

# FROM VOLUME SPLATTING TO RAYCASTING

## *A Smooth Transition*

Guillaume Gilet, Jean-Michel Dischler

*Laboratoire des Sciences de l'Images, de l'Informatique et de la Télédétection, Université Louis Pasteur, Bd Sebastien Brant, Illkirch, France  
Guillaume.Gilet@lsiit.u-strasbg.fr; Dischler@lsiit.u-strasbg.fr*

Luc Soler

*Institut de Recherche contre le cancer de l'appareil digestif, 1 Place de l'hôpital, Strasbourg, France  
Luc.Soler@ircad.u-strasbg.fr*

**Keywords:** Volume Rendering, Point-Based Rendering, Raycasting.

**Abstract:** Splatting-based methods are well suited to render large hierarchical structured or unstructured point-based volumetric datasets. However, as for most object-order volume rendering methods, one major problem still remains the processing of huge amounts of elements usually necessary to represent highly detailed densely sampled datasets, which can lead to poor performances. Resampling into 3D textures to apply hardware-based raycasting is a common way of improving framerate in such cases but often with a certain precision loss related to limited texture memory. Switching between these two rendering techniques is interesting in order to keep the advantages of both and has thus been proposed before, but not in a smooth and hierarchical manner. In this paper, we show that for most point-based volumetric datasets, some parts of the model can be rendered more efficiently with a texture-based method, whereas other parts can be rendered as usual using splatting. We address the issue of providing a smooth hierarchical transition between the two methods using a data-dependent approach based on a per-pixel ray-driven rendering scheme. In practice, our transition scheme allows users a good control of the performance/quality trade-off. Through a comparison with standard hierarchical EWA splatting, we show that our smooth transition can lead to an improvement of framerate without introducing visual inconsistencies or artifacts.

## 1 INTRODUCTION

Volume rendering is a useful technique for the efficient visualization of volumetric datasets. Whether these datasets result from simulations, from measurements of some physical processes or from modern 3D scanning devices, they are very often expressed as irregularly sampled point clouds. The most natural method to visualize these point-based datasets is splatting, first introduced by Westover (Westover, 1989). The splatting process reconstructs a continuous field from the sampled scalar field using 3D reconstruction kernels associated with each scalar value and is therefore well suited for direct rendering of large unstructured point-based datasets. However, like all object order algorithms, such methods are often bound by the inherent point complexity thus showing poor performance for highly detailed and very densely sampled datasets. Resampling such datasets into 3D textures in order

to use a hardware-accelerated raycasting scheme is a common method for real-time visualization. Whereas such a texture-based scheme offers an efficient interactive visualization tool, its usefulness can be hampered by several constraints, such as GPU texture memory limitation. It is therefore often impractical to use a texture-based volume visualization tool for large point-based datasets.

Since each method has its strengths and drawbacks, it seems to be an interesting idea to devise a hybrid scheme using an appropriate combination of these methods, in order to increase performance. Several methods have been introduced to allow the visualization of large unstructured datasets using such a hybrid scheme. However, whereas these methods achieved efficient results, few relied on an effective combination scheme of both rendering principles with easy and efficient user control of visual results when switching between both.

In this paper, we introduce a new method based on an efficient combination of the EWA (*Elliptical Weighted Average*) volume splatting framework and hardware-accelerated raycasting. Basically, we propose a scheme aiming at a smooth transition between a flexible point-based method and texture-based hardware-accelerated methods. The main idea is to render each part of a dataset using an efficient combination of point/texture-based methods. Contrary to other transition methods, we do not propose a binary choice between the two methods, but rather an efficient combination of these methods, thus maximizing rendering performance. Our algorithm also takes full advantage of the latest hardware accelerations. In practice, our method improves the framerate of splatting without introducing visual inconsistencies.

The remainder of this paper is structured as follows. Section 2 discusses some related works. Next, we present our combination scheme and simplification criteria. Finally, before concluding, results and an analysis of our method are discussed in section 4.

