View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Repository@USM

SEGMENTATION ASSISTED OBJECT DISTINCTION FOR DIRECT VOLUME RENDERING

By

ARASH AZIM ZADEH IRANI

Thesis is submitted in fulfillment of the requirements

For the degree of

Doctor of Philosophy

June 2013

ACKNOWLEDGEMENTS

First and before all I would like to remember ALLAH (SWT) and thank him for helping me through every stage of my life including this thesis. My best regards goes to my family for their patience and support. My sincere appreciation hereby goes to Dr.Bahari Belaton for the respect and confidence he vested in me and for the freedom he gave me. Lastly, I thank all my friends and colleagues for their kind company.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS ii

TABLE OF CONTENTS iii

LIST OF FIGURES vii

LIST OF TABLES xiv

LIST OF ABBREVIATIONS xv

LIST OF PUBLICATIONS xvi

ABSTRAK xvii

ABSTRACT xx - xxii

CHAPTER 1: INTRODUCTION

1.1 Introduction 1

1.2 Motivation 2

1.3 Problem Statement 2

1.4 Objectives 5

1.5 Contributions 6

1.6 Background 7

1.7 Outline 14

1.8 Chapter Summary 14

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction 15

2.2 Ray Casting 17

2.3 Shear-Warp 20

2.4 Splatting 28

2.5 Texture Mapping and GPU Driven Acceleration 31

2.6 Multi-dimensional Transfer Functions for Robust volume Rendering 33

2.7 Segmentation Based Direct Volume Rendering 36

2.8 K-means Based Image Segmentation 41

2.9 Discussion 45

2.9 1 An Evaluation of Direct Volume Rendering Literature 46

2.9.2 A Critical Approach to Advanced Volume Classification Methodologie 48

2.9.3 A Justification for K-means Clustering Utility 50

2.9.4 An Analysis of Reconstruction Filters for Volume Rendering 51

2.10 Chapter Summary 55

CHAPTER 3: OVERALL RESEARCH FRAMEWORK

3.1 Introduction 57

3.2 An Integrated Visualization Design 57

3.3 Chapter Summary 61

CHAPTER 4: K-MEANS BASED HYBRID SEGMENTATION DESIGN

4.1 Introduction 62

4.2 Preprocessing 65

4.3 Unsupervised Neural Map for Cluster Center Initialization 65

4.4 K-means Clustering 69

4.5 Statistical Optimization 70

4.6 Edge Detection and Analysis 72

4.7 Spatial Adjustment 74

4.8 Test and Evaluation 75

4.8.1 Case Study: Color Image Segmentation 77

4.9 Chapter Summary 80

CHAPTER 5: AN ENHANCED RENDERING ARCHITECTURE

5.1 Introduction 83

5.2 Directed Objects Sets Construction 86

5.3 Focus / Context Separation 89

5.4 Global Composition Mode 89

5.5 Interpolation Design and High Resolution Boundaries 92

5.6 Texture Assignment and Local Rendering 97

5.7 Ray Casting: A Model Based Approach 98

5.8 Test and Evaluation 103

5.8.1 Case Study 1: Abdomen – Pelvis 108

5.8.2 Case Study 2: Frog 127

5.8.3 Case Study 3: Heart – Abdomen 141

5.8.4 Case Study 4: Heart 157

5.9 Chapter Summary 164

CHAPTER 6: CONCLUSION 166

REFERENCES 169

APPENDIX A: SOFTWARE ENGINEERING

A1. Software Architecture 177

A2. Configuration Issues 180

A3. Implementation 181-190

LIST OF FIGURES

	Page
Figure 1.1 Shortcomings of ray casting	3
Figure 1.2 (a) Overlapping classification	4
Figure 1.2 (b) Imprecise transfer function design	4
Figure 1.3 An ambiguous interpolation result originates from multiple vector parameters	5
Figure 1.4 Enlistment of contributions	7
Figure 1.5 Correlation among computer graphics, imaging and visualization	8
Figure 1.6 A Typical visualization course	9
Figure 1.7 Ray casting based direct volume rendering	11
Figure 2.1 Virtual sampling	18
Figure 2.2 Shifted sampling in object order ray casting	19
