Volume Visualization IN4086 Data Vizualization*

Group 11†

Nivedita Prasad
TU Deflt
4712099, Delft
N.Prasad@student.tudelft.nl

Riya Maan
TU Delft
4719913, Delft
R.Maan@student.tudelft.nl

Aishwarya Shastry
TU Delft
4743016, Delft
A.Shastry@student.tudelft.nl

ABSTRACT

This is Volume Visualization report of Group 11 for IN4086 Data Visualization.

KEYWORDS

Interpolation, Ray-casting, Color, Opacity, Volume Rendering

ACM Reference Format:

Nivedita Prasad, Riya Maan, and Aishwarya Shastry. 2018. Volume Visualization IN4086 Data Vizualization: Group 11. In *Proceedings of* . ACM, New York, NY, USA, 5 pages.

1 INTRODUCTION

The aim of this project is to extend a volume renderer to make a complete and functional ray-caster. Direct Volume Rendering deals with the volume data which has internal structure in contrast to indirect volume rendering which uses isosurfaces. In this project we are using ray cast rendering that is an image ordering technique where we have to calculate the color and opacity of each pixel. This ray cast rendering approach is used to develop a Volume renderer. The visualization techniques we have used are Maximum Intensity Projection, Color Composting, 2D transfer function and Phong shading models to highlight important features present among the dataset examples provided to us.

2 VISUALIZATION TECHNIQUES

In this section we have described in detailed about the visualization techniques namely Trilinear interpolation,Trilinear interpolation with gradients, Maximum Intensity Projection, Color Composting, 2D transfer function, Gradient based Opacity Weighting, Blinn-Pong shading.

2.1 Trilinear Interpolation

Interpolation is the method we use to estimate the values in unknown positions. Tri-linear interpolation linearly interpolates vertices of a cell. A volume data is made up of voxels which are the vertices and its intensity is a scalar value. The exact position of the voxels have to be estimated as the desired point might not be present in the voxel positions. This is where Tri-Linear interpolation will help us by linearly interpolating the vertices of a cell. A basic representation is depicted in Fig. 1 [1]. In the figure, the 3D cube has voxels(data points) as its vertices. The basic calculations that have to be done before we perform trilinear interpolation *'c'* are :

- Calculate x_d,y_d and z_d, which are the position weights with respect to corresponding x, y and z coordinates
- The position weights are used to first interpolate x direction for each of the 4 faces which are the blue points in the Fig. 1 i.e. c_{00}, c_{01}, c_{10} and c_{11}
- The intensity values x_0, y_0 and z_0 in f (x_0,y_0,z_0)

$$x_d = \frac{x - x_0}{x_1 - x_0};$$
$$y_d = \frac{y - y_0}{y_1 - y_0};$$
$$z_d = \frac{z - z_0}{z_1 - z_0}$$

$$c_{00} = f(x_0, y_0, z_0)(1 - x_d) + f(x_1, y_0, z_0)x_d;$$
$$c_{01} = f(x_0, y_0, z_1)(1 - x_d) + f(x_1, y_0, z_1)x_d;$$
$$c_{10} = f(x_0, y_1, z_0)(1 - x_d) + f(x_1, y_1, z_0)x_d;$$
$$c_{11} = f(x_0, y_1, z_1)(1 - x_d) + f(x_1, y_1, z_1)x_d$$

The above are the four points to interpolate in Y direction.

$$c_0 = c00(1 - y_d) + c_{10}y_d;$$
$$c_1 = c01(1 - y_d) + c_{11}y_d$$

These 2 points are used to interpolate in the z-direction fiving the final estimated intensity.

$$c = c_0(1 - z_d) + c_1 z_d$$

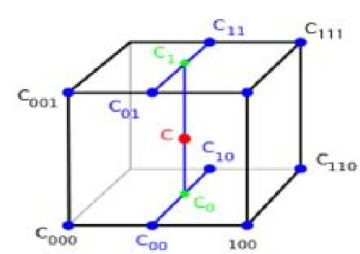

Figure 1: Trilinear Interpolation

*Produces the permission block, and copyright information
†

,
.

