SAVONIA AMMATTIKORKEAKOULU

# Implementation of volume rendering in C# for LightningChart


## Alexey Tukalo


Bachelor's Thesis

May 18, 2016 ________________________

Bachelors degree (UAS)

**SAVONIA UNIVERSITY OF APPLIED SCIENCES** **THESIS Abstract**

| | | | |
|---|---|---|---|
| Field of Study<br>Technology, Communication and Transport | | | |
| Degree Programme<br>Degree Programme in Information Technology | | | |
| Author<br>Alexey Tukalo | | | |
| Title of Thesis<br>Implementation of volume rendering in C# for LightningChart | | | |
| Data | May 18, 2016 | Pages/Appendices | 14 |
| Supervisor<br>Arto Toppinen | | | |
| Client Organization/Partners<br>Arction Oy | | | |

Abstract

Arctive Oy is a Finnish software company, based in Kuopio. Their main product is LightningChart, the fastest C# framework for the visualisation of scientific, engineering, trading and research data. The library contains a bunch of tools for visualisation of XY, 3D XYZ, smith and polar graph, 3D pie/donut views, 3D objects.

The company wanted to extend the LightingChart's abilities of polygonal 3D models rendering by volume rendering. It gives Arction an opportunity to attract new clients to the product. In result the framework provides a unique possibility to render volume and polygonal models at same visualisation.

The project started from a literature research and comparison of different volume visualisation techniques, to choose the best one for the Arction's case and implement it inside the framework. The implementation of the volume rendering engine is based on DirectX used together with C# via SharpDX API and HSLS shader language for low level optimisation of rendering calculations.

The final chapter of the report contains an evaluation of the results and suggestion for a future development of the engine.

Keywords
Visualisation, Ray Casting, 3D, C#, LightningChart, DirectX, HLSL, Image Processing, Volume Rendering, Rendering

ACKNOWLEDGEMENTS

I am very thankful to Arction Oy for offering me an opportunity to take part in the development of the project. I really like the office atmosphere and freedom in terms of my working style and schedule allowed by the company.

My special thanks go to Mr. Pasi Toummainen, CEO of the company, who expressed interest in my idea to extend the library by the volume rendering engine, gave me permission to work on the project and guided me especially in the very early part of the development process.

Moreover, I would like to say thank you to my supervisor of thesis, Arto Toppinen, for his mentoring and support during the report writing stage of my work.

In addition, I would like to express my deepest gratitude Karlsuruhe Institute of Technology, there I got the first experience with volume rendering via Ray Casting. I am especially grateful to Nicolas Tan, Jerome, who was my mentor during the part of my internship related to modification of Tomo Ray Caster 2 and to Aleksandr Lizin, the creator of the volume rendering engine based on WebGL.

# Contents

# Chapter 1

# Introduction

This chapter contains brigh information about the motivation bihind volume rendering, my personal baground in computer graphics especially volume rendering. It allso introduce Acrtion the owner of the project, examplains the reasons of Arction's interest in the development, set reqerments for the final product.

## 1.1 Motivation

Volume data is very common our day. An importance of the type of datasets will grow in the near future, because of development in the field of 3D data acquisition and possibilities to perform the visualisations on a modern office workstation with an interactive frame rate.

Volume rendering is a process of multi-dimensional data visualisation into a two-dimensional image which gives the observer an opportunity to recognize meaningful insights in the original information. The technology allows us to represent 3 dimensions of the data via position in a 3D space and 3 more via color of the point.

The dataset can be captured by various numbers of technologies like: MRI[1], CT[2], PET[3], or USCT[4]. They also can be produced by physical simulations, for example fluid dynamics. Volumetric information plays a big role in medicine for an advanced cancer detection, visualization of aneurisms and treatment planning. This kind of rendering is also very useful for non-destructive material testing via computer tomography or ultrasound. Geoseismic researches produce huge three-dimensional datasets. Their visualisations are used in an oil exploration and planning of the deposit development.