Figure 2.3 Subdivision of cube into 24 corresponding tetrahedral	25
Figure 2.4 (a) General configuration	26
Figure 2.4 (b) Parallel configuration	26
Figure 2.4 (c) Tilted configuration	26
Figure 2.5 (a) Ordinary ray casting	28
Figure 2.5 (b) Shear-warp algorithm	28
Figure 2.6 Radial or sphere kernel is used for projection	29
Figure 2.7 Number of points used for splatting directly affects the rendering results	29
Figure 2.8 Angled slices are aligned to image plane	31
Figure 2.9: The behavior of primary data value (f), gradient magnitude (f') and second directional derivative (f'')	35
Figure 2.10 Steps of a modeling approach to three dimensional object reconstruction	39

Figure 2.11 Two conceptual levels of volume rendering 40

Figure 2.12 Segmentation architecture 43

Figure 2.13 Perceptual tower 45

Figure 2.14 Interrelationship among pieces of the literature leading to this work 50

Figure 2.15 (a) 2D sampling of space domain 53

Figure 2.15 (b) 2D sampling of frequency domain 53

Figure 2.16 (a) Pre-aliasing 55

Figure 2.16 (b) Post-aliasing 55

Figure 3.1: An overall view of the integrated visualization design 58

Figure 4.1 K-means based hybrid segmentation design 63

Figure 4.2 Interrelationship between phases 1.3 and 1.4 64

Figure 4.3 (a) Network initialization 66

Figure 4.3 (b) Neighborhood adjustments 66

Figure 4.4 Pseudo code of an interval computation for map node initialization 68

Figure 4.5 Diagonal directions include top left, top right, bottom left and bottom right 70

Figure 4.6 Graphical representation of ratio's nominator 71

Figure 4.7 (a) A 3×3 neighborhood before recombination 75

Figure 4.7 (b) After recombination 75

Figure 4.8 (a) Whale color image 78

Figure 4.8 (b) A 14×14 neural map 78

Figure 4.8 (c) Segmentation result 78

Figure 4.9 (a) Boat color image 79

Figure 4.9 (b) A 14×14 neural map 79

Figure 4.9 (c) This work's segmentation result 79

Figure 4.9 (d) Segmentation result 79

Figure 4.10 (a) Butterfly color image 80

Figure 4.10 (b) A 14×14 neural map 80

Figure 4.10 (c) This work's segmentation result 80

Figure 4.10 (d) Segmentation result 80

Figure 5.1 An enhanced rendering architecture 84

Figure 5.2 A typical object set array 87

Figure 5.3 IDs 1 and 2 represent two objects at the same depth while IDs 3 and 4 represent two objects at different depths 87

Figure 5.4 Pseudo code of object sets list construction 88

Figure 5.5 Double rendering design 89

Figure 5.6 Pseudo code of double rendering algorithm 91

Figure 5.7 Tri-linear interpolation 93

Figure 5.8 One dimensional nearest neighbor approximation 95

Figure 5.9 Three dimensional nearest neighbor approximation 95

Figure 5.10 Universal threshold construction 97

Figure 5.11 Texture assignment table 98

Figure 5.12 (a) Optical model of ray casting 100

Figure 5.12 (b) Implementation model of ray casting 100

Figure 5.13 (a) single level implementation of ray casting 102

Figure 5.13 (b) Double level implementation of ray casting 102

Figure 5.14 Sagittal and coronal and axial directions of MRI scan 107

Figure 5.15 Short and long directions of MRI scan 108

Figure 5.16 Chest and lung at 600% magnification (a) Ray casting 111

Figure 5.16 Chest and lung at 600% magnification (b) This work 111

Figure 5.17 Bone and liver at 600% magnification (a) Ray Casting 112

Figure 5.17 Bone and liver at 600% magnification (b) This work 112

Figure 5.18 Ray casting's interpolation error verses this work's interpolation error 116