,

2.2 Trilinear Interpolation with Gradients

When dealing with higher order transfer functions, we don't have intensity on x, y and z values instead we have a 3-D vector which is our gradient. This results in three linear interpolations for each single linear interpolation for x, y and z dimension of the gradient vector which sums upto 21 linear interpolations per sample point in a ray.

2.3 Maximum Intensity Projection

The aim of Maximum Intensity Projection is to find the highest maximum intensity using trilinear projection so that the opacity of the pixel will be higher. A ray is passed through the volume data for each pixel in the image. Then multiple samples are casted along the ray and the sample with highest maximum intensity is chosen and assigned a color nearing white. In the *tracerayMIP()* function, the number of sample points along the ray is calculated. This is done by calculating the distance between entry and exit points and dividing the distance by the sample step and for each sample point we will find maximum intensity observed per ray.

2.4 Color Composting

Thee aim of this function is to provide color to each pixel. For this, the ray is sampled from back to front. The *composite* function returns an accumulated value for each pixel. The accumulated value is the merging of red, green, blue and the main components of each ray to finalize the final color of each pixel. The 1D transfer function, which is a linear function maps color and opacity values to pixels based on their intensity values. This is implemented in the *composting* method in *RaycastRenderer.java*. The color is obtained from the transfer function. The number of iterations are found and the sample is interpolated. Each of the sample is color accumulated using the equation

$$accColor = opacity * color + (1 - opacity) * accColor$$

where *accColor* is the colour accumulated so far and *color* and *opacity* of the sample. The accumulated opacity *accOpacity* is calculated as :

$$accOpacity = (1 - opacity) * accOpacity$$

The color from all the samples in the ray is accumulated and the channels are merged and the resulting color of the pixel is returned.

2.5 2D Transfer Function

Transfer functions are functions which make the volume data visible by mapping the data values to various properties like color, opacity[2] Transfer functions main task is for good quality volume rendering. A 2d transfer function maps opacity, color and radius for these data values and is considered good for boundary visualization.

2.6 Gradient Based Opacity Weighting

Gradient Based Opacity Weighing [3] particularly used for Surface Classification. It works in such a way that the pixels with intensity equivalent to a specified value become opaque and the opacity gradually decreases as we move further away from the point.

$$a_x = \begin{cases} 1 & \text{if } |\nabla f_x| = 0 \text{ and } f_x = f_v \\ 1 - \frac{1}{1}\frac{f_v - f_x}{|\nabla f_x|} & \text{if } |\nabla f_x| > 0 \text{ and } f_x - r|\nabla f_x| \leq f_x \leq f_x + r|\nabla f_x| \\ 0 & \text{otherwise.} \end{cases}$$

Where a_x = calculated opacity of the sample
f_x = $i$intensity of the sample
f_v = base intensity
r = radius selected in the widget

2.7 Blinn-Phong Shading

The Blinn Phong Shading Model is a modification of the Phong Model. The modification is in the way Specular contribution is computed. The equation for each color channel is

$$accColor = ambient + (color * N.L) + (N.HalfwayVec)^{\alpha}$$

where $HalfwayVec = L + viewvector/|L + viewvector|$ The images produced show more specular light than the normal Phong Shading Model. It is also seen to be faster method when the distance between the viewer and light source is very large.

3 RESULTS AND EVALUATION

3.1 Carp

In the Fig 2, we show result of the MIP mode on the carp dataset. The skeleton of the carp whose sample values are high is most white than the skin whose sample values are not high. Hence this visualization technique is useful for the study of the skeleton of the carp. In Fig 3, it is easily visible that the levels of the object is not visible. In Fig 4, we have used the transfer function on the carp dataset and we can now highlight and focus on the maximum intensity part and other intensity surfaces parallely.

