VOLUME PAINTING: INCORPORATING VOLUMETRIC RENDERING WITH LINE INTEGRAL CONVOLUTION (LIC)

A Thesis

by

JAEWOOK LEE

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

August 2005

Major Subject: Visualization Sciences

VOLUME PAINTING: INCORPORATING VOLUMETRIC RENDERING WITH LINE INTEGRAL CONVOLUTION (LIC)

A Thesis

by

JAEWOOK LEE

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

August 2005

Major Subject: Visualization Sciences

ABSTRACT

Volume Painting:

Incorporating Volumetric Rendering with

Line Integral Convolution (LIC). (August 2005)

Jaewook Lee, B.F.A., Pusan National University

Chair of Advisory Committee: Dr. Ergun Akleman

This thesis presents an expressive (non-photorealistic) rendering approach created by combining volumetric rendering techniques with the Line Integral Convolution (LIC) in 3D space. Although some techniques that combine volume rendering with the LIC have been introduced in computer graphics, they are mainly used for the scientific visualization fields, such as the visualization of 3D fluid fields. Unlike earlier research, we will focus on artistic representation, which is significantly different than scientific visualization research.

We will implement a brush-stroke effect on the implicit surfaces by using the LIC. The implicit surfaces are described as volume datasets that are created by the voxelization of triangular meshes. To acquire smearing effects on the surface we convolve along the vector fields with the densities of the voxels of the datasets. These vector fields are defined by users as texture maps. The final images are rendered with volume ray casting, integrating colors and densities of voxels with Perlin noise along vector fields. The Perlin noise provides randomness and allows us to generate scratches. Smearing effects on the surface of an object create the illusion of 3D brush-strokes as if a painter had created brush strokes on a canvas.

The rendering system is implemented using standard C and C++ programming languages. 3D models are then created using Alias Maya^{TM} and Topmod^{TM}.

To my family and friends

ACKNOWLEDGMENTS

I would like to express my thanks to my graduate committee chair, Dr. Ergun Akleman, for his guidance and for his attention to detail. I am also grateful to my committee members, Dr. Donald H. House and Dr. Ricardo Gutierrez-Osuna, for providing additional knowledge throughout this process. I would also like to thank Zhang Xiao for his willingness to allow me to use his program for the voxelization and for advice on the process of creating a thesis. My gratitude goes to all of the faculty and staff of the Viz Lab who have nurtured my exuberance for knowledge and have aided me to achieve my goals. I would like to thank all the friends that I have made during my time in the Viz Lab for their inspiration and enthusiasm. Most of all, I would like to thank my parents for their unconditional and continuous support, trust, encouragement and love through the years.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I

INTRODUCTION

In this thesis, we introduce a new Non-Photorealistic Rendering (NPR) technique, called Volume Painting. An example of Volume Painting is shown in Figure 1.

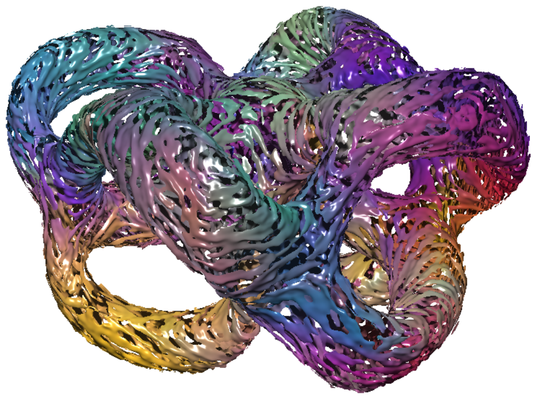


Fig. 1. Volume Painting of the Implicit Geometry.

During the last decade, NPR has become an important research area in Computer Graphics. NPR covers all rendering techniques which do not create realistic images. NPR is a fertile research area which has a wide variety of developments. There is still room to develop creative approaches in the field that can provide new expressive rendering techniques.

NPR techniques generally focus on simulating traditional artistic techniques such as impressionist paintings or pen-and-ink drawings. One of the active research directions that

The journal model is *IEEE Transactions on Visualization and Computer Graphics.*

has not been developed in NPR research is the creation of completely a new expressive rendering technique that does not simulate any existing painting styles.

Our new NPR technique is developed by incorporating volume rendering with Line Integral Convolution (LIC). The combination of volume rendering and LIC, previously used for scientific visualization, provides insight to the volumetric data [1]. In Volume Painting, our goal is not to provide perspective for the volumetric data, but merely to create visually appealing images by using the combination of volume rendering and Line Integral Convolution (LIC).

Volume painting allows artists to create their own creative styles by modulating different parameters. Our results are personal in the sense that it is an artistic representation based on the artist's viewpoint.

I.1. Inspiration and Motivation

This thesis comes from Line Integral Convolution technique, which has developed for visualizing 2D vector fields [2]. LIC is a low-pass filtering technique along a given streamline to place a smearing effect upon textured images. It has been applied to a variety of research fields, such as the visualization of fluid flow and the creation of textures.

LIC is very useful for the creation of artistic images in 2D as shown in Figure 2. Examples of Figure 2 are the images which are filtered by 2D LIC along a given vector field with different length of the streamline. LIC can successfully simulate brush-strokes, which is one of the most noticeable characteristics of traditional paintings. Smearing in a given vector direction can create the look of painter's brush by representing the directional information of the brush. By adjusting by parameters of the LIC, it is possible to simulate exuberant painting styles or artistic expressions.

Our goal in this thesis is to achieve an artistic smearing effect in 3D by using the LIC.

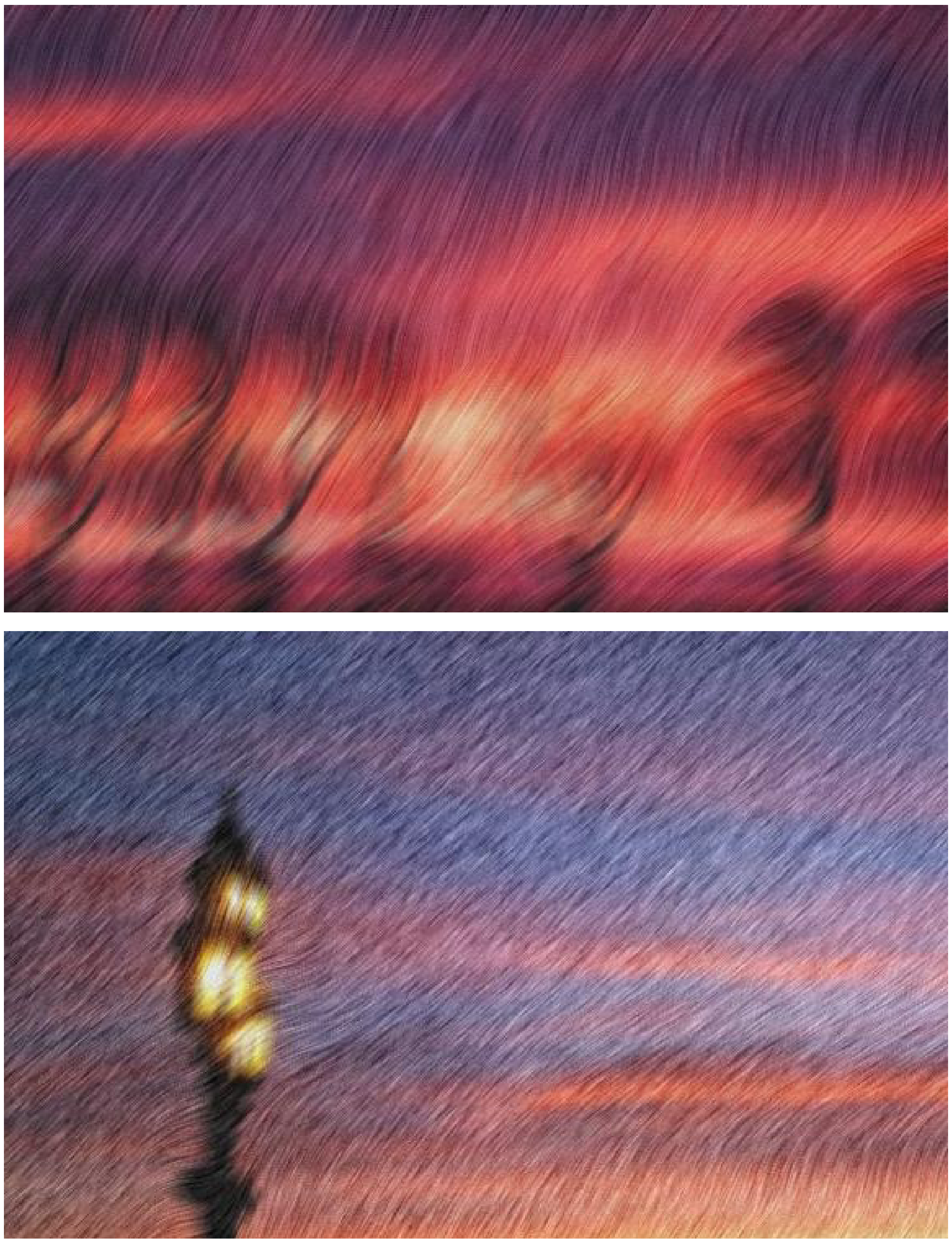


Fig. 2. Examples of the Line Integral Convolution (LIC) That Were Created by the Author as a Class Project in Viza654 - Digital Image Course.

To achieve this goal we have integrated the volume rendering with the LIC. Our technique, called Volume Painting, provides 3D artists with the capability of creating sculptured paintings.

CHAPTER II

RELATED WORK

Our work is related to three areas in Computer Graphics Research. These are Non-Photorealistic Rendering, Volume Rendering and Line Integral Convolution (LIC).

II.1. Non-Photorealistic Rendering (NPR)

Throughout history there have been a variety of painting styles. The methods of painting have evolved to include mixed media and various painting styles. Painters have also explored various brush strokes repeatedly to achieve their own different and unique effects. Research related to painterly 3D renderings have increased dramatically during the last decades. These techniques are generally called non-photorealistic rendering (NPR). With NPR techniques, it is possible to generate visually pleasing images that are easier to comprehend than the images that are created by using photorealistic techniques. NPR approaches have distinctive advantages for depicting 3D objects efficiently. For instance, a NPR approach can help expose a crucial part of a 3D image [3]. Some NPR approaches can be used to simplify the trivial parts of a 3D image, to control the viewer's attention. [4]. Meier [5] used this as a painterly rendering technique for animation, not for still images. Curtis et al. [6] presented a watercolor technique generated by computer simulating the traditional watercolor paintings. Hertzmann [7] showed a technique which paints an image with a series of spline brush-strokes.