[1]Magnetic resonance imaging
[2]Computer tomography
[3]Positron emission tomography
[4]Ultrasound computer tomography

## 1.2 Personal backgound

I received my first experience in the visualisation of volumetric data during my internship at the Institute of Data Processing and Electronics, which belongs to the Karlsruhe Institute of Technology (KIT). I was a part of the 3D Ultrasound Computer Tomography (USCT) team. Their main goal is the development of a new methodology for early breast cancer detection. During the work placement I had to develop an algorithm to visualise five-dimensional datasets. In result the algorithm was integrated into Tomo Ray Caster 2[5] and USCT's edition of DICOM Viewer.

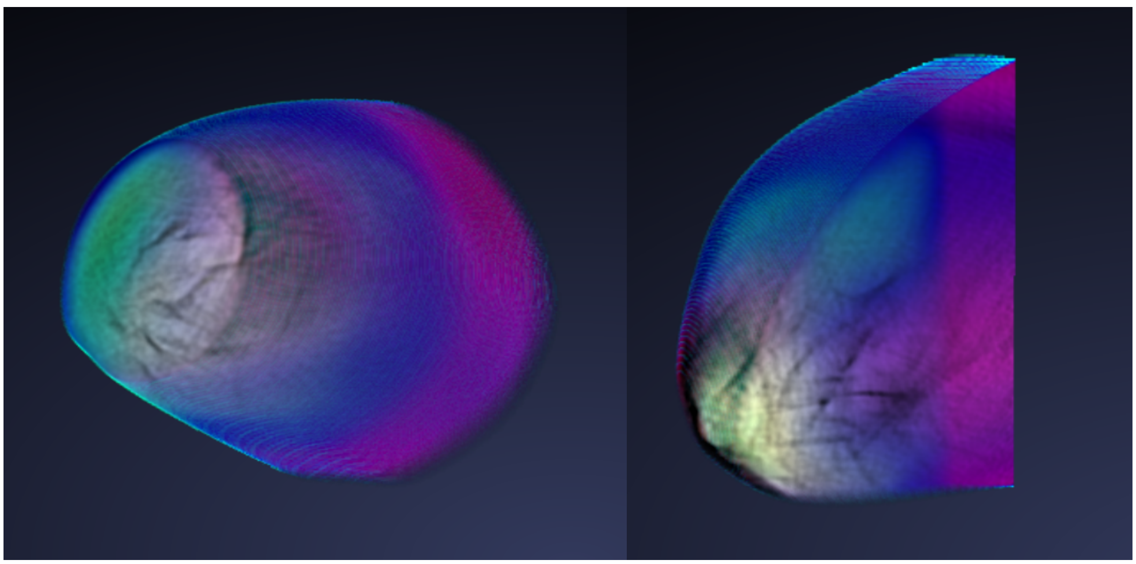

Figure 1.1: Volume visualisation of breast phantom made by USCT

During the project I made my very first steps in modern computer graphics. I got my first experience in work with WebGL during customisation of the Tomo Ray Caster, learned GLSL, my first shader language, I also gained a lot of knowledge about image processing and scientific data visualisation, which became the basis for my thesis work.

[5] JavaScript framework for the visualisation of 3D data, developed in Institute of Data Processing and Electronics

## 1.3 Arction Oy and Ligthning Chart

Arction Oy is a Finnish software company based in Kuopio. Their team has a strong background in computer graphics and science. The main product of the company called LightningChart Ultimate. It is the fastest C# library for scientific and engineering data visualisation. The library is capable to draw massive XY, Polar, Smith and 3D XYZ graphs, polygonal mesh models, surfaces, 3D pies/donuts and Geographic information.The library has an API for .NET WinForm and WPF applications, it is also possible to use it for a traditional Win32 C++ software development. The main advantage of the library is the fact that it is based on low-level DirectX graphics routines developed by Arction, then the most part of competitors use graphics routines which belongs to System.Windows.Media.