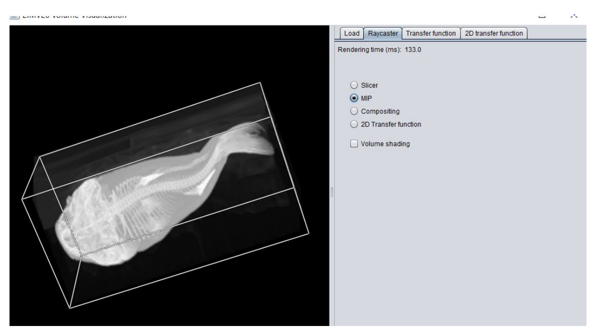

Figure 2: Carp dataset - MIP function

3.2 Backpack

In Fig 5 , it is visible that the contents of the backpack is clearly visible in the MIP function , where as color composting Fig 6 highlights the unnecessary features of the bag like its outline and pockets rather than all the contents of the bag while performing a check. Additionally, when 2D transfer function Fig 7 is applied only few objects are visible and rest of the contents have disappeared along

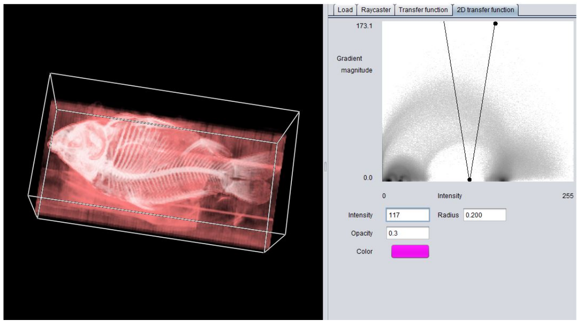

Figure 3: Carp dataset - Color Composting function

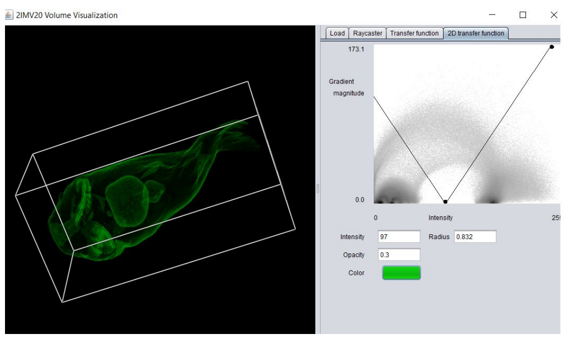

Figure 4: Carp dataset - 2D-Transfer function

with the outline of the bag. Hence MIP function is clearly performing well on this dataset.

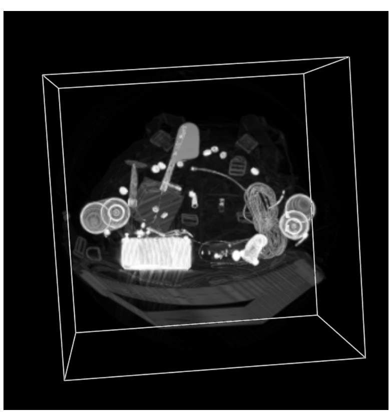

Figure 5: Backpack - MIP function

3.3 Tooth

Here, we can observe that the transfer function Fig 9, gives better results than the color composting method applied on the tooth dataset. this is clearly visible in Fig 8, whereas the MIP function applied on the tooth gives only an opaque image of the whole screen where nothing is visible. Hence 2D transfer function performs well on the tooth data set highlighting the densities of the tooth well.