II.2. Volume Rendering

Volume rendering techniques are utilized in the field of medical imaging. In the beginning, volume rendering was helpful to visualize large numbers of data sets acquired from Magnetic Resonance Imaging (MRI) machines, as well as Computed Temography (CT) scans. Both of these techniques is used to diagnose medical conditions such as brain tumors. With the advances in computing power and lower costs for memory storage, NPR techniques are used to create visual effects for the entertainment industry as well. Levoy [8] explored the application of volume rendering to display surfaces from the sampled scalar functions. Westermann and Ertl [9] presented ideas which use the advantage of modern hardware through standard APIs like OpenGL in volume rendering applications. Kniss et al. [10] demonstrated an important class of 3D transfer functions for scalar data and described a set of direct manipulation widgets.

II.3. Line Integral Convolution (LIC)

The Line Integral Convolution (LIC) has been used in the visualization of vector fields. the LIC [2] is actually a technique which filters a textured image along the streamline of the vector field. 3D implicit objects are visualized using volume rendering techniques. Interrante [11] also showed strategies for effectively portraying 3D flow using volume data with the LIC. Interrante [12] suggested how the set of principal directions and curvatures can be understood to define a natural ”flow” over the surface of an object. However, all of these techniques are for the visualization of scientific information and not for artistic representations, and while they drew the LIC effect by using the evenly distributed random point samples of an input texture, or evenly distributed set of tiny opaque particles, we instead use a noise function for the randomness and more artistic look. Lu et al. [3] implemented a non-photorealistic volume rendering system with the stippling technique. The stippling technique attempts to recreate traditional artistic method. However, it does not show the directional information of brush-strokes. Ebert and Rheingans [13] also proposed the method of non-photorealistic rendering of volume models. They discussed a LIC technique based on 2D spatial domain, not 3D.

The major difference between our method and the above listed methods is that we focus on an artistic rendering technique, which allows the user to adjust the direction of the brush-stroke intuitively. The brush-stroke will be controlled by a texture map that the user defines. Therefore, we represent the implicit object more effectively in a NPR style, specifically with the brush-stroke which has the directional information.

CHAPTER III

METHODOLOGY

In this chapter, we present the framework of our system. The structure of our system is given in Figure 3.

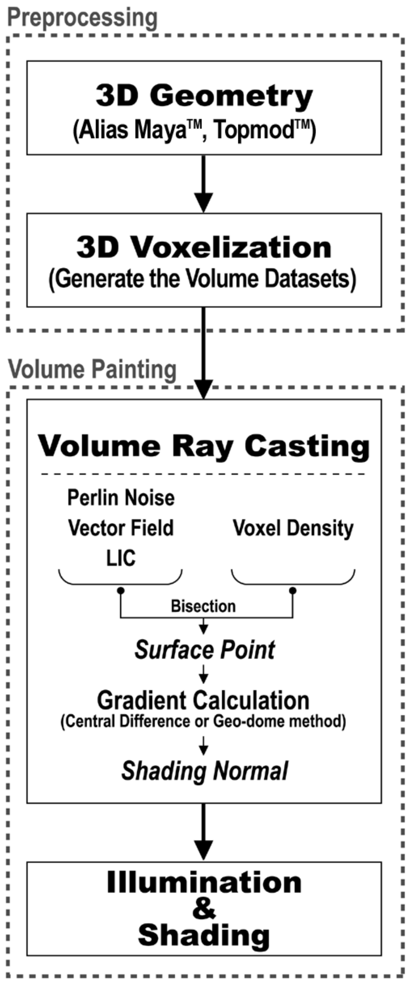


Fig. 3. Framework of Volume Painting.

III.1. Framework

This system has two main steps, the preprocessing and the volume painting. The preprocessing generates the volume datasets of triangular meshes modelled using Alias Maya^{TM} or Topmod^{TM} [14] shown in Figure 4. These volume datasets consist of voxel index, signed distance fields color, and directional information. The signed distance field is converted to density information with a transfer function, and used for estimating the real geometry normal (not affected by LIC) for calculating the brush-stroke directions.

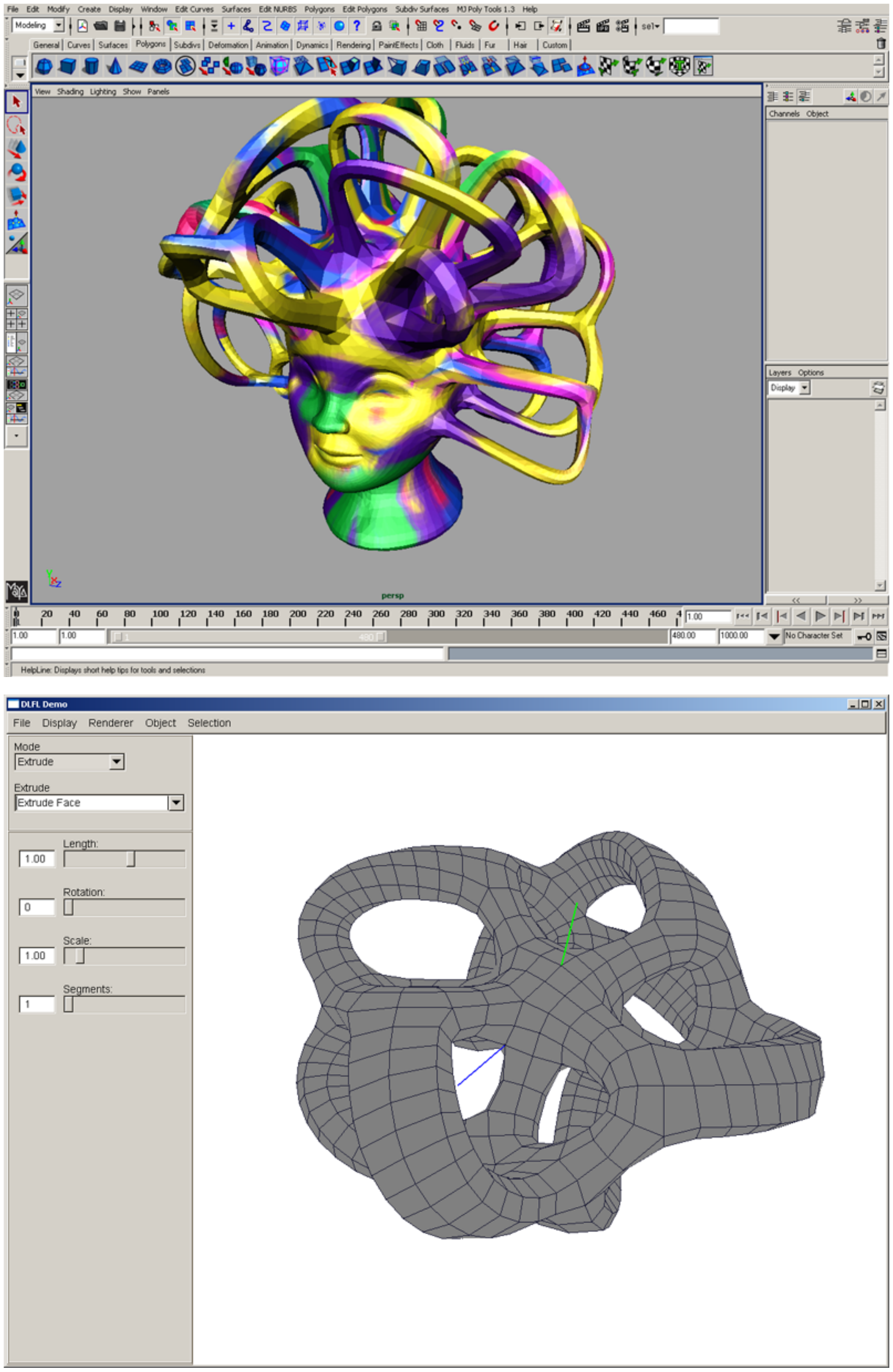
Unfortunately, the volume datasets are a 3D voxel array of which size is $30 \times 30 \times 30$, $50 \times 50 \times 50$, or the greatest size the physical memory of the processing computer allows. While larger voxel size can generate a higher quality image, it requires a large amount of physical memory and a longer processing time.

In the Volume Painting step, the proposed volume ray casting method allows for the visualization of the volume datasets in a much more efficient manner.

III.2. Preprocessing

III.2.1. Modelling with Alias Maya^{TM}, or Topmod^{TM}

Even though some parametric equations, implicit functions, or mathematically defined scalar fields can be utilized in generating the high quality image with this volume painting system, the usage of commercial or proprietary software enables artists to create the object without any limitation. It also allows for modifications to the input at any time. Therefore, any commercial or proprietary software is desirable to assist in sculpting the object. All the models are created in Alias Maya^{TM} or Topmod^{TM} in this thesis and textured in Alias Maya^{TM} with two textures, one texture for color information and the other for directional information as shown in Figure 5.


File Edit Modify Create Display Window Edit Curves Surfaces Edit NURBS Polygons Edit Polygons Subdiv Surfaces MJ Poly Tools 1.3 Help
Modeling
General Curves Surfaces Polygons Subdivs Deformation Animation Dynamics Rendering PaintEffects Cloth Fluids Fur Hair Custom
View Shading Lighting Show Panels
Channels Object
Layers Options
Display
persp
No Character Set
DLFL Demo
File Display Renderer Object Selection

HelpLine: Displays short help tips for tools and selections

Mode: Extrude

Extrude: Extrude Face

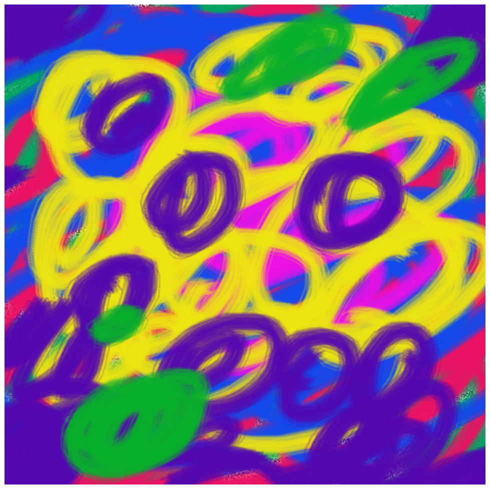
Length: 1.00

Rotation: 0

Scale: 1.00

Segments: 1

Fig. 4. Geometries Modelled and Textured in Alias Maya^{TM} and Topmod^{TM}


(a)

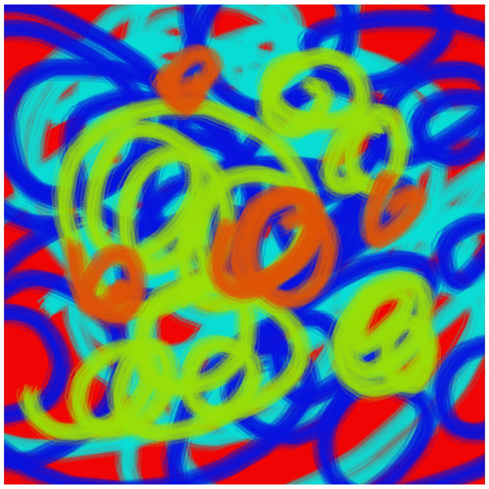

(b)

Fig. 5. Two Textures : (a) A Texture for Color Information, (b) A Texture for Directional Information

III.2.2. Voxelization

Voxelization is a process that creates a signed distance field. It assigns values to each voxel of the volume datasets. The signed distance field is composed of positive or negative float numbers and zero numbers. A positive sign indicates that the position of the voxel value is outside the given surface. A negative sign indicates that the position of the voxel value is inside the given surface. A zero means that the position of the voxel value is exactly at the boundary of the surface. Therefore, the absolute value of the positive or negative float numbers is the distance between the center of a voxel and the closest point on the surface. The smaller absolute value is closer to the boundary of the surface. The sign is determined by the dot product of the surface normal and the vector, which connects the center of a voxel to the closest point on the surface. Color and directional values can be assigned to a voxel by accessing the UV texture coordinates according to the quadratic equation for the triangle [15]. Finally, the volume dataset file, which has four components (voxel index, signed distance field, color, and directional information) is created. It can be rendered through the volume painting process.