Figure 5.19 Chest and lung gray level slabs and their corresponding segmentations 117

Figure 5.20 (a) Raw MRI data and its histogram before preprocessing 118

Figure 5.20 (b) Segmentation of part a 118

Figure 5.20 (c) Raw MRI data and its histogram after preprocessing 118

Figure 5.20 (d) Segmentation of part c 118

Figure 5.21 Volumetric visualization of chest and lung based on Figure 5.19 119

Figure 5.22 Volumetric visualization of chest and lung via standard ray casting 120

Figure 5.23 Kidney and liver gray level slabs and their corresponding segmentations 121

Figure 5.24 Volumetric visualization of kidney and liver based on Figure 5.23 122

Figure 5.25 Volumetric visualization of kidney and liver via standard ray casting 123

Figure 5.26 pelvis gray level slabs and their corresponding segmentations 124

Figure 5.27 Volumetric visualization of pelvis based on Figure 5.26 125

Figure 5.28 Volumetric visualization of pelvis via standard ray casting 126

Figure 5.29 Gray level slabs from frog's top layers and their corresponding segmentations 129

Figure 5.30 Gray level slabs from frog's middle layers and their corresponding segmentations 130

Figure 5.31 Gray level slabs from frog's middle layers and their corresponding segmentations 131

Figure5.32 Gray level slabs from frog's bottom layers and their corresponding Segmentations 132

Figure 5.33 Gray level slabs from frog's bottom layers and their corresponding Segmentations 133

Figure 5.34 Volumetric visualization of whole frog in vertical position 134

Figure 5.35 Volumetric visualization of whole frog from horizontal position 135

Figure 5.36 Volumetric visualization of whole frog from an angle 136

Figure 5.37 Volumetric visualization of whole frog from bottom 137

Figure 5.38 A close up volume of the frog's head structure and eyeballs 138

Figure 5.39 A close up volume of the frog's chest 138

Figure 5.40 Volumetric visualization of whole frog from side 139

Figure 5.41 Volumetric visualization of whole frog from belly 139

Figure 5.42 Volumetric visualization of whole frog in vertical position 140

Figure 5.43 Volumetric visualization of whole frog skin 140

Figure 5.44 Sagittal slabs of heart-abdomen. Left ventricle, right ventricle, descending thoracic aorta, chest and liver are segmented 142

Figure 5.45 Radiologist annotation of sagittal heart-abdomen slabs 143

Figure 5.46 Volumetric visualization of left ventricle, right ventricle, thoracic aorta and liver 144

Figure 5.47 Sagittal slabs of heart-abdomen. Thoracic aorta, left atrium, aortic root, right atrium, liver and chest are segmented 145

Figure 5.48 Radiologist annotation of sagittal heart-abdomen slabs 146

Figure 5.49 Volumetric visualization of thoracic aorta, left atrium, aortic root, right atrium and liver 147

Figure 5.50 Axial slabs of heart-abdomen. Liver, chest, skin, flesh and heart are segmented 148

Figure 5.51 Radiologist annotation of axial heart-abdomen slabs 149

Figure 5.52 Volumetric visualization of right ventricle, liver, skin and chest 150

Figure 5.53 Coronal heart-abdomen slabs. Heart, liver, flesh, skin and lung are segmented 151

Figure 5.54 Radiologist annotation of coronal heart-abdomen slabs 152

Figure 5.55 Volumetric Visualization of heart, liver and lung 153

Figure 5.56 Coronal heart-abdomen slabs. Partial heart, kidney, spinal cord, lung and flesh are segmented 154

Figure 5.57 Radiologist annotation of coronal heart-abdomen slabs 155

Figure 5.58 Volumetric visualization of heart, kidney, spinal cord and lung 156

Figure 5.59 Short heart slabs. Heart, lung, liver, flesh and skin are segmented 158