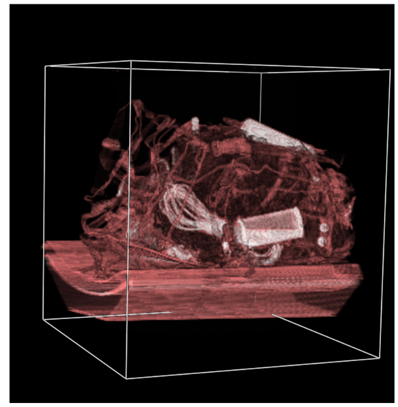

Figure 6: Backpack - Color Composting function

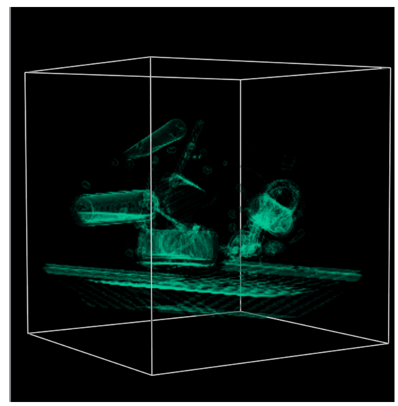

Figure 7: Backpack - 2D-Transfer function

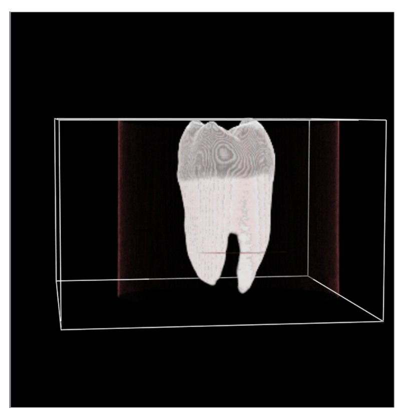

Figure 8: Tooth dataset - Color Composting function

3.4 Piggybank

In Fig 10 In Fig 11, we can see the coins more clearly when composite and volume shading are applied together. The transfer function gives less detail than composite and volume combined together 12,w

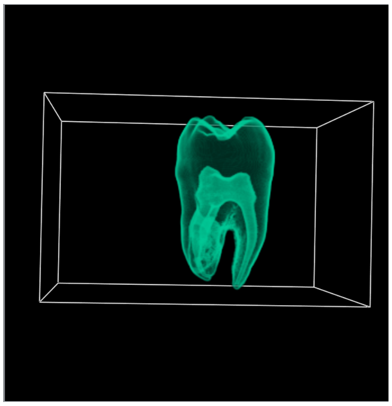

Figure 9: Tooth dataset - 2D-Tranfer function

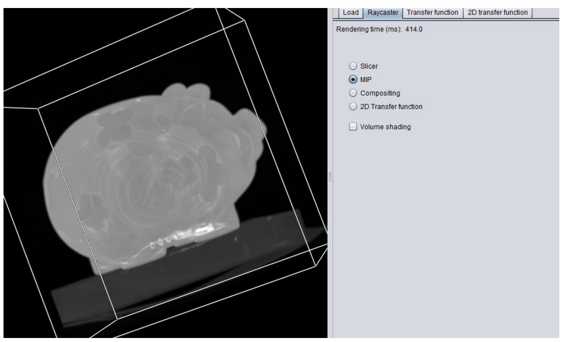

Figure 10: PiggyBank - MIP function

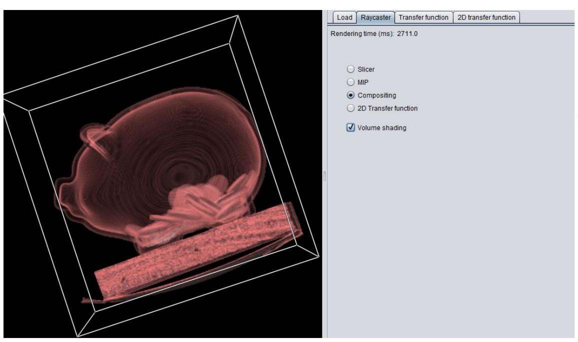

Figure 11: PiggyBank - Color Composting function

3.5 Bonsai

The transfer function gives more information on the bonsai dataset Fig 15. As the leaves of the bonsai is visible even at very low intensity. The compositing function Fig 14 is less detailed, while the MIP Fig 13 do not show accurate depictions of the bonsai.