III.3. Volume Painting

III.3.1. Volume Ray Casting

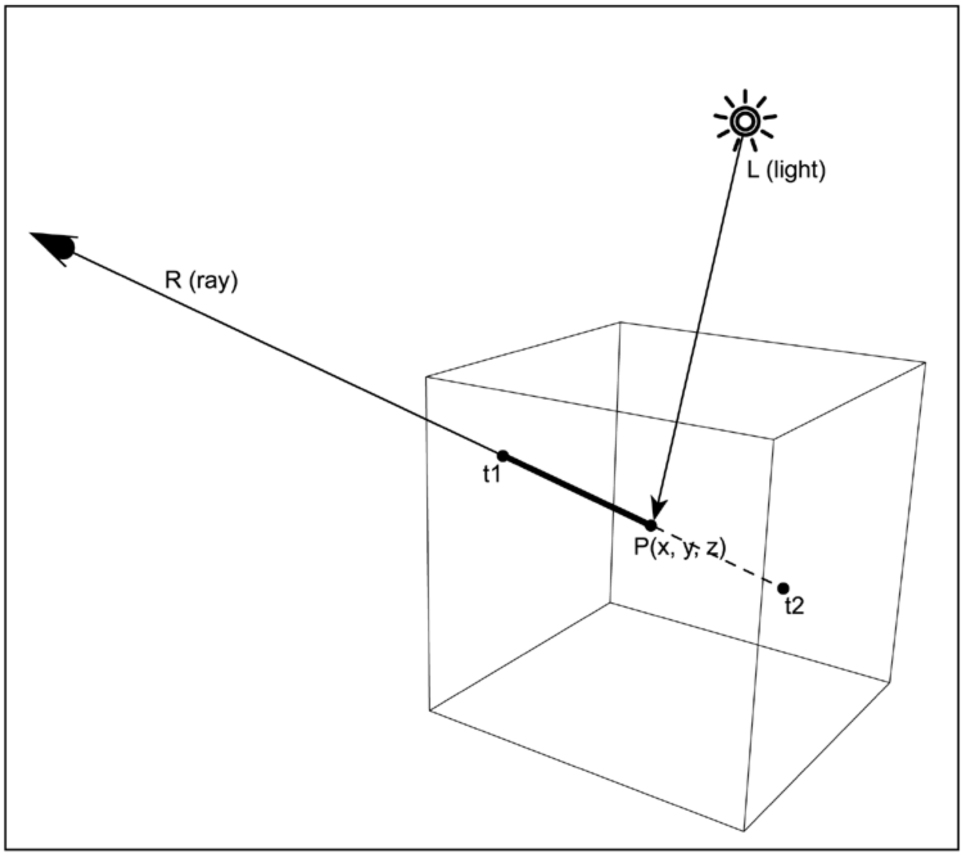


Fig. 6. Volume Ray Casting.

Ray casting is a rendering technique used to get a high-quality image of a solid geometry. It is a technique useful in volumetric rendering as well. Volume ray casting can create a high quality image with fewer limitations of other volume rendering techniques. Figure 6 shows how volume ray casting works. A ray put into a density function of three spatial variable (x, y, z) between t1 and t2. As a ray emits from the camera, it is passed through the block of voxels. Interpolated samples are taken along a ray for accumulation within a bounding box. The gradient is processed and the shading is evaluated. Since one pixel may represent hundreds of values sampled along a ray. These values need to be collected into one sample. Accumulation is achieved with a composition function.

$$C_{out} = C_{in} + (1 - \alpha_{in})C_i,$$

$$\alpha_{out} = \alpha_{in} + (1 - \alpha_{in})\alpha_i.$$

where C_{out}, C_{in} and C_i are all associated colors and α_{out} and α_{in} are associated alpha values. The function can create an early-termination when the opacity, α_{in}, reaches a value close to 0.9999. Instead of using this compositing method, we use of the bisection method to determine the boundary of the object surface. The bisection method divides the distance of the two temporary intersection points in half, the method places the surface point between the exterior and interior intersections until the distance of those two positions is less than a given threshold. The results of compositing and bisection method are shown in Figure 7 and Figure 8 respectively. Figure 7 is a image by the volume rendering with the compositing method of a simple sphere with Factional Brownian Motion [16] and Figure 8 is the result of the volume rendering with the bisection method of 3D Julia Set [17]. Fundamentally, the signed distance field consists of positive or negative float values which represent voxel values and zero values which represents the surface. As the ray goes inside the surface, the value become negative. This indicates that between the temporary intersection point which has the positive value and the temporary intersection point which has the negative value, exists the surface point or permanent intersection point. This signed distance field is also used to estimate the real geometry normal, which plays a vital role in determining the direction of the brush-stokes. Actually, the signed distance field is converted to the opacity value which ranges from 0 to 1 to grasp the normal for the shading.

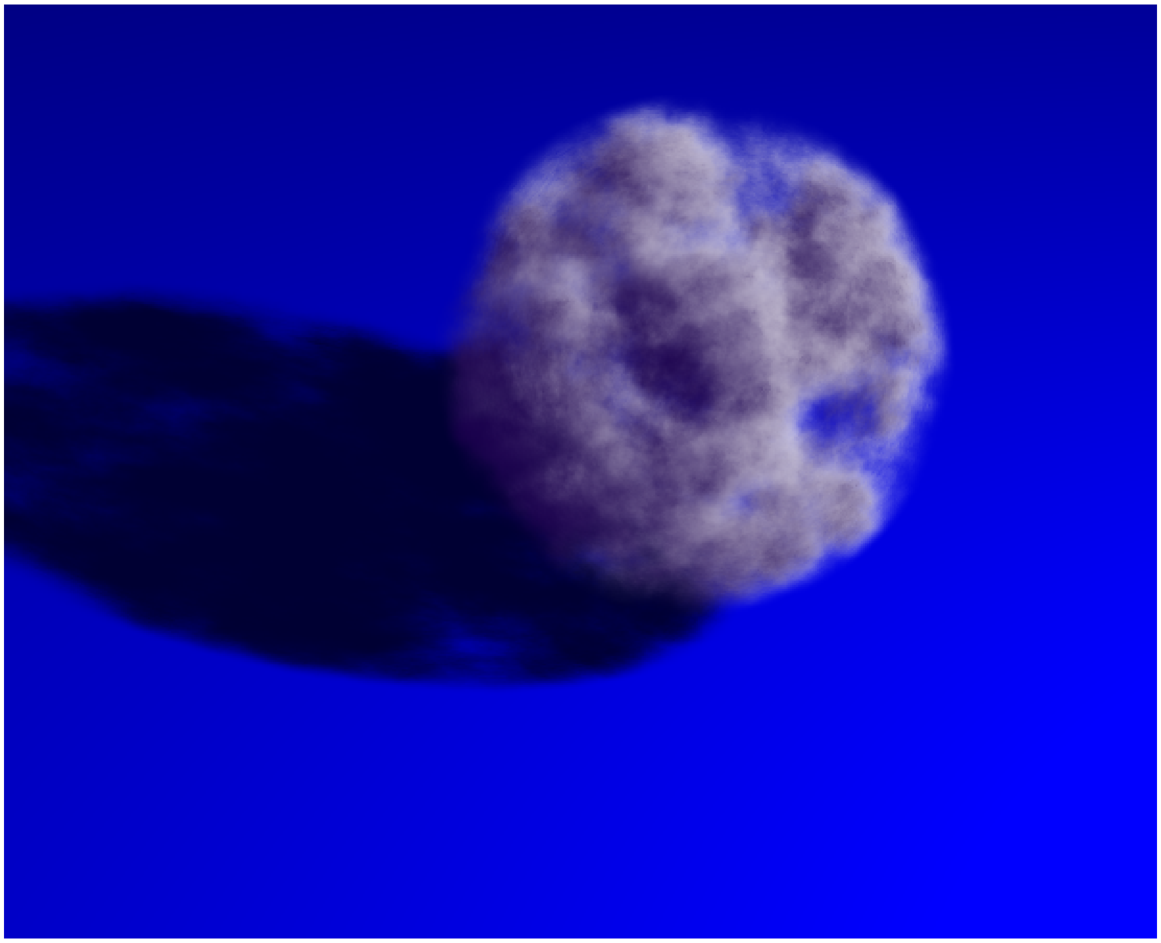


Fig. 7. Example of the Compositing : Compositing with Fractional Brownian Motion

III.3.2. Gradient Processing

The gradient is thought of as a surface normal in volume rendering. The estimate of the gradient requires a much different approach than obtaining the normal vector of a general solid geometry. In traditional computer graphics, a surface normal is needed to shade a pixel of a geometry that is rendered. In volume rendering, we do not have a surface normal