## 2 PREVIOUS WORKS

Direct volume rendering methods have in common an approximative evaluation of the volume rendering integral for each screen pixel, i.e. the integration of attenuated colors and extinction coefficients along each corresponding viewing ray. Color and extinction coefficients are computed by a *classification* step. Classification is achieved by means of a transfer function which maps scalar values  $s = s(x)$  of the dataset to colors  $c(s)$  and extinction coefficients  $\tau(s)$ . By assuming that the viewing ray  $x(\lambda)$  is parameterized by  $\lambda$  the distance to the viewpoint, the classical volume rendering integral can be written as:

$$I = \int_0^D c(s(x(\lambda))) \exp\left(-\int_0^\lambda \tau(s(x(\lambda'))) d\lambda'\right) d\lambda \quad (1)$$

with  $D$  being the maximum distance.

The continuous data field is usually represented by a discrete function with values at vertices along with an interpolation scheme, based on a reconstruction kernel. This reconstruction kernel has a great impact on image quality and signal reconstruction accuracy.

Two main approaches can be distinguished for direct volume rendering : Forward projection methods and backward projection methods. Whereas in forward projection methods rays are cast from the image into the volume ((Roettger et al., 2003)), backward projection methods map volume elements onto the screen ((Zwicker et al., 2001),(Cohen,

2006)). An overview of recent volume rendering methods can be found in (Kaufman and Mueller, 2005). Several of these algorithms have now been successfully modified to take benefits from latest graphics hardware ((Ma et al., 2003) presents a brief overview of several hardware-accelerated volume rendering methods).

Several volume splatting algorithms focus on improving the image quality, such as (Mueller et al., 1999), (Neophytou et al., 2006) and (Neophytou and Mueller, 2003). Xue and Crawford (Xue and Crawford, 2003) compared several hardware accelerations for splatting algorithms, showing the efficiency of shader-based methods for large datasets rendering. The popular EWA (Elliptical Weighted Average) Splatting ((Zwicker et al., 2001), (Chen et al., 2004)) provides an efficient framework for interactive splat based volume visualization. However, in order to obtain correct and high quality images with splatting methods, splats must often be depth-sorted, resulting for most highly complex and densely sampled models in a severe impact on performance. There are several techniques to improve splatting performances by reducing the number of splats to be projected ((Mueller et al., 1999),(Laur and Hanrahan, 1991)), but with some limitations.

To overcome issues of splatting methods when dealing with densely sampled datasets and issues of raycasting methods when rendering large unstructured datasets, hybrid methods are a natural solution ((Ma et al., 2002)). In a similar manner, Wilson et al. introduced in (Wilson et al., 2002) a hardware-assisted hybrid scheme using semi-opaque splatting in conjunction with a texture-based method using a uniformly subsampled version of the dataset. Recently, Kaehler proposed in (Ralf Kaehler and Hege, 2007) an efficient hierarchical hybrid representation for the storage and rendering of large unstructured datasets.

These methods basically propose a choice between using a point-based or a texture-based method or both to render a subpart of a dataset. In contrast, we propose in this paper a hybrid method based on an effective combination of these techniques through the use of a smooth transition between both.

### 3 FROM SPLATTING TO RAYCASTING : A UNIFORM REPRESENTATION

This section describes our new hybrid rendering scheme. The idea is to render the volumetric dataset with a hybrid splatting/raycasting method.

As highlighted in (Meissner et al., 2000), we have on the one hand a splatting method efficient to render sparse structures and on the other hand a hardware-based raycasting suited to render densely sampled datasets. Therefore, the motivation of our method is to obtain a framework allowing for the rendering of a dataset using a combination of both methods, thus attaining higher performance. In order to achieve an

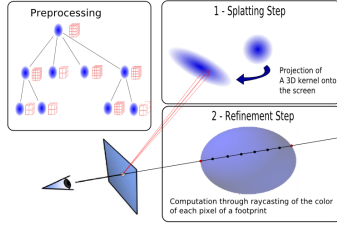


Figure 1: Principle of the method. An element is projected onto the screen plane, yielding a Gaussian 2D footprint. Details are added to each pixel with ray marching inside the corresponding 3D texture.

efficient combination of both object-order and image-based methods, we need a uniform representation of these techniques. To this end, we choose to define our new hybrid technique as composed of two steps :

- A splatting step, corresponding to the correct projection of 3D kernels into the screen, yielding 2D footprints.
- A refinement step, i.e. the computation of a color value for each pixel of the corresponding footprint.