Figure 5.60 Radiologist annotation of short heart slabs 159

Figure 5.61 Volumetric visualization of liver, heart, lung and skin 160

Figure 5.62 Long heart slabs. Heart, spinal cord and lung are segmented 161

Figure 5.63 Radiologist annotation of long heart slabs 162

Figure 5.64 Volumetric visualization of heart 163

Figure A1 Overall software design 177

Figure A2 Configuration of patented VTK libraries 181

Figure A3 Classes, functions and variables for the neural map 183

Figure A4 Classes, functions and variable for K-means based parsing 184

Figure A5 Classes, functions and variables for progressive display and analysis 185

Figure A6 Classes, functions and variables for histogram equalization and display 186

Figure A7 (a) Inheritance diagram for vtkVolume16Reader 187

Figure A7 (b) Collaboration diagram for vtkVolume16Reader 187

Figure A8 (a) Inheritance diagram for vtkStructuredPointsReader 188

Figure A8 (b) Collaboration diagram for vtkStructuredPointsReader 188

Figure A9 (a) Inheritance diagram for vtkLookupTable 189

Figure A9 (b) Collaboration diagram for vtkLookupTable 189

Figure A10 (a) Inheritance diagram for vtkVolumeRayCastFunction 190

Figure A10 (b) Collaboration diagram for vtkVolumeRayCastFunction 190

LIST OF TABLES

	Page
Table 2.1 The literature review structure	15
Table 2.2 An instance of Hounsfield values	37
Table 2.3 Comparison among direct volume rendering techniques	47
Table 3.1 This work's distinctive features (an extension of table 2.3)	59
Table 4.1 Experimental list	76
Table 4.2 Characteristics of dataset	76
Table 5.1 Experimental list	104
Table 5.2 Characteristics of datasets	105
Table 5.3 Characteristics of abdomen-pelvis experimental figures	110
Table 5.4 Comparison between tri-linear interpolation and this work's interpolation method	114
Table 5.5 Characteristics of frog experimental figures	128

LIST OF ABBREVIATIONS

ID	Identity Tags
DVR	Direct Volume Rendering
DF	Diagonal Factor
MIP	Maximum Intensity Projection
NPR	Non Photo Realistic Rendering
RGBA	Red, Green, Blue, Alpha
HSV	Hue, Saturation, Value
GPU	Graphic Processing Unit
LOD	Level of Details
RLE	Run Length Encoding
ROI	Region of Interest
CUDA	Compute Unified Device Architecture
AI	Artificial Intelligence
BCC	Body Centered Cartesian
2D	Two Dimensional
3D	Three Dimensional
PGM	Portable Gray Map
JAI	Java Advanced Imaging
VTK	Visualization Tool Kit
API	Application Programming Interface
TCL	Tool Command Language

LIST OF PUBLICATIONS

1. Irani, A. A. Z. Belaton, B. (2009) A K-means Based Generic Segmentation System. In: IEEE Proceedings. Sixth International Conference on Computer Graphics, Imaging and Visualization, 11/14 August 2009, Tianjin, China. p.300-307.

2. Irani, A. A. Z. Belaton, B. (2012) An Automated Adaption of K-means Based Hybrid Segmentation System into Direct Volume Rendering Object Distinction Mode for Enhanced Visualization Effect. In: IEEE Proceedings. Ninth International Conference on Computer Graphics, Imaging and Visualization, 24/27 July 2012, Hsinchu, Taiwan. p.62-69.

