complexity). We show how the scheme is inherently data adaptive and how the content of the transfer function can be used to optimize the selection of the correlated samples. An illustration of the difference relative to the traditional approach is provided in Figure 1.

The paper is structured as follows. First we recap the popular optical model of volume rendering with an emphasis on the discretization that it typically leads to. We then outline some of the challenges, and their related work, that arise from such a discretization, both for the general case and within the context of visualization. Finally we present our approach.

2. Background And Related Work

The arguably most popular physical model of volume rendering is derived from optical models of light transport through participating mediums, as introduced by Max in [Max95]. The participating media is here modeled as "clouds" of small particles, where light that passes through such media extinguishes at a rate proportional to the number of particles per volume fraction (particle density). The continuous differential equation describing emission and absorption is the following

$$\frac{dI}{ds} = c(s)\tau(s) - \tau(s)I(s), \tag{1}$$

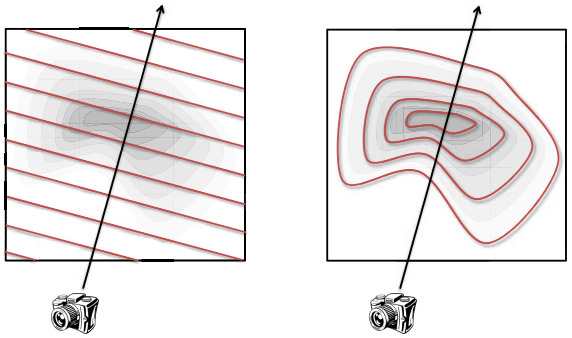

Figure 1: *Left:* Spatially correlated sampling used in standard volume rendering . *Right:* Radiometrically correlated sampling used in our ISO based volume rendering.

where s is a length parameter along a ray, $c(s)$ is the amount of light emitted, $\tau(s)$ is the amount of light absorbed and $I(s)$ is the light intensity. The solution to this differential equation is the well known *volume rendering integral* (VRI)

$$I(D) = I_0 e^{-\int_0^D \tau(t)\,\mathrm{d}t} + \int_0^D c(s)\tau(s)e^{-\int_s^D \tau(t)\,\mathrm{d}t}\,\mathrm{d}s. \quad (2)$$

The first term describes the amount of light coming from the end of the ray attenuated by the absorption in the volume. The second term describes light that is emitted and absorbed along the ray within the volume. The past twenty years have seen a large amount of work dedicated to the integral in Equation 2. A practical analytical closed-form solution has, unfortunately, proven to be elusive as such solutions tend to rely on the *erf* function, which is rather costly to evaluate. As a result, most of the literature concerns discrete numerical solutions, most notably Riemann sums of piecewise constant segments.

If the color and opacity each are assumed to be constant for each segment i, $c(s) \approx c_i$ and $\tau(s) \approx \tau_i$, then the continuous expression in Equation 2 can be approximated as

$$I(D) \approx \sum_{i=0}^{n} c_i \tau_i ds_i \prod_{j=i+1}^{n} 1 - \tau_j ds_j. \quad (3)$$

where each segment corresponds to a sample along the viewing ray. The question is how to place the samples in such a way that the VRI can be accurately and efficiently evaluated.

We will now briefly discuss the general problem of discretizing and solving Equation 2 in relation to related work on two aspects, numerical and perceptual. Many of these works address two main questions:

Where to sample? It is desirable to have the highest possible quality given as few samples as possible without overhead.

How many samples? It is desirable to determine the lowest number of samples needed to achieve an error that is not visible to the user.

The general problem: Approximation errors

Numerical approximations of an integral most often introduce errors. One way of minimizing such an error is to apply importance based sampling. Much work has been done to come up with good error metrics and even more work has been done to minimize the errors. Notable sub fields on this topic include Monte Carlo based schemes [Sha03], adaptive sampling techniques [Lev90], level of detail rendering [WWH*00] and predictive schemes [NH93].

The visualization problem: Structured noise

Differences in the approximation errors across pixels leads to noise in the rendered image. Unfortunately this noise is structured and often results in noticeable artifacts. One of the most prominent in the field of volume rendering is the so called wood grain effect. Several techniques have tackled this problem. The most notable being jittering [EHK*06], which randomly moves the start point of the ray to alleviate the visual appearance by making noise unstructured. Another partial solution is pre-integrated volume rendering, which in many cases can remove the artifacts [KE04] by implicitly super-sampling the ray. Similarly, additional samples can be enforced to correlate with sharp features in the transfer function [KHW*09] at the cost of solving a third degree polynomial between samples. Another approach to reduce visual artifacts is to include perceptual based sampling in the image plane using iterative methods and models of the human visual system [BM98].