4 CONCLUSION

Direct Volume Rendering is used to render volume data in this project even though indirect volume visualization and 2D image rendering are present. We have used the ray casting method to select color of each pixel by sending a ray through the volume. The theoretical explanation of each of the visualization technique such as maximum intensity projection, Color Composting and 2D transfer function which is explained accompanied by the results and evaluation.

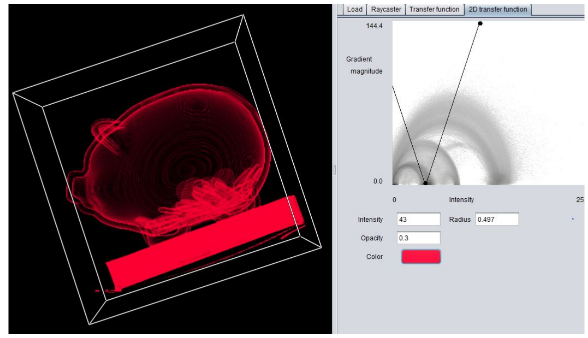

Figure 12: PiggyBank - 2D-Transfer function

Figure 13: Bonsai dataset - MIP function

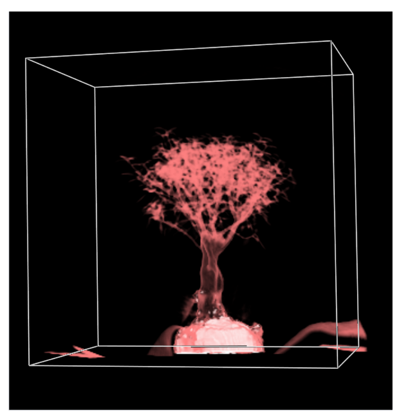

Figure 14: Bonsai dataset - Color Composting function

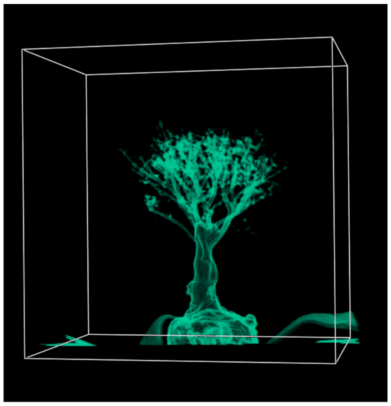

Figure 15: Bonsai dataset - 2D-Transfer function

5 INDUVIDUAL REPORT

5.1 Riya Maan

I have contributed to most of the coding of the project and it was a pleasure to work as a team. This project gave me an opportunity to dive into different 3D visualization techniques. Tri-linear interpolation, ray tracing algorithms and Blinn shading model were particularly interesting. My team mates coped up with me while working and documented the processes well. It was fun to work with different datasets and to be able to manipulate different views of the 3D object.

5.2 Nivedita Prasad

This was a very fun project to work with. We as a team have really enjoyed this project as it has helped us understand the nuances of visualization techniques. I contributed to part of the coding and 65% of the report writing. It was a pleasure to document all the fun and learning we had experienced during the project. I had concentrated on maximum intensity projection particularly and the final results.

5.3 Aishwarya Shastry

I worked parallely with my teammates and contributed to the rest of the report and coding as we worked in unison. It was a informative project overall driving us to understand volume visualization techniques which would have not been possible through just lectures. The project was a challenge for all of us and we were not able to extend the Triangular widget as we wanted to.

REFERENCES

[1] [n. d.]. ([n. d.]). https://en.wikipedia.org/wiki/Trilinear_interpolation

[2] [n. d.]. ([n. d.]). https://people.cs.uchicago.edu/~glk/talks/pdf/Kindlmann-DVRwithMDTFandDTIVis-SIGGRAPH-2005.pdf

[3] Marc Levoy. [n. d.]. Display of surfaces from volume data" journal =. ([n. d.]).