PENGASINGAN OBJEK DIBANTUKAN SEGMENTASI UNTUK PERSEMBAHAN VOLUM LANGSUNG

ABSTRAK

Penyurihan sinar adalah teknik persembahan volum langsung untuk memperlihatkan tatasusunan 3D data sampel. Ia mempunyai aplikasi penting dalam pengimejan perubatan dan biologi. Walau bagaimanapun, ia terdedah kepada keputusan klasifikasi yang bercelaru. Ia juga terdedah kepada pertindihan nilai fungsi pindah dan ketidakupayaan mekanisme menghuraikan voksel yang mantap untuk pengasingan objek. Dalam penyelidikan ini, kami mencadangkan satu pendekatan berasaskan pemprosesan imej untuk meningkatkan keupayaan teknik penyurihan sinar dalam proses membezakan objek. Rangka kerja penyurihan sinar diubahsuai untuk menampung maklumat keahlian objek yang dijana oleh algorithma segmentasi hibrid berdasarkan K-min. Maklumat keahlian objek dalam bentuk tag ID diberikan kepada bucu-bucu kubus. Satu penimbal intra-objek dirangka dan diselaraskan dengan penimbal antara-objek untuk membolehkan modul persembahan sejagat menerapkan pelbagai proses persembahan setempat (sekunder). Persembahan bermulti-fasa ini memberikan dua kelebihan ke atas kaedah persembahan sejagat. Pertama, operasi interpolasi dan komposisi kini berorientasikan kedalaman -- kaedah interpolasi adaptif (trilinear atau jiran terdekat) digunakan berdasarkan bilangan objek yang terdapat pada pelbagai tahap kedalaman volum, pada waktu yang sama peningkatan LOD (tahap butiran) bagi objek yang dikehendaki. Semuanya ini mengoptimumkan bilangan pengiraan matematik yang diperlukan. Kedua, penyetempatan reka bentuk fungsi pindah yang membolehkan penggunaan fungsi

pemindahan binari (tidak bertindih) untuk penetapan warna dan kelegapan. Tag ID objek dihasilkan melalui satu set teknik pemprosesan imej yang diolah secara kreatif menerusi reka bentuk algoritma segmentasi hybrid berasaskan K-min. Kaedah prapemprosesan seperti penapis laluan tinggi/laluan rendah dan penyamaan histogram digunakan jika perlu untuk penyingkiran hingar dan pembetulan keharmonian. Rangkaian neural tanpa pengawasan digunakan untuk menentukan pusat kelompok dan meningkatkan ketepatan kelompok. Nisbah penentu Fisher digunakan untuk menukar kumpulan awalan (dihasilkan menerusi beberapa larian pengelompokan K-min) ke dalam segmen-segmen optimum bagi mewakili warna yang wujud dan ciri-ciri pembentukan. Pengesanan pinggir digunakan untuk menyingkirkan sempadan penting daripada penggabungan semula yang tidak relevan. Penggabungan semula dan faktor toleransi epsilon digunakan untuk penalaan spatial. Lima kajian kes digunakan untuk menguji dan menganalisis kerja yang dicadangkan di dalam tesis ini. Bongkah-bongkah Axial abdomen-pelvis diperolehi dari Hospital Universiti Sains Malaysia, bongkah-bongkah menegak katak diperolehi dari Lawrence Berkeley National Laboratory, bongkah-bongkah sagittal, korona dan axial jantung-perut dan bongkah-bongkah panjang dan pendek jantung yang diperolehi dari kumpulan penyelidikan MRI Auckland digunakan sebagai dataset bebas tahap kelabu bagi empat kajian kes. Satu lagi kajian kes menggunakan imej warna yang diperolehi daripada kerja penyelidikan Microsoft. Empat jenis penilaian yang terdiri daripada perbandingan cebis demi cebis terhadap penyurihan sinar, perbandingan menyeluruh dengan teknik persembahan rujukan, perbandingan cebis demi cebis terhadap segmentasi rujukan dan perbandingan berasaskan objek terhadap segmen-segmen yang diekstrak telah dijalankan. Keputusan eksperimen menyimpulkan bahawa penyelesaian ini mampu mengurangkan artifak dalam dataset

isipadu yang kompleks - kabur dan organ berganda yang bertindih. Hakikat ini ditunjukkan oleh perbandingan cebis demi cebis pada tahap pembesaran. Ketepatan interpolasi diperbaiki oleh penapis yang dibina semula yang dicadangkan dari nilai purata kesilapan relatif. Keseluruhan bilangan persampelan yang diperlukan dan operasi komposisi dikurangkan oleh persembahan berorientasikan kedalaman ke atas voxel yang ditanda. Walaupun sasaran utama kerja penyelidikan ini adalah imej tahap kelabu dari bidang perubatan/biologi, imej warna juga digunakan untuk menunjukkan kelebihan reka bentuk segmentasi ini untuk mengendalikan imej warna.