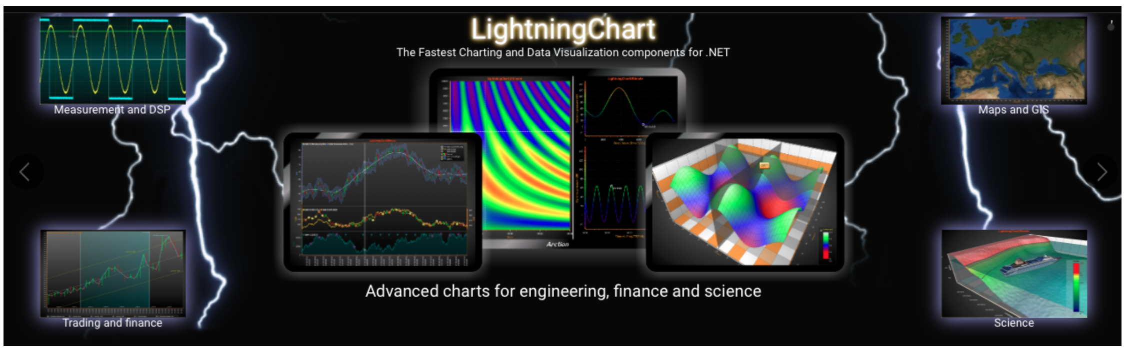


Figure 1.2: Example of Lightning Chart possibilities from the main page of Acrtion

## 1.4 Project Goals

So, as you can conclude from the previous section, LightningChart is very advanced software for 3D rendering based on polygons and lines and I came up with an idea to extend it by the special rendering engine for visualisation of volumetric data. It will give Arction's clients the unique possibility to combine visualisation of volume datasets with a wide range of other 3D possibilities of the library.

The rendering engine must be able:

- to render large multi-dimensional volumes with an interactive frame rate.
- to move and rotate the model in the chart's space.
- to provide clients with possibilities to apply windowing and thresholding to the initial dataset.

- to render the model semi-transparent.

Basically, this tool will give end user possibilities to change the contrast and brightness of the model's visualisation for better recognition of tiny details and make areas, which are out off certain range totally, transparent, it will also reveal insights into the internal structure of the model to the user via semi-transparency.

# Chapter 2

# Theory

This chapter explains the theory behind the project. It should introduce the main consepts of computer graphics, specify difference between polygonal meshmodel and volume rendering. It also contains overview of different volume rendering technique with their advantages and disadvantages in terms of speed, final image quality, flexibility and other implementation issues.

## 2.1 Rendering

At this Visualisation of 3D object as 2D image called rendering. Usually, 2D image is based pixel. In this case of grayscale pucture, it is two-dimensional array and the value of the array elements represents brightness of correspondent pixels on the screen. Configuration of colored images are dependend from color model, the most popular one is RGB, it represents image as three different grayscale images for three different colors called chanells. In case of RGB color models the images contains Red, Green and Blue values, sometimes it also keeps an information about Opacity and the chanel called Alpha.

Color model is mathematical abstrasction which allows computers to calculate brightness of corespondent point on the screen. RGB is original one for modern computer graphics, because it represents colors in the way they are physicly repdroduced on screen. There are several other color models. They have thier own advanges, for example some of them gives us an advenced editing possibilities while others represent physucal characteristic of different types of output devises like printers.

Multidimensinal data can be represented in two different ways: as surface and as volume. Future in this chapter we are going to talk about this two consepts a little bit closer. We will deffine their advantage and disadvantages, common and uncommon features. Moreover we are going to disscuss an implementation ditalies of the techniques on modern hardware.

## 2.2 Polygonal Rendering

Today we are litirally suraunded by surface rendering based on polygonal mesh. The technology is used on computer games, design, cinema, science enigering and etc. The technology is so popular that entier 3D graphic pipeline is build around the idea.