3. A Coherent Sampling Method

Our new sampling method utilizes two important aspects of the VRI and human visual system, that a user specifies absorption through a transfer function and that structured noise is visually disturbing. We use these two realizations when answering the two questions in the previous section, where to sample and how many samples to take. We previously discussed that errors across pixels leads to visually disturbing artifacts. This is especially noticeable at sharp boundary transitions, where two consecutive samples along a ray have low and high opacity. This produces a high numerical error that becomes visually apparent because neighboring rays are likely to also have a high numerical error.

We reduce the structured noise by allowing a maximum value change between each pair of samples. This naturally leads to more samples in transitional regions and fewer samples in homogeneous regions. The key here is that the values at which samples are enforced to are global. As such, each layer of samples corresponds to an iso-surface. The surfaces are based on an importance sampling of the transfer function content as illustrated in Figure 2. Surfaces are extracted from the non-zero opacity regions in the transfer function and we are thus retrieving visually important samples that have a high impact on the result of the VRI. Furthermore,

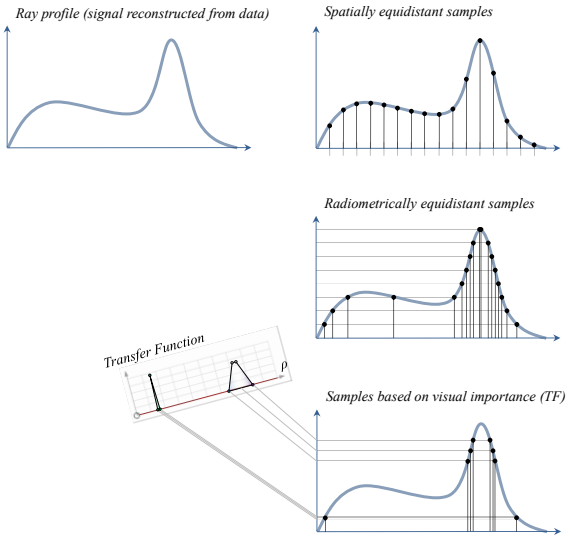

Figure 2: *Top Left:* Ray profile. *Top Right:* Samples uniformly distributed in the spatial domain (horizontal axis). *Middle Right:* Samples uniformly distributed in the attribute space (vertical axis). *Bottom Right:* Samples based on visual importance.

by extracting the surfaces at globally selected iso-values, we guarantee that neighboring rays in image space will sample the VRI in a coherent manner, and therefore also remove the wood grain artifact. Another benefit is that no samples will be placed in regions with zero opacity resulting in an efficient empty space skipping.

The second question is how to know how many samples need to be taken in order to ensure an artifact free image. Here we can utilize the knowledge that we can bound the error since the distances in both the spatial and data domain are known between each sample pair. That is, since we have generated surfaces based on values in the data domain, we know that the data variation between two surfaces are bounded by the values used to generate the surfaces. This information can be combined with the distance between the two surfaces in order to derive a maximum approximation error and thus continue to sub sample along the ray until going below the desired error. This particular aspect of the method is, however, not included in the scope of this paper, which focuses more on the former question.

We further introduce a novel data representation that allow the VRI to be evaluated without access to the original volume data. This is based on forming an assumption on how the data varies between each pair of sample points, i.e. between two surfaces. We use the assumption that the data, and thereby the transfer function response, varies linearly.

3.1. Implementation

Algorithm 1 *Coherent Sampling.*

Require:
A set of iso-values formed from the transfer function content.
Algorithm:
for all iso-values **do**
Generate proxy geometry (marching cubes)
end for
Setup per pixel linked list
for all Proxy geometries **do**
Render proxy geometry
Store entry, exit point and iso-value per pixel
end for
Sort list of per pixel fragments based on depth
Evaluate VRI using sorted fragment list as sample sequence

The overall structure of our method is described in Algorithm 1 and Figure 3. The implementation can be divided into two main parts:

A Extraction of proxy geometry, for each iso-surface, in the form of an closed manifold triangle mesh. This step is performed in object space, potentially as a pre-process.
B Management of fragments from rasterized iso-surfaces, in the form of a sorted linked list intersection points for each pixel. This step is performed in image space during rendering.