SEGMENTATION ASSISTED OBJECT DISTINCTION FOR DIRECT VOLUME RENDERING

ABSTRACT

Ray casting is a direct volume rendering technique for visualizing 3D arrays of sampled data. It has vital applications in medical and biological imaging. Nevertheless, it is inherently open to cluttered classification results. It suffers from overlapping transfer function values and lacks a sufficiently powerful voxel parsing mechanism for object distinction. In this research work, we are proposing an image processing based approach towards enhancing ray casting technique's object distinction process. The ray casting architecture is modified to accommodate object membership information generated by a K-means based hybrid segmentation algorithm. Object membership information is assigned to cubical vertices in the form of ID tags. An intra-object buffer is devised and coordinated with inter-object buffer, allowing the otherwise global rendering module to embed multiple local (secondary) rendering processes. A local rendering process adds two advantageous aspects to global rendering module. First, depth oriented manipulation of interpolation and composition operations that lead to freedom of interpolation method choice based on the number of available objects in various volumetric depths, improvement of LOD (level of details) for desired objects and reduced number of required mathematical computations. Second, localization of transfer function design that enables the utilization of binary (non-overlapping) transfer functions for color and opacity assignment. A set of image processing techniques are creatively employed in the design of K-means based hybrid segmentation algorithm. Preprocessing methods

such as high pass / low pass filters and histogram equalization are optionally used for noise removal and harmony rectification. An unsupervised neural network is used to initialize cluster centers and improve the clustering accuracy. Few rounds of K-means clustering are performed to identify preliminary groups. Fisher discriminant ratio is used to convert preliminary groups into optimum segments representative of inherent color and formation characteristics. Edge detection is used to exempt important boundary / pattern information from irrelevant recombination. Recombination and epsilon tolerance factor are used for spatial tuning. Five case studies are established for testing and analysis of this work's contributions. Axial slabs of abdomen-pelvis obtained from Universiti Sains Malaysia's hospital, vertical slabs of frog obtained from Lawrence Berkeley National Laboratory, sagittal, coronal and axial slabs of heart-abdomen and long and short slabs of heart obtained from Auckland MRI research group are used as independent gray level datasets for four case studies. Another case study uses color images obtained from a Microsoft collaborated research work. Four types of evaluations comprising of piecewise comparison against standard ray casting, whole some comparison against a bench mark rendering, piecewise comparison against annotated benchmark and object based comparison of extracted segments were carried out. Experimental results conclude that the solution effectively reduces artifacts of complicated volumetric datasets which are noise prone and overlapping. This fact is emphasized by piecewise comparisons at high level of magnification. The interpolation accuracy is improved by the proposed reconstruction filter in terms of average relative error. Overall, the number of required sampling and composition operations is largely reduced by depth oriented rendering of tagged voxels. Even though medical / biological gray level images are the main target of this work, color images are used

to demonstrate the advantageous aspect of the segmentation design in respect to handling color images.

CHAPTER 1

INTRODUCTION

1.1 Introduction

Given the great dependency of human analysis and judgment abilities on realistic and sensible visualization, appropriate volume rendering techniques for effective visualization may be deemed fundamental. Most of the currently available direct volume rendering methods such as ray casting, shear-warp, splatting and texture mapping employ transfer function for object distinction. Non- photorealistic based volume rendering (NPR) is introduced as another way towards perceptual object visualization [Csebfalvi et al. 2001; Lu et al. 2002]. However, a major drawback of using these methods is that all voxels of a volumetric data set are treated in an identical manner without using any priori information that specifies object membership on a per-voxel basis. Inability to properly distinguish among multiple objects of interest solely based on transfer function is often the case. Segmentation is a formidable approach for handling this problem. It infers object membership information for each object of interest, yielding a tag for each volume's voxel.