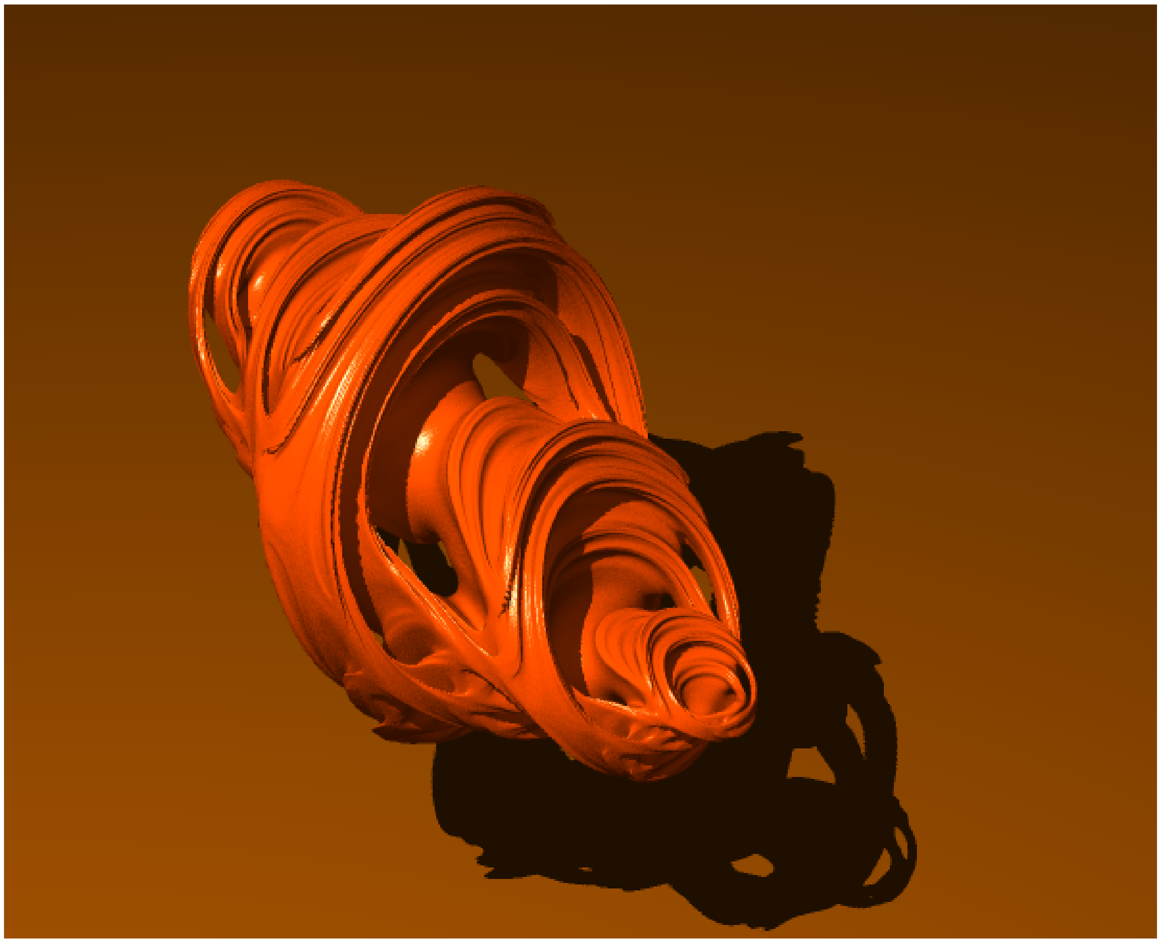


Fig. 8. Example of the Bisection : 3D Julia Set

available since generally we do not know where a surface is in the datasets. The gradient is a measure of how quickly the voxel density changes. It is a key component of the volume rendering pipeline, since the property of the material determines the density of the volume datasets. Therefore we can recognize what kind of geometry will be rendered. The gradient is a 3D vector, which is calculated by the gradient estimator with the interpolated densities of voxels in the volume datasets.

III.3.2.1. Central Difference Method

Central Difference Method is the most general and common gradient estimation method. It is not the best way in terms of the quality, but it is fast and easy to implement. Voxel density function, $f(x, y, z)$ is defined at the position of the volume datasets, (x, y, z). The gradient vector is defined as $\vec{N} = [Nx, Ny, Nz]$. Nx, Ny and Nz are components of a 3D gradient vector.

The Central Difference Method is defined as follows:

$$Nx = f(x-1, y, z) - f(x+1, y, z),$$

$$Ny = f(x, y-1, z) - f(x, y+1, z),$$

$$Nz = f(x, y, z-1) - f(x, y, z+1).$$

III.3.2.2. Geo-Dome Method

Geo-Dome is a spherical geometry which has vertices a unit-length away from its center. The gradient in the Central Difference Method is determined by the derivative of voxel densities. Therefore, if no derivative of the voxel densities exists, the gradient can not be estimated. Even in this situation, the Geo-Dome method can calculate the gradient by positioning the Geo-Dome at the surface point. It adds up all outside vectors, which have

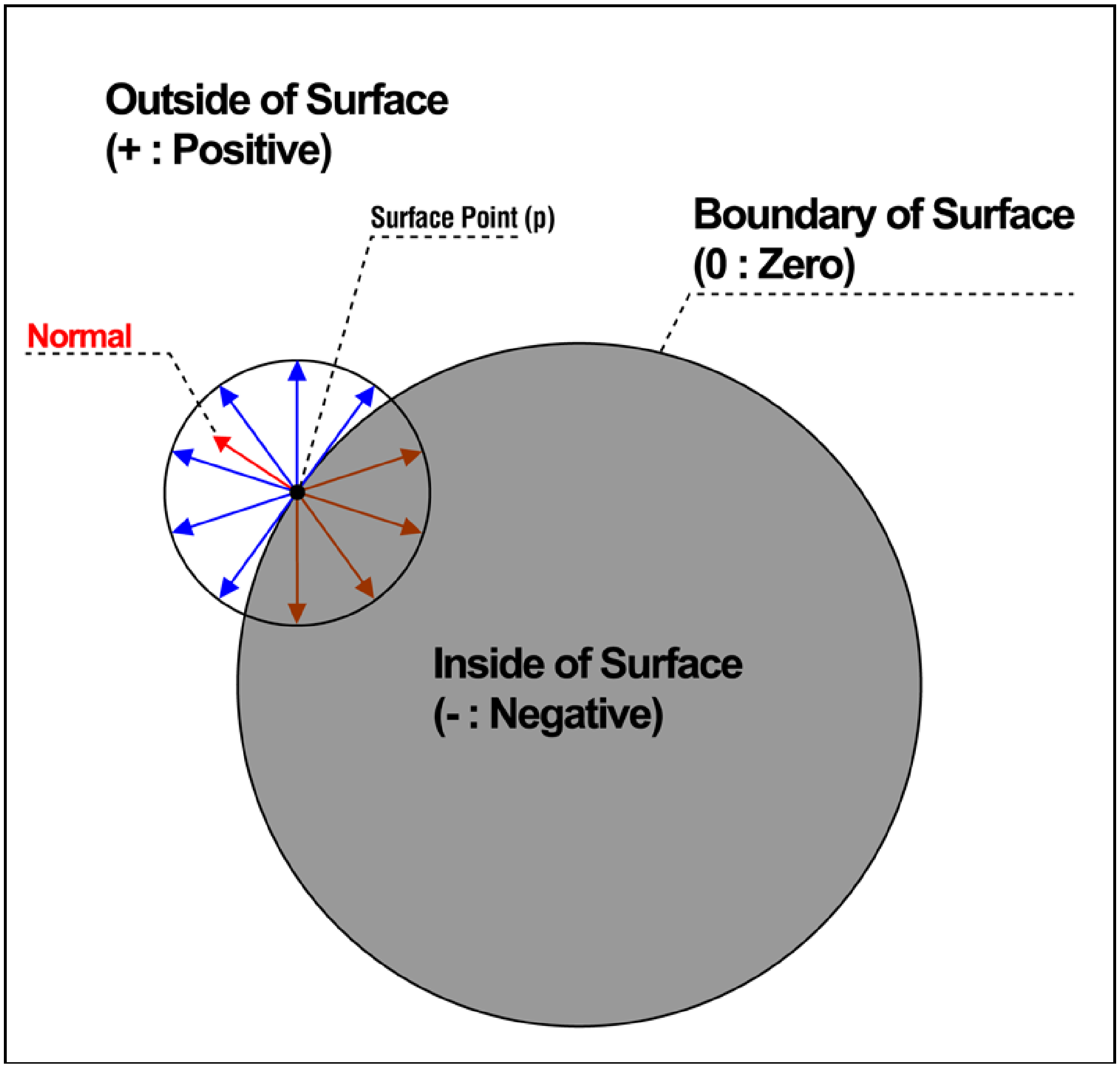
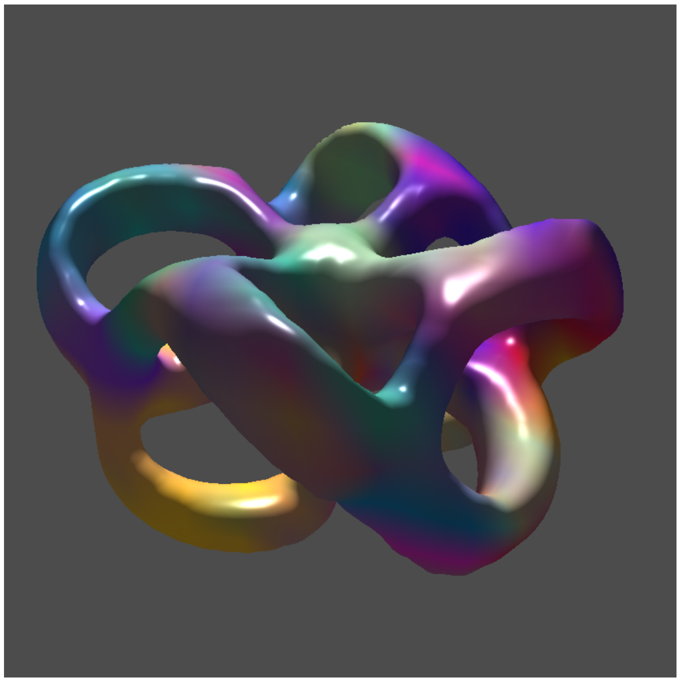


Fig. 9. Geo-Dome Method


(a)

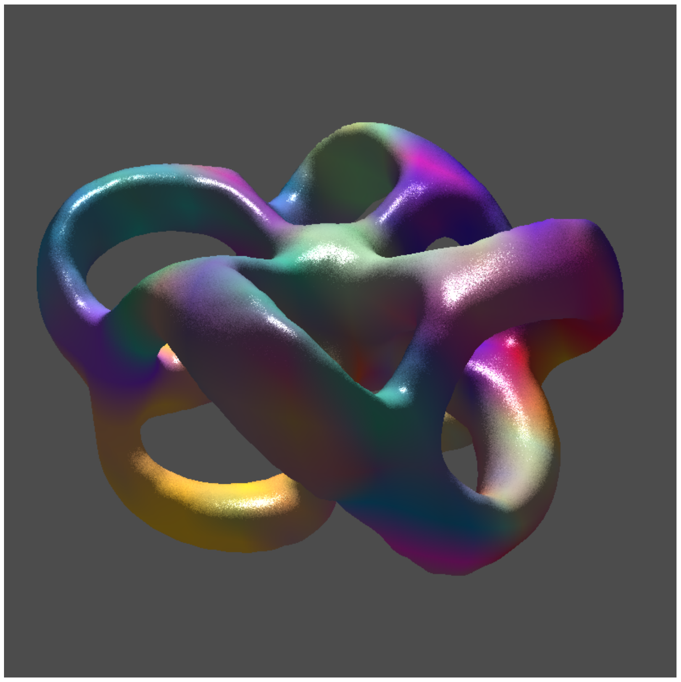

(b)