Traditionally, 3D surfaces are constructed out of huge amount of polygons connected to a mesh. Due to simplisity, they usually have triangular shape. It is possible to describe traingle via list of three coordinates called vertexs. The internal area of the shape filled by color during rastarization step. The color is calculated as dot product between the surface normal vector and the vector of light.

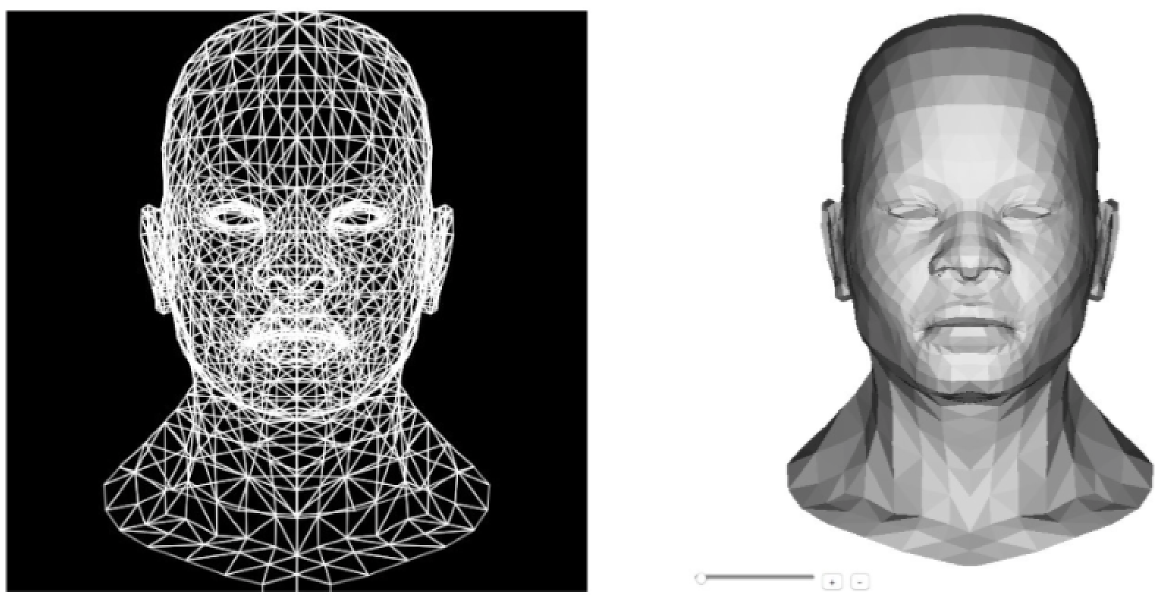

Figure 2.1: Wireframe and flat shading polygonal meshmodel

Surface normal vector of triangle can be calculated as a cross product of two triangle's sides, but it will give an acceptable result only for very flat surfaces. That's why curve serfaces usually contains an aditional normal vectors for every vertex. The are able to significantly improve detalisation of the model. The information kept in them is used during shading and tesseletion of the model's geometry.

## 2.3 Volume Rendering

### 2.3.1 Indirect

### 2.3.2 Direct

#### Texture-based

#### Ray Casting

#### Splatting

#### Shear-warp

# Chapter 3

# Implementation

## 3.1 Tools

### 3.1.1 C#

### 3.1.2 DirectX 11

#### Redering Pipeline

#### HLSL

### 3.1.3 SharpDX

### 3.1.4 LightningChart Ultimate

## 3.2 Visualisation process

### 3.2.1 Loading and preprocessing of dataset

### 3.2.2 Multi-pass rendering

#### First pass

#### Second pass

#### Empty space skipping

#### Ray function

# Chapter 4

# Conclusion

## 4.1 Results

### 4.1.1 Rotation and position

### 4.1.2 Settings

#### Windowing

#### Thresholding

#### Slice range clipping

### 4.1.3 Mouse picking

## 4.2 Disscusion

## 4.3 Future Development

# Chapter 5

# Appendix