This chapter is broken into two main parts. Part 1 includes Sections 1.1 through 1.5 and strives to clearly and briefly convey what is this work all about, what problem(s) is it solving and what is it trying to achieve. Part two includes Section 1.6. It provides a background review of relevant visualization, computer graphics and image processing concepts and techniques which are fundamental towards understanding and scrutiny of this thesis.

1.2 Motivation

Enhanced visualization effect could be made possible by proper integration of imaging merit and computer graphic techniques. So far, consideration of an integrated visualization frame work that could enable the adaption of a viable segmentation design into a particular rendering architecture has been an open domain for innovative solutions. Present literature on segmentation based volume visualization suggests significantly exclusive approaches, yielding a gap for generic solutions with reasonable outcome for handy visualization tasks. Automation is seldom advertised if not ignored as compared to none or semi automatic frame works.

1.3 Problem Statement

The first question that comes to the mind of a person who intends to enhance or improve a particular technique may be that, what are the shortcomings of that technique. Only by knowing the shortcomings or weaknesses of a technique one could propose a relevant solution. As may be noticed from Figure 1.1, in order to highlight or demonstrate the overall problem that we are going to solve within this work, we have divided the weakness of ray casting technique into three main groups which are cluttered classification, cumbersome transfer function design and ambiguity and wasted resources.

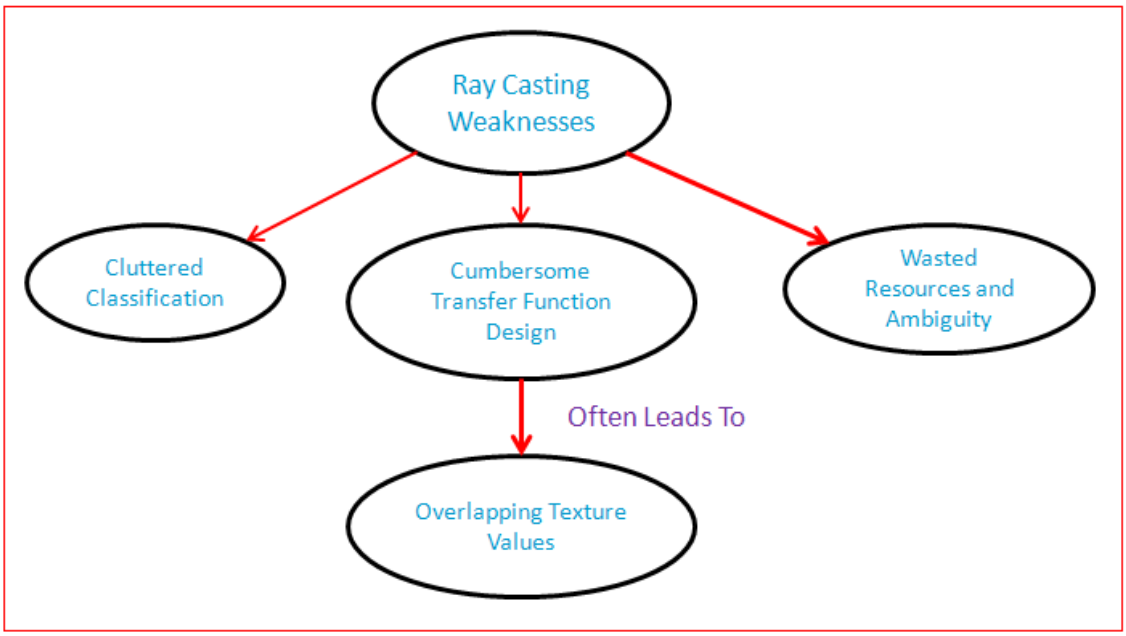

Figure 1.1: Shortcomings of ray casting.

In ray casting classification is basic. It is based directly on the raw data. A distinct range of values or in other words intensities is assigned to each particular object. However, it often occurs that an object does not entirely fall within its predefined range. Thus, occupying or overlapping other object(s)' ranges which leads to poor classification.