As one can see, both classical rendering schemes can be straightforwardly mapped to this representation, i.e. classical raycasting methods being the projection of a single 3D kernel bounding the dataset and detailed through a raycasting algorithm, classical splatting methods being the projection of a collection of 3D kernels with a unique value for each kernel.

Figure 1 shows the principle of our method. The key idea is to render the dataset as a collection of overlapping 3D *detailed* kernels. These kernels can be obtained by resampling the dataset at a given rate and assigning a 3D texture and an interpolation kernel

to each sample point. The resampling rate of a dataset and the resolution of the corresponding 3D texture are key control-features of our hybrid scheme and must be defined for each region (contiguous subpart) of the dataset. To this end, we provide a hierarchical representation of the input volumetric dataset, where each node is a coarse representation of a region and is associated with a 3D texture containing the necessary information for the rendering of the associated region.

As shown in figure 2, a subpart of this hierarchical representation is selected at run time and rendered using a GPU-accelerated splatting scheme. Details are then added to each footprint using raycasting principles with the 3D texture associated to the node. The composition of each detailed footprint yields the final image.

We explain in the following subsection the actual rendering method of a set of overlapping detailed 3D kernels. Details concerning the hierarchical representation are given in subsection 3.2.

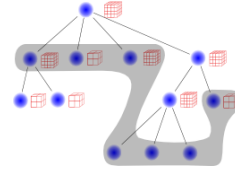


Figure 2: Hierarchical representation. Each node represents its child nodes and is associated with a 3D texture.

#### 3.1 Hybrid Rendering

As described before, the integration of raycasting principles into the splatting method must be carefully analyzed in order to ensure consistent results with an improvement of performance.

At each node of our hierarchical structure is attached a truncated spherical interpolation function (a radially symmetric Gaussian in our case). For each fragment  $f_{i,x,y}$  of a footprint  $F_i$  at screen coordinates  $x, y$  resulting from the projection of a Gaussian kernel  $G_i$ , a ray is cast through the fragment into the volume. We focus on the intersection of this ray with the kernel  $G_i$ . Since  $G_i$  is in our implementation a truncated radially symmetric Gaussian, its bounding box  $B_i$  can be seen as a sphere centered at the original sample point. The ray originating at fragment  $f_{i,x,y}$  intersects the bounding sphere in two points  $p_a$  and  $p_b$  (with  $p_a = p_b$  in extreme cases).

We then use a raycasting scheme into the adequate 3D texture stored in the graphics hardware to enhance the appearance of  $f_{i,x,y}$  by adding details along the segment  $p_ap_b$ . Through a careful analysis of the

3D texture resolution and the extent of the truncated kernel  $G_i$ , we can derive an adequate sampling step along the segment, and thus divide the segment  $p_a p_b$  into  $n$  equal sections of length  $l$ .

Each of these segments is associated with a color and opacity value using a lookup into a 2D table computed through a classical pre-integrated classification scheme (Engel et al., 2001). Due to the specificity of our scheme, segment lengths may vary along with the size of the kernel. To be able to rely on the 2D table lookup and still ensure correct results, we use the approximation scheme proposed by (Roettger et al., 2000) and weight the result of the 2D lookup with the length  $l$  of the segment.

## 3.2 Hierarchical representation

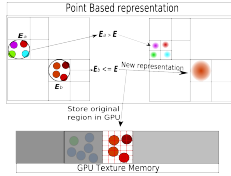


Figure 3: Creation process of our hybrid representation. Groups of points meeting the criterion are represented by a single kernel. The information of the original region are stored into the GPU texture memory.