For the first part, which takes place *before* rasterization, we use a hardware implementation of the *marching cubes* algorithm, performed as a three step process:

A.1 Use *instanced rendering* to trigger a single vertex per voxel in the volume
A.2 Use a *geometry shader* to create the maching cubes triangles
A.3 Use *transform feedback* to capture the output of the geometry shader into a buffer on the GPU

For the third part, which takes place *after* rasterization, we use a GPU variant of linked lists based on A-buffers

B.1 Render the mesh representations for all iso-surfaces (to be rasterized)
B.2 Store all fragments on a per-pixel basis as linked lists in an A-buffer
B.3 Sort each linked list based on fragment depth

After the execution of B, each list of fragments can be interpreted as an ordered list of iso-surface intersections. For volume rendering, this means that the volume rendering integral can be evaluated along each ray' by looping over the list of fragments, treating each fragment as a sample. In this case, sample positions are given by the direction of the ray and the depth of each fragment, while the corresponding sample values are implicitly given by the particular iso-surface associated with each fragment.

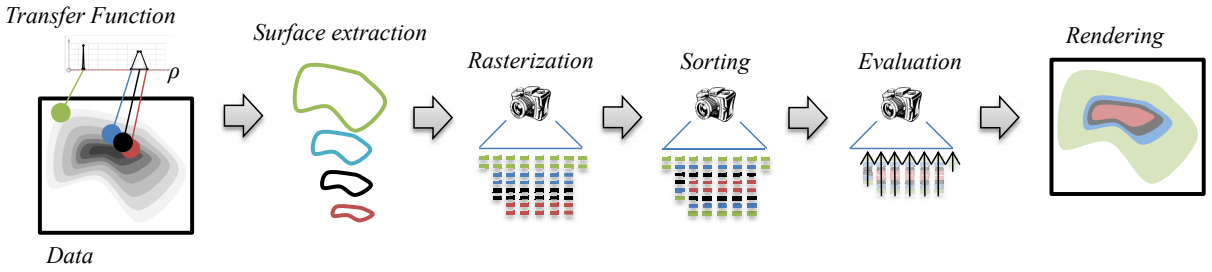

Figure 3: Our proposed solution which relies on extraction, rasterization and sorting of proxy geometry to generate samples for the evaluation of the volume rendering integral.

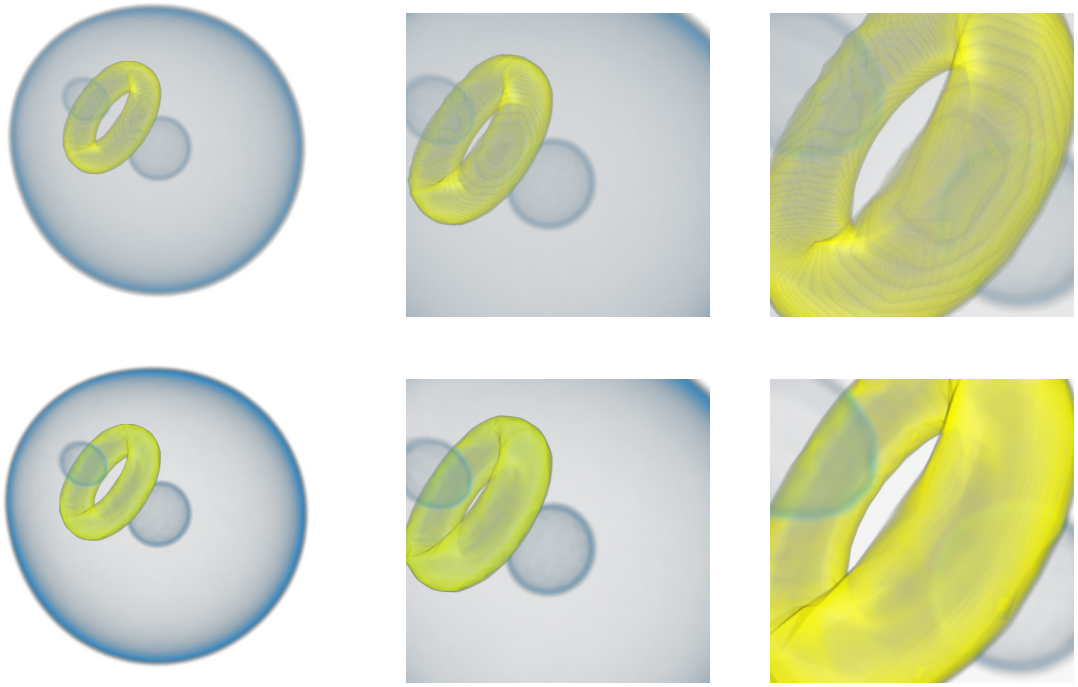

Figure 4: *Top row:* DVR of the Nucleon data set at an average of 4 samples per voxel. *Bottom row:* ISO at 26 surfaces for a total of 132k triangles. Using orthographic projection and looking from one side uniform sampling requires 164 samples per ray while our method only requires on average 30 samples per ray and still result in superior image quality.