Fig. 10. Implicit Geometry : (a) Central Difference Method, (b) Geo-Dome Method

positive voxel values, and take away inside vectors, which have negative voxel values, as shown in Figure 9. This gradient vector should be normalized. Even though it may make some artifacts, because of the limitation of the number of the vertices of the Geo-Dome, it can be removed by rotating the Geo-Dome around a random axis and with a random magnitude. The result is a tolerably noisy. Figure 10 shows the difference between the Central Difference method and the Geo-Dome method. The Geo-Dome vertices are defined as follows : $V_n = [\vec{v}_0, \vec{v}_1, \ldots \vec{v}_i, \ldots \vec{v}_K]$. A given function f returns the voxel value of the Geo-Dome vertices V_n. The gradient $\vec{\nabla} f$ at an arbitrary voxel location $\mathbf{p} = (x, y, z)$ in the volume datasets is calculated by

$$\vec{\nabla} f \approx \sum_{i=0}^{K} a_i \vec{v}_i$$

where

$$a_i = 1 \quad \text{if } f(\vec{v}_i + \mathbf{p}) \geq 0$$

$$a_i = 0 \quad \text{otherwise.}$$

The normal vector is computed as

$$\vec{n} = \frac{\vec{\nabla} f}{|\vec{\nabla} f|}$$

III.3.3. Vector Field

A brush-stroke is arbitrarily directional information created by a painter on his canvas. The brush-stroke can be represented in computer graphics as a 3D vector. The 3D vector fields are key to implementing the 3D brush-stroke which has the information of motion. A variety of vector fields such as fluid, linear, vortex and the like, can be defined and visualized in the volume painting process. In this thesis, we are focusing on achieving an effect of the vector fields which are tangent to the surface. Such a vector field $\vec{v}$ can be obtained by the cross product with the gradient vector of the real geometry $\vec{n}$ and the

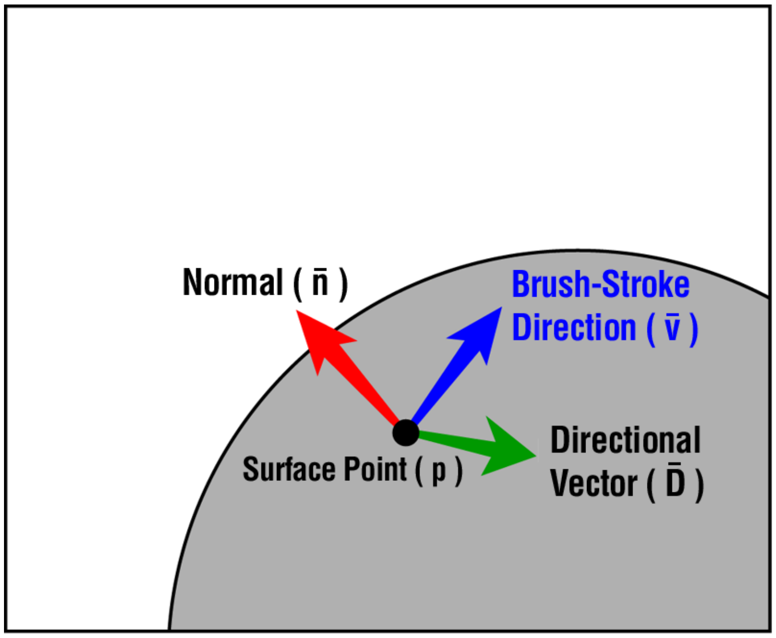


Fig. 11. Calculation of the Direction of a Brush-Stroke

directional vector $\vec{D}(s,t)$ as

$$\vec{v} = \vec{n} \times \vec{D}(s,t).$$

where s, t are texture coordinates computed from the $\mathbf{p}$. An example of $\vec{D}(s,t)$ is shown in Figure 11, which is defined previously with a texture map in the volume datasets.

III.3.4. Brush-Stroke

The brush-stroke is created when the surface intersection point is determined by the volume ray casting. Perlin noise [18] is used to create the brush-stroke effect on the surface as shown in Figure 12.

At first, we normalize P between 0 and 1 as follows: $h(\mathbf{p}) = \text{normalize}(P(w\mathbf{p}))$.

$$g(\mathbf{p}) = \frac{1}{N}\sum_{i=0}^{N} h(\mathbf{p_i})$$

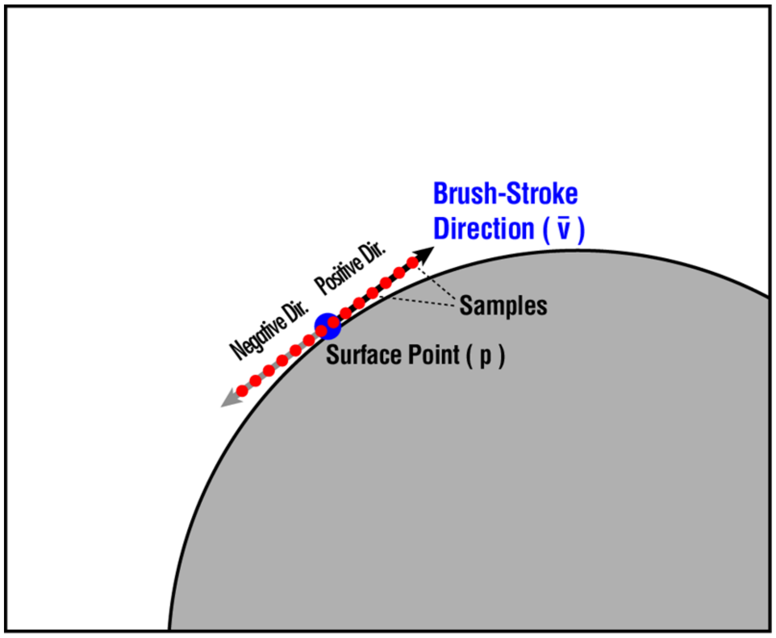


Fig. 12. Line Integral Convolution Along the Vector Field

where $P()$ is the Perlin noise function and w is the frequency of Perlin noise. $\mathbf{p}$ is the surface point and $\mathbf{p_i}$ are the positions of samples. N is the number of samples along the brush-stroke direction $(\vec{v})$.

We compute the opacity α in two different ways using g noise.

1. To create a painted object with solid internal structure, we use the following equation to compute the opacity: $\alpha = g(\mathbf{p}) + f(\mathbf{p})$.

2. To create a painted object with empty inside, we use the following equation to compute the opacity: $\alpha = g(\mathbf{p}) + |f(\mathbf{p}) - a|$.

where a is an arbitrary real number that changes the thickness of painted area, and f normalizes the signed distance field between 0 and 1. The surface point for the painted object $(\tilde{\mathbf{p}})$ is estimated by determining whether or not α is larger than a given threshold.

III.3.5. Illumination and Shading

Each voxel has color information defined by a texture map in the volume datasets. A color value convoluted along the brush-stroke direction $(\vec{v})$ becomes the diffuse color of the Phong illumination model [19] at the surface point for the painted object, $(\tilde{\mathbf{p}})$. The rendering equation is defined as follows:

$$I = k_a C(\tilde{\mathbf{p}}) + \sum_{i=1}^{n} [k_d C(\tilde{\mathbf{p}})(\vec{N} \cdot \vec{L}_i) + k_s O_s (\vec{R}_i \cdot \vec{V}^n].$$

where $C()$ collects the color information of the surface point for the painted object, $(\tilde{\mathbf{p}})$, from the volume datasets. I is the illumination color. k_a, k_d and k_s are the coefficients of the ambient, diffuse and specular surface shading color respectively. $\vec{N}$ is the normal vector . O_s is the specular surface shading color. $\vec{L}$ and $\vec{R}$ are a point light and a refection vector respectively.

CHAPTER IV

IMPLEMENTATION AND RESULTS

IV.1. Design of Structure

Volume painting technique is implemented by the following tools :

- **Alias Maya™ and Topmod™.**
- **Standard C and C++.**

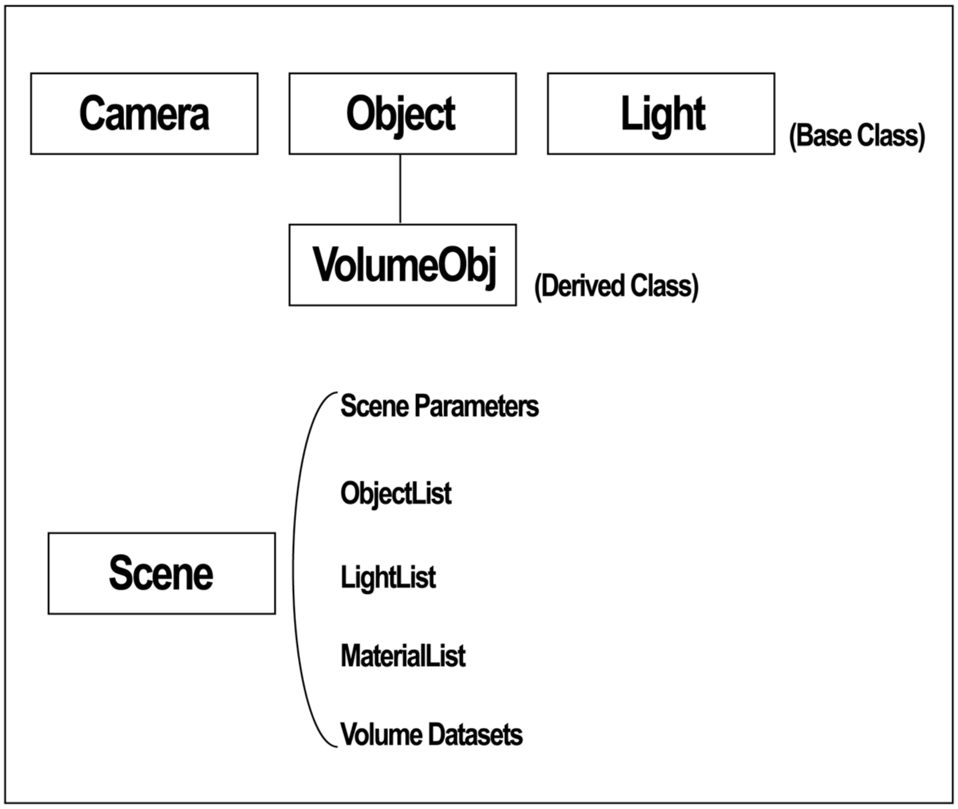
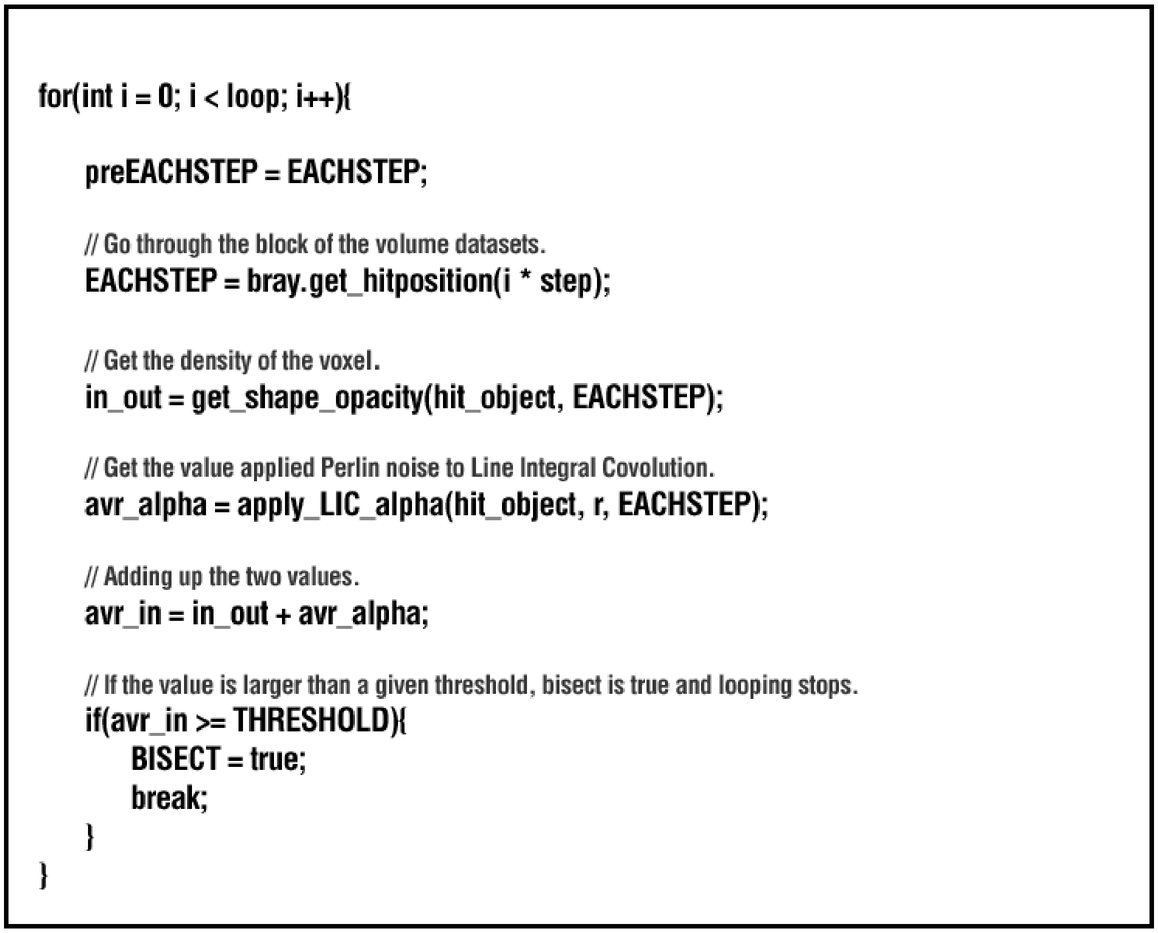