We choose to simplify the volumetric dataset using an octree as in (Laur and Hanrahan, 1991). The input volumetric dataset is divided into a collection of hierarchical nodes, each node representing a subpart of the volumetric dataset. A feature-sensitive top-down approach is applied during the construction of the point hierarchy.

Our representation is based on a division criterion  $E$ . This criterion represents relevant features and properties of a subpart of the dataset. This criterion is used to determine the parameters of our hybrid method, such as the sampling rate of the dataset, the resolution of the 3D textures, thus orienting our rendering scheme toward splatting or raycasting for each subpart of a dataset. We propose in this paper two different criteria described in sections 3.2.1 and 3.2.2. Each region approximated by a single node (i.e. meeting the criterion requirement) is stored into a 3D texture of equivalent resolution. This simplification scheme yields a simplified representation of the volume coherent with our hybrid rendering solution.

### 3.2.1 Visual variation criterion

As highlighted in (Meissner et al., 2000), splatting and raycasting methods produce slightly different visual results. This difference is the main cause of popping artifacts in most hybrid methods. The aim of our method being a smooth artifact-less transition, these visual differences should be minimized. Intuitively, we want to resample and orient toward a raycasting solution the regions which will induce low visual variations. In order to detect such regions, we need to define a qualitative error metric measuring the visual variation between the two schemes. This variation  $v_r$  is expressed for a viewing ray  $r$  traversing the volume. We choose to define this variation as the distance in color space between two colors computed along the viewing ray  $r$  by evaluating the volume rendering integral (1) using the two different approximation schemes (splatting and raycasting):

$$v_r = |C_{splat} - C_{raycast}| \quad (2)$$

This visual variation reflects the visual differences between the splatting and raycasting solution for a single viewing ray and can be considered as a measurement of visual artifacts.

We choose to express the visual variation  $V$  over a complete contiguous region as the average value of the squared variation along all possible viewing rays traversing this region :

$$V = \int_{\Omega} v_r^2 \partial\Omega \quad (3)$$

with  $\Omega$  the sphere of all possible view directions.

We propose to approximate this integral by using a discrete sum evaluated over a small subset  $N$  of viewing rays determined by using a stochastic distribution of viewing rays across a region.

Although the precomputation of the transfer table used by the ray casting scheme in section 3.1 can be easily computed, thus allowing the transfer function to be edited on the fly, our variation measurement scheme is more complex. As of this moment, it is not possible to modify the transfer function and keep an interactive rendering, even for a small value of  $N$ . For some applications however, it can be desirable to edit the transfer function on the fly. To this end, we provide another division criterion, thus allowing an interactive simplification of the dataset.

### 3.2.2 Data variance criterion

The idea is to use data variance inside a region as a decision criterion for our simplification. Intuitively, regions of a dataset with high variance will introduce more visual differences. It is then compared with a

user defined threshold, defined empirically as of now. Whereas this criterion is plainly less efficient than the visual variation measurement criterion, it allows for an interactive simplification. It is to be noted that this variance analysis can be performed with or without taking the transfer function into account. Nonetheless, this criterion gives rather good and often sufficient results for most datasets.

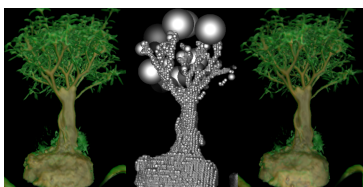


Figure 4: Tree Model. Left : Traditional splatting. 8M splats - 2.2 fps. Middle : Our hybrid representation. Right : Hybrid method with detailed textures. 46k splats - 8.3 fps