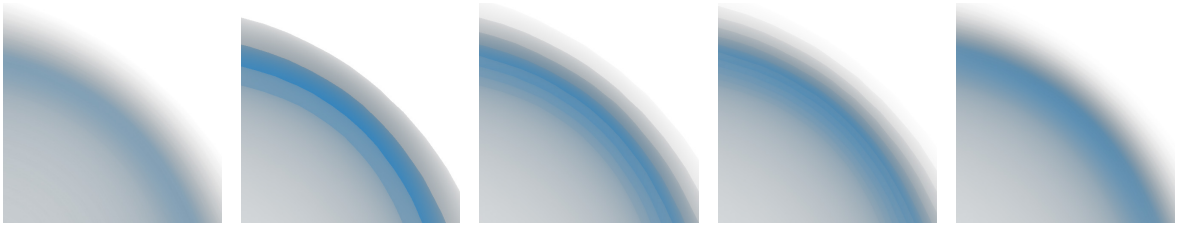

Figure 5: *Left-to-right:* DVR reference rendering of the Nucleon data set followed by ours with 5, 9, 13 and 31 surfaces (46k, 82k, 119k, 282k triangles respectively). The ISO rendering converges towards the reference with increasing number of surfaces. Convergence rate is directly dependent on transfer function width, the wider the transfer function, the more surfaces are needed for convergence.

4. Results

To evaluate the performance of the approach, we have used the a data set of a nucleon at a resolution of $41 \times 41 \times 41$ voxels. Figure 4 shows a comparison between our approach (*bottom row*) and standard volume rendering with samples distributed uniformly in the spatial domain (*top row*). The three images per row shows increasing magnification from left to right. The transfer function used consists of two separate, triangle shaped primitives (thus a total of four linear segments, see Figure 6 (top)). The image rendered with our approach utilizes one surface per segment end point in the transfer function (6 in total) with an additional 5 surfaces per linear slope (20 in total). The following table is an estimation of the number of samples used per ray (note that this number varies across the image)

White area No intersections and thus zero samples.
Blue area Intersections with 2×13 surfaces (in and out of the blue primitive).
Yellow area Intersections with $(2+2) \times 13$ surfaces (in and out of both the blue and the yellow primitives).
Extreme areas At a few locations rays pass in and out twice for each of the yellow and blue areas for a total of 8×13 surface intersections.

Note that for the majority of the rays, the average number of intersections, and thus samples that need to be evaluated and fragments that need to be sorted, is less than 30. Even the worst case scenario for this particular data and transfer function results in just above 100 samples per ray. The standard rendering was performed using an (above) average four samples per voxel for a total cost of 164 samples per ray, and still contains clearly visible wood grain artifacts.

We have also evaluated the amount of iso-surfaces needed to approximate a single linear segment of the transfer function. Figure 5 shows the same nucleon data set rendered with a single triangle primitive (two linear segments, see Figure 6 (bottom)). As the primitive is much wider, covering nearly half the data range of the volume, the spatial distance between the surfaces will be larger and therefore more surfaces are required to minimize the error than in the previous example. The most prominent visual artifact effect occurs at the edges of structures where rays that either intersect or miss a surface get different evaluations, effectively creating a visual step. More surfaces naturally makes the steps smaller such that they eventually disappear. The leftmost image was rendered with standard volume rendering at a very high sampling rate while the remaining images were rendered with the proposed method using increasing number of surfaces per transfer function segment. Note that the effects are magnified by the zoomed in camera setting and that less surfaces are necessary while viewing the entirety of the object.

Finally, derivative expressions can be included in the generation of the proxy geometry and made available during rendering. This enables surface based gradients to be used instead of volumetrically based gradients.