Fig. 13. Structure of Basic Classes

Basically, the program is composed of four basic classes and the volumetric object. The volumetric object is derived from the basic object class as shown in Figure 13. The scene class has most of the Volume Painting functions, scene parameter variables and the object list, light list, material list, and volume datasets. As a ray from the camera goes into



```
for(int i = 0; i < loop; i++){

    preEACHSTEP = EACHSTEP;

    // Go through the block of the volume datasets.
    EACHSTEP = bray.get_hitposition(i * step);

    // Get the density of the voxel.
    in_out = get_shape_opacity(hit_object, EACHSTEP);

    // Get the value applied Perlin noise to Line Integral Covolution.
    avr_alpha = apply_LIC_alpha(hit_object, r, EACHSTEP);

    // Adding up the two values.
    avr_in = in_out + avr_alpha;

    // If the value is larger than a given threshold, bisect is true and looping stops.
    if(avr_in >= THRESHOLD){
        BISECT = true;
        break;
    }
}
```

Fig. 14. Example of C++ Code

the volume block of datasets, the above function estimates the real boundary by the addition of the voxel density to the value of the Perlin noise applied with the LIC, which creates the new value. Then we need to see if the new value is within our threshold parameters as shown in the sample code of Figure 13. This completes the calculation process for our non-photo realistic volume painting surface.

IV.2. Results

Volume Painting is a novice rendering technique which simulates the brush-stroke of the traditional painting. The brush-stroke is controlled by the vector field obtained from a texture map that the user defines. Moreover, the width of a brush-stroke can be adjusted adequately with the parametrization of the frequency of Perlin noise. The expressive effects of the brush-stroke makes the image look like an artistic painting in 3D space. Furthermore, it can be animated by simulating the vector field. It can also be evaluated in terms of an aesthetic viewpoint.

Figure 15, Figure 16, and Figure 17 are the results of Volume Painting. Figure 18 shows the various brush-stroke sizes by adjusting the frequency of Perlin noise. Figure 19, Figure 20, and Figure 21 visualize the difference of a general volume rendering technique and Volume Painting.

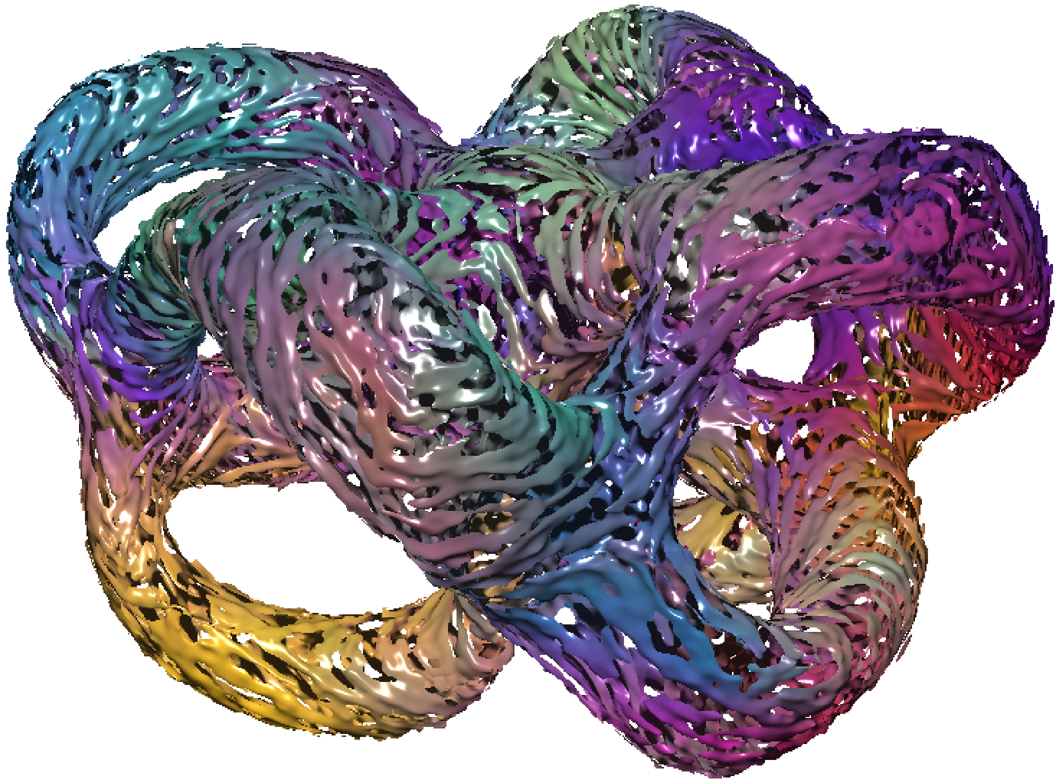


Fig. 15. Implicit Object 1.

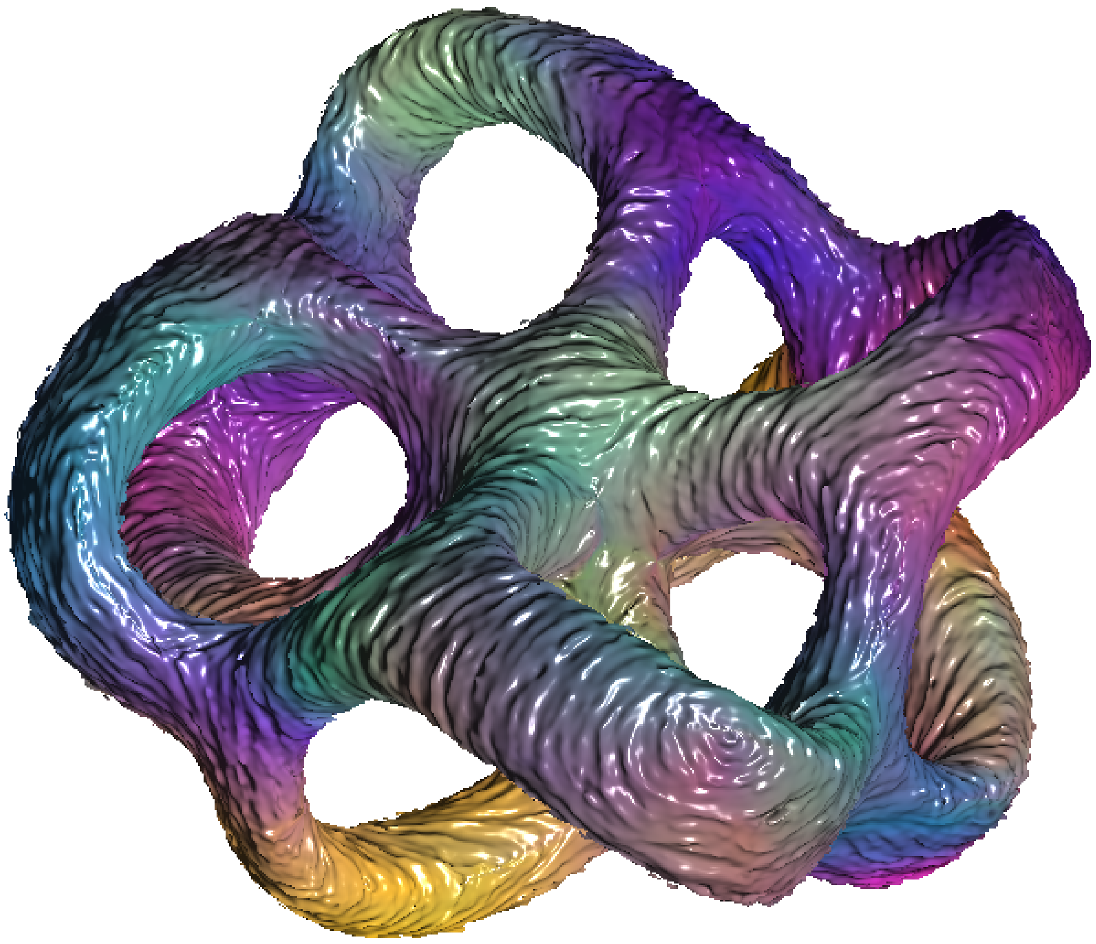


Fig. 16. Implicit Object 2.