## 4 RESULTS

This section deals with a study of visual quality and performance of our method. We compare our hybrid rendering obtained with different parameters with the EWA splatting rendering of the same volumetric data. We show results on both structured and unstructured datasets. Timings were collected on an Intel Core 2 Duo Processor at 2.4Ghz and a GeForce 8800 graphics card using a viewport resolution of  $512 \times 512$ . As can be seen on figure 4 (right), our hybrid method allows the rendering of datasets with a low splats count, yet maintaining the same visual result as with traditional EWA splatting (left). This induces an increase in performance over traditional splatting without quality loss. Our visual variation criterion can lead for most models to an efficient reduction (50-80%) of the point count necessary to render a dataset with almost the same visual quality. Figure 5 shows the impact of different user-defined quality thresholds on quality and framerate of our method. Naturally, as the threshold grows higher, performance increases while quality is degraded in some areas. On figure 6, we show the differences of the two criteria respectively described in sections 3.2.1 and 3.2.2. While most features are preserved, areas with low variation (across the neck of the model) are degraded with the variance data criterion but are accurately represented using the visual difference criterion. Figure 7 shows the same model rendered with and without texture details. Our method also induces an increase of framerate for unstructured datasets, as shown in figure 8.

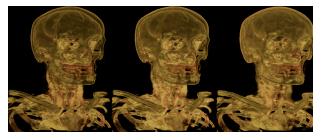


Figure 5: A  $512^3$  head dataset. Left : 2M splats -  $\epsilon = 2\%$  - 2.9 fps; Middle : 386k splats -  $\epsilon = 5\%$  - 4.8 fps; Right : 250k splats -  $\epsilon = 10\%$  - 8.8 fps.

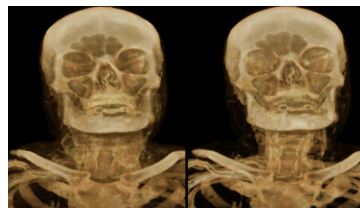


Figure 6:  $512^3$  Head Model rendered with 530k splats using the two criteria. Left : Visual Variation criterion. Right : Variance data criterion

## 5 CONCLUSIONS

We presented in this paper a hardware-accelerated hybrid volume rendering scheme. Our method provides a smooth transition between texture-based and point-based techniques. Through an adequate division of the dataset, each region of the dataset can be rendered with an efficient combination of both methods. Furthermore, we provided efficient division criteria for the analysis and simplification of a dataset according to user-defined parameters controlling the quality/speed trade-off. This technique leads to a smooth transition between two traditional principles and an increase in performance over traditional splatting methods for high quality rendering of point-based volumetric datasets.

Several issues can be improved. In the future, we want to increase the efficiency of our framework. First of all, the use of a different hierarchical structure (such as RBF-based representation) can allow for a representation more fitting to data, thus allowing for both performances and visual quality increase. Another improvement is to derive the idea of *adaptive* hardware accelerated EWA splatting ((Chen et al., 2004)) to further enhance framerate. However, such an improvement is not trivially devised in our case. An adaptive scheme can be used to affect, not only the quality of the reconstruction kernel, but also the sampling step of the raycasting scheme. Furthermore, an adaptive criterion should be devised while taking into account our visual variation metric and several importance factors, such as the size or visibility of a region (as a measurement of visual importance of a region on the screen), as to enhance the splats count reduction

without introducing visual artifacts. As it is, visual variations over a region are bound by our metric, but these variations are accumulated onto the screen by different locally variation-bound regions. To this end, a thorough analysis of our metric can be proposed to lead to a better adjustment of parameters and provide an approximation of the total visual variation on the final resulting image.

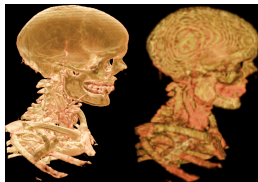


Figure 7: A  $512^3$  medical dataset. Left :Our hybrid method. 2M splats at 2.1 fps. Right : Without texture details. 2M splats at 12.3 fps.

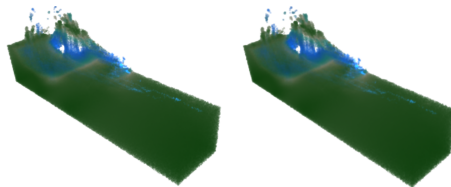


Figure 8: An unstructured fluid dataset obtained through simulation MAC. Left : splatting method - 1.5M splats - 11 fps. Right : our method - 250k splats - 16 fps.

## ACKNOWLEDGEMENTS

The medical dataset was gratefully provided by IRCAD : [www.ircad.fr](http://www.ircad.fr).