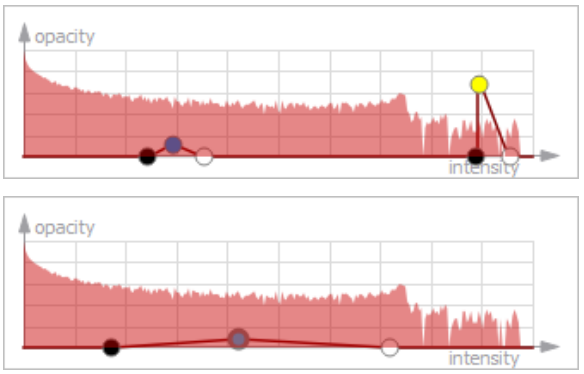

Figure 6: Transfer functions for Figure 4 *(top)* and Figure 5 *(bottom)* respectively.

5. Discussion and Future Work

This novel sampling approach and data representation opens up several opportunities. One implication of using the approach described in this paper is that the source data is not required to reside on the GPU during rendering. Since all "samples" are effectively acquired during the extraction of the iso-surfaces, no additional sampling needs to be performed during rendering. This naturally has both advantages:

+ Source data is not needed during rendering (zero texture fetches).
+ Very efficient data compression for visually sparse data (i.e, that the amount of visible voxels is small relative the size of the data).
+ Transfer functions no longer need to be evaluated during rendering as they will always be "sampled" at the value of the iso-surfaces.
+ The method is limited by the number of necessary samples, not directly by the number of voxels of the source volume.
+ Samples are coherent across pixels which effectively prevents the wood grain effect.

and disadvantages:

− Changing the transfer function means re-creating or re-uploading proxy geometry.
− Neighboring pixels can have vastly different evaluations due to tangent problem.
− Proxy meshes can be large due to granularity.
− Proxy meshes can be large due to an arbitrary number of shells for each mesh.
− Proxy meshes are inefficiently stored due to implicit data structure (could be mitigated by additional processing in the CPU).

Overall, the method has high potential due to the fact that its complexity seems to scale with information content rather than volume size. The fact that it contains a built in mech-

anism for importance driven rendering without much overhead in terms of specific management of LODs is also a great bonus. On the other hand, the method is somewhat sensitive to mechanism that increase the number of triangles that need to be generated. As such, it will probably be less suited for data that contain high amounts of noise, data that is uniform in nature, or where the visible data covers most of the volume. There is also great potential with respect to transfer functions as the transfer function will always, and only, be "sampled" at the values of the iso-surfaces. In fact, during rendering, the transfer function can be reduced to an array of pre-sampled values that is accessed without interpolation.

It is hard to make generalizations for the performance of the method as the computational complexity is highly dependent on the particular data and use case. Our empirical experience so far is that the rendering speed is up to par with other implementations for data that is sufficiently sparse, and then often with better visualization quality. In the future we will make a more thorough analysis of the individual parts of the method, including their limits and their sweetspots. We will also continue to work on the problem of potentially large meshes and will also investigate the possibilities to compute bounded errors during rendering.

References

[BM98] BOLIN M. R., MEYER G. W.: A perceptually based adaptive sampling algorithm. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques* (1998), ACM, pp. 299–309. 2

[EHK*06] ENGEL K., HADWIGER M., KNISS J., REZK-SALAMA C., WEISKOPF D.: *Real Time Volume Graphics*. Ak Peters Series. A K Peters, Limited, 2006. 2

[KE04] KRAUS M., ERTL T.: Pre-integrated volume rendering. *The Visualization Handbook 2* (2004), 3. 2

[KHW*09] KNOLL A., HIJAZI Y., WESTERTEIGER R., SCHOTT M., HANSEN C., HAGEN H.: Volume ray casting with peak finding and differential sampling. *Visualization and Computer Graphics, IEEE Transactions on 15*, 6 (2009), 1571–1578. 2

[Lev90] LEVOY M.: Volume rendering by adaptive refinement. *The Visual Computer 6*, 1 (1990), 2–7. 2

[Max95] MAX N.: Optical models for direct volume rendering. *Visualization and Computer Graphics, IEEE Transactions on 1*, 2 (jun 1995), 99 –108. 1

[NH93] NING P., HESSELINK L.: Fast volume rendering of compressed data. In *Visualization, 1993. Visualization'93, Proceedings., IEEE Conference on* (1993), IEEE, pp. 11–18. 2

[Sha03] SHAPIRO A.: Monte carlo sampling methods. *Handbooks in operations research and management science 10* (2003), 353–426. 2

[WWH*00] WEILER M., WESTERMANN R., HANSEN C., ZIMMERMANN K., ERTL T.: Level-of-detail volume rendering via 3d textures. In *Proceedings of the 2000 IEEE symposium on Volume visualization* (2000), ACM, pp. 7–13. 2