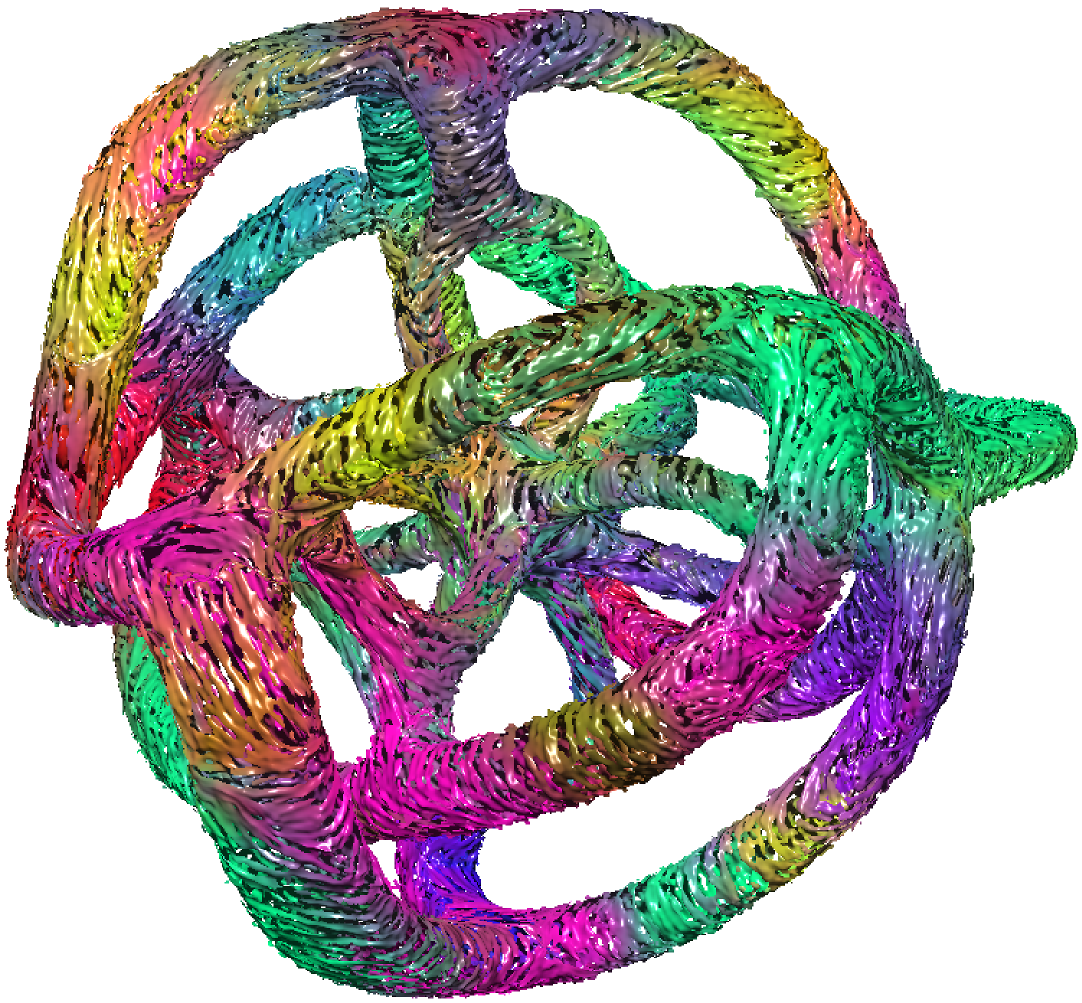


Fig. 17. Implicit Object 3.

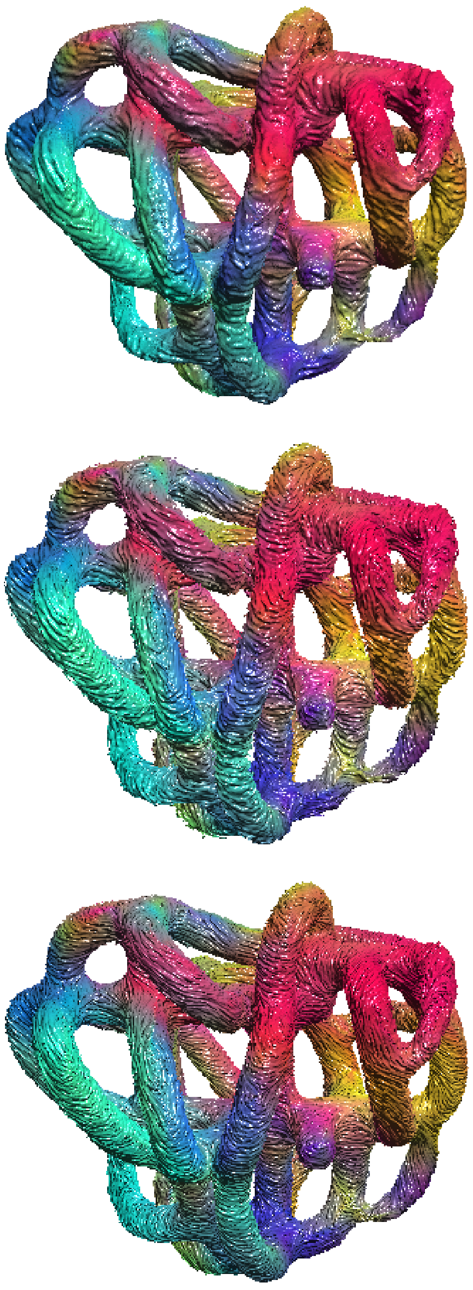


Fig. 18. Various Widths of the Brush-Strokes.

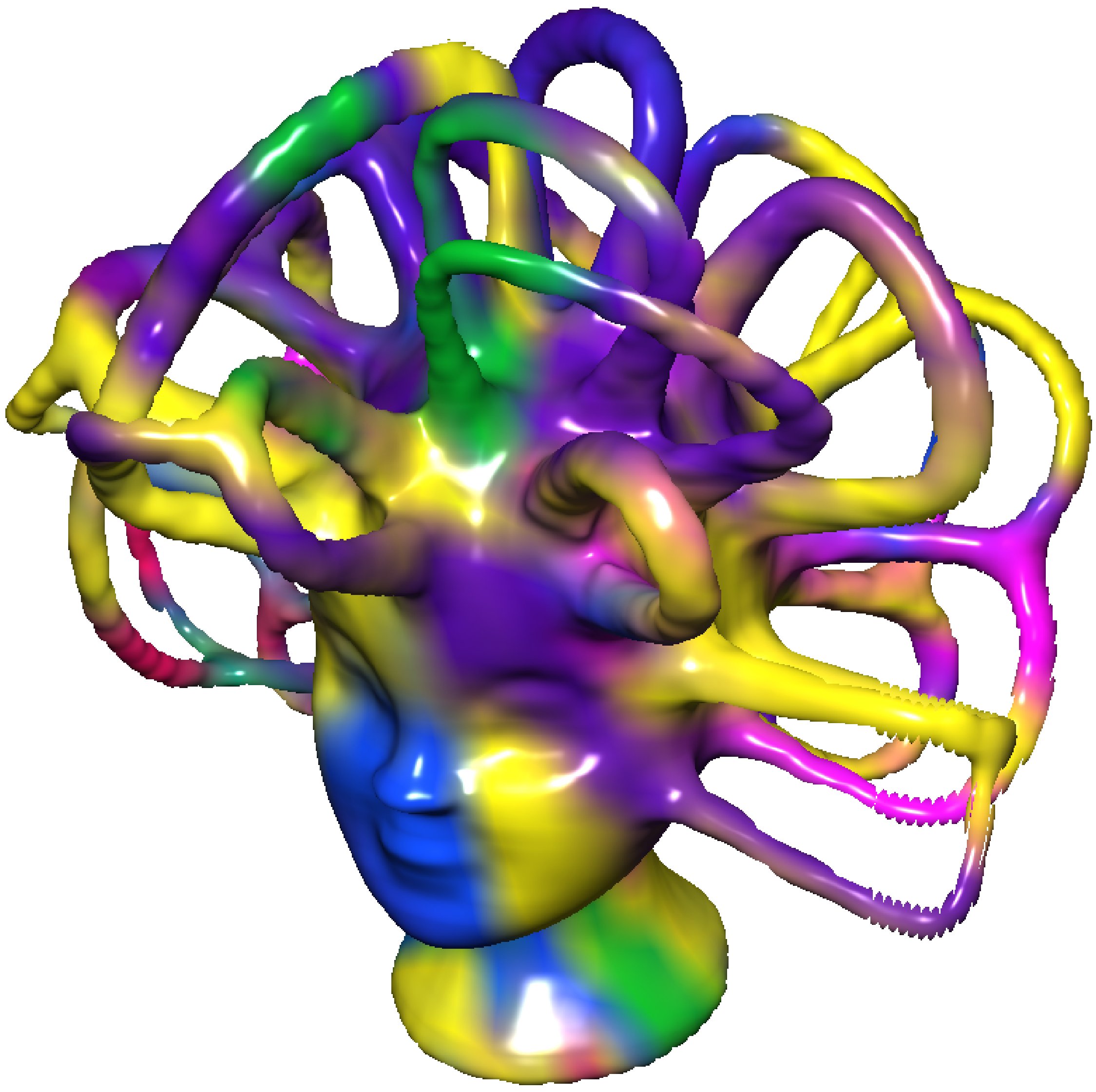


Fig. 19. Simple Volume Rendering of a Girl’s Head.