## REFERENCES

- Chen, W., Ren, L., Zwicker, M., and Pfister, H. (2004). Hardware-accelerated adaptive ewa volume splatting. In *VIS '04: Proceedings of the conference on Visualization '04*, pages 67–74.
- Cohen, J. D. (2006). Projected tetrahedra revisited: A barycentric formulation applied to digital radiograph reconstruction using higher-order attenuation functions. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):461–473.
- Engel, K., Kraus, M., and Ertl, T. (2001). High-quality pre-integrated volume rendering using hardware-accelerated pixel shading. In *HWWS '01: Proceedings of the ACM SIGGRAPH/EUROGRAPHICS workshop on Graphics hardware*, pages 9–16. ACM Press.
- Kaufman, A. and Mueller, K. (2005). Overview of volume rendering. In *The Visualization Handbook*.
- Laur, D. and Hanrahan, P. (1991). Hierarchical splatting: a progressive refinement algorithm for volume rendering. In *SIGGRAPH '91: Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, pages 285–288. ACM Press.
- Ma, K.-L., Lum, E., and Muraki, S. (2003). Recent advances in hardware-accelerated volume rendering. *Computers and Graphics*, 27(5):725–734.
- Ma, K.-L., Schussman, G., Wilson, B., Ko, K., Qiang, J., and Ryne, R. (2002). Advanced visualization technology for terascale particle accelerator simulations. In *Supercomputing '02: Proceedings of the 2002 ACM/IEEE conference on Supercomputing*, pages 1–11.
- Meissner, M., Huang, J., Bartz, D., Mueller, K., and Crawfis, R. (2000). A practical evaluation of popular volume rendering algorithms. In *VVS '00: Proceedings of the 2000 IEEE symposium on Volume visualization*, pages 81–90.
- Mueller, K., Shareef, N., Huang, J., and Crawfis, R. (1999). High-quality splatting on rectilinear grids with efficient culling of occluded voxels. *IEEE Transactions on Visualization and Computer Graphics*, 5(2):116–134.
- Neophytou, N. and Mueller, K. (2003). Post-convolved splatting. In *Joint Eurographics - IEEE TCVG Symposium on Visualization 2003*, pages 223–230.
- Neophytou, N., Mueller, K., McDonnell, K. T., Hong, W., Guan, X., Qin, H., and Kaufman, A. (2006). Gpu-accelerated volume splatting with elliptical rbfs. In *IEEE TCVG Symposium on Visualization 2006*.
- Ralf Kaehler, T. A. and Hege, H.-C. (2007). Simultaneous gpu-assisted raycasting of unstructured point sets and volumetric grid data. In *Eurographics / IEEE VGTC Workshop on Volume Graphics 2007* (2007).
- Roettger, S., Guthe, S., Weiskopf, D., Ertl, T., and Strasser, W. (2003). Smart hardware-accelerated volume rendering. In *Proc. of VisSym '03*.
- Roettger, S., Kraus, M., and Ertl, T. (2000). Hardware-accelerated volume and isosurface rendering based on cell-projection. In *Proceedings of the 11th IEEE Visualization 2000 Conference*. IEEE Computer Society.
- Westover, L. (1989). Interactive volume rendering. In *VVS '89: Proceedings of the 1989 Chapel Hill workshop on Volume visualization*, pages 9–16. ACM Press.
- Wilson, B., Ma, K.-L., and McCormick, P. S. (2002). A hardware-assisted hybrid rendering technique for interactive volume visualization. In *VVS '02: Proceedings of the 2002 IEEE symposium on Volume visualization and graphics*, pages 123–130.
- Xue, D. and Crawfis, R. (2003). Efficient splatting using modern graphics hardware. In *Journal of Graphics Tools*, volume 8, pages 1–21.
- Zwicker, M., Pfister, H., van Baar, J., and Gross, M. (2001). Ewa volume splatting. In *VIS '01: Proceedings of the conference on Visualization '01*, pages 29–36. IEEE Computer Society.