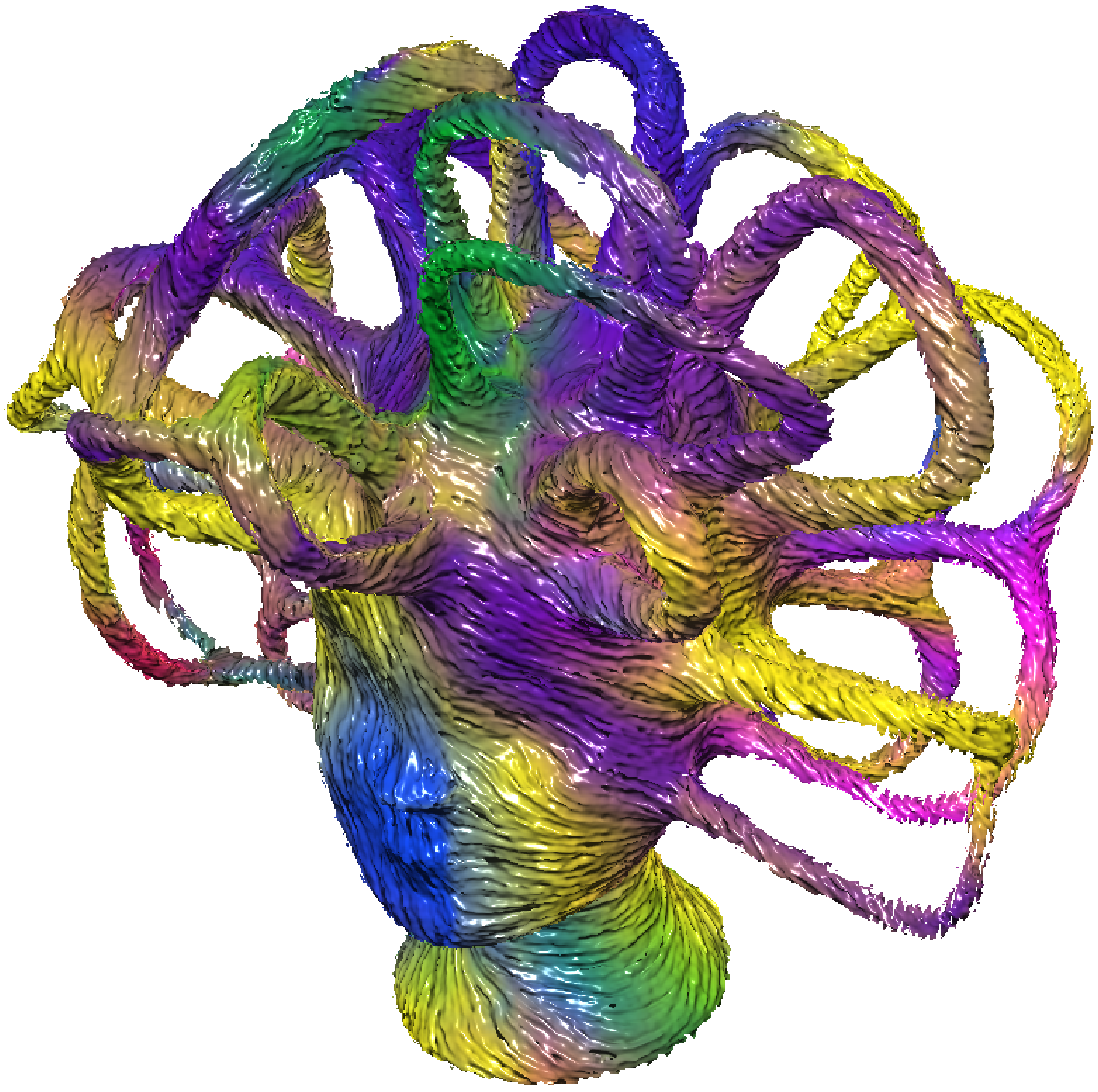


Fig. 20. Volume Painting of a Girl's Head 1.

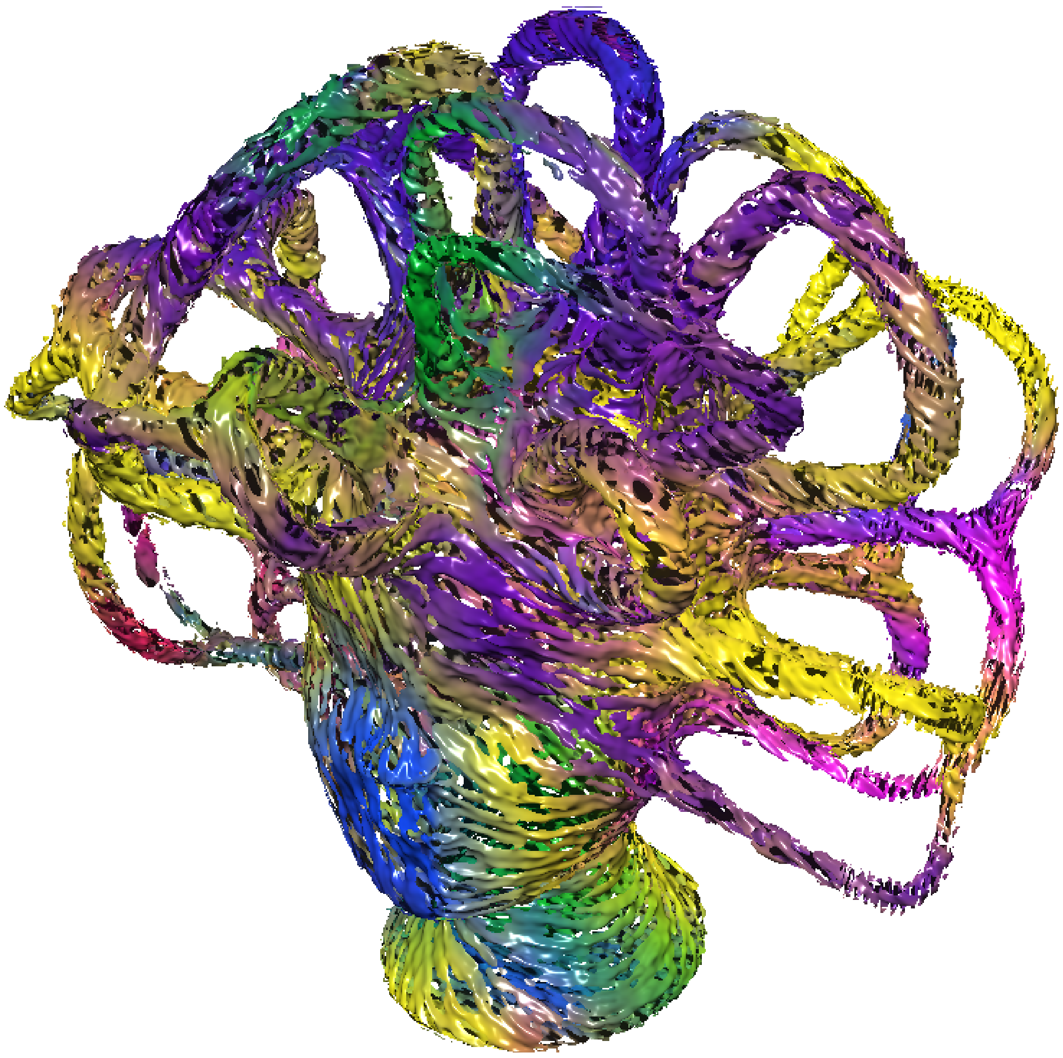


Fig. 21. Volume Painting of a Girl’s Head 2.

CHAPTER V

CONCLUSION AND FUTURE WORK

NPR techniques are prevalent in the computer graphics world. There are many different NPR techniques which are available in across areas in the field. This currently is a popular area for developing new techniques. Although volume rendering techniques are still used mainly as photorealistic rendering in various areas, its application makes a variety of possibilities of NPR availible. This thesis is intended to realize of one of those possibilities. This thesis shows that the combination with volume rendering and LIC can produce a gorgeous result from the viewpoint of both technical and artistic areas of research.

In future, this rendering system can be improved to reduce the computation speed, improve the image quality and implement real time interactivity. Additionally, a user friendly interface can also be developed.

REFERENCES

[1] H. Shen, C. Johnson, and K.-L. Ma, “Visualizing Vector Fields Using Line. Integral Convolution and Dye Advection,” in *Proceedings of the 1996 Volume Visualization Symposium*, 1996, pp. 63-70.

[2] B. Cabral and L. Leedom, “Imaging Vector Fields Using Line Integral Convolution,” in *Computer Graphics Proceedings, Annual Conference Series*, 1993, pp. 263-269.

[3] A. Lu, C. J. Morris, D. S. Ebert, P. Rheingans, and C. Hansen, “Non-Photorealistic Volume Rendering Using Stippling Techniques,” in *Proceedings IEEE Visualization ’02*. IEEE, 2002, pp. 211-218.

[4] O. Deussen and T. Strothotte, “Computer-Generated Pen-and-Ink Illustration of Trees,” in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, 2000, pp. 13-18.

[5] B. J. Meier, “Painterly Rendering for Animation,” in *Proceedings SIGGRAPH ’96*. ACM, 1996, pp. 477-484.

[6] C. J. Curtis, S. E. Anderson, J. E. Seims, K. W. Fleischer, and D. H. Salesin, “Computer Generated Watercolor,” in *Proceedings SIGGRAPH ’97*. ACM, 1997, pp. 421-430.

[7] A. Hertzmann, “Painterly Rendering with Curved Brush Strokes of Multiple Size,” in *Computer Graphics Proceedings, Annual Conference Series*, 1998, pp. 453-460.

[8] M. Levoy, “Display of Surfaces from Volume Data,” *IEEE Computer Graphics and Applications*, vol. 8, no. 3, pp. 29-37, 1988.

[9] R. Westermann and T. Ertl, “Efficiently Using Graphics Hardware in Volume Rendering Applications,” in *Proceedings SIGGRAPH ’98*. ACM, 1998, pp. 169-176.

[10] J. Kniss, G. Kindlmann, and C. Hansen, “Interactive Volume Rendering Using Multi-Dimensional Transfer Functions and Direct Manipulation Widgets,” in *Proceedings IEEE Visualization ’01*. IEEE, 2001, pp. 255-262.

[11] V. Interrante and C. Grosch, “Strategies for Effectively Visualizing 3D Flow with Volume LIC,” in *Proceedings IEEE Visualization ’97*. IEEE, 1997, pp. 421-424.

[12] V. Interrante, “Illustrating Surface Shape in Volume Data via Principal Direction Driven 3D Line Integral Convolution,” in *Proceedings SIGGRAPH ’97*. ACM, 1997, pp. 109-116.

[13] D. Ebert and P. Rheingans, “Volume Illustration: Non-photorealistic Rendering of Volume Models,” in *Proceedings IEEE Visualization ’00*. IEEE, 2000, pp. 195-202.

[14] E. Akleman, J. Chen, and V. Srinivasan, “A New Paradigm for Changing Topology During Subdivision Modeling,” in *Proceedings Pacific Graphics ’00*, 2000, pp. 192-201.

[15] P. J. Schneider and D. H. Eberly, *Geometric Tools for Computer Graphics*, San Francisco: Morgan Kaufmann Publishers, 2003.

[16] A. A. Apodaca, L. Gritz, *Advanced Renderman - Creating CGI for Motion Pictures*, San Francisco: Morgan Kaufmann Publishers, 2000.

[17] J. Hart, D. Sandin and L. Kauffman, “Ray Tracing Deterministic 3-D Fractals,” in *Proceedings SIGGRAPH ’89*. ACM, 1989, pp. 289-296.

[18] K. Perlin, “Improving Noise,” in *Proceedings SIGGRAPH ’02*. ACM, 2002 pp. 681-682.

[19] B.-T. Phong, “Illumination for Computer Generated Pictures,” *Communications of the ACM*, vol. 18, no. 6, pp. 311-317, 1975.

VITA

Jaewook Lee
C414 Langford Center 3137 TAMU
College Station, TX 77843-3137
jaewooklee2001@yahoo.com

Education

M.S. in Visualization Sciences	Texas A&M University, August 2005
B.F.A. in Graphic Design	Pusan National University, February 1997