Since transfer function design is based on classification, a poor classification leads to imprecise transfer function design. On the left side of Figure 1.2 we have air, fat, soft tissue and bone as the four overlapping objects and on the right side of Figure 1.2 we have their relevant transfer function design which evidently mixes the relevant colors and opacities of distinct objects.

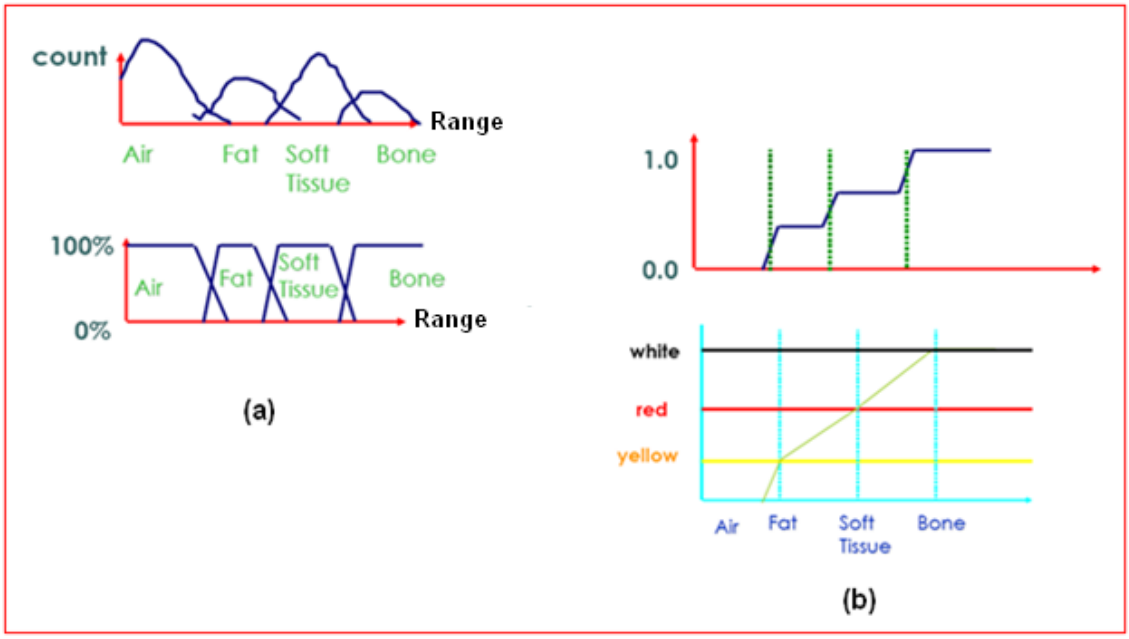

Figure 1.2: (a) Overlapping classification (b) Imprecise transfer function design.

Performing consecutive interpolations and compositions throughout the volume as a whole without a sense of direction not only consumes graphic resources but also reduces the level of details for the objects we are actually interested in. As may be noticed from Figure 1.3, in ray casting an ambiguous situation occurs when four elemental vector values at each cubical vertex are used to carry out interpolation. Interpolating all of these parameters at once leads to serious visual artifacts.

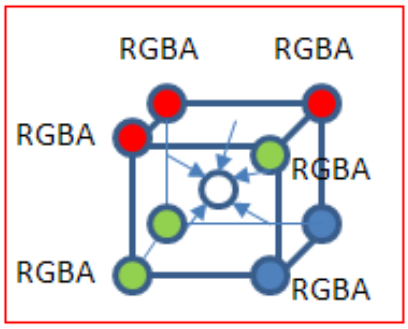

Figure 1.3: An ambiguous interpolation result originates from multiple vector parameters.

1.4 Objectives

The main objective of this work is to achieve enhanced visualization effect through amendment of ray casting based direct volume rendering. In general, the direct volume rendering process involves classification, transfer function formulation and rendering as its three respective and interdependent phases. Thus, in order to improve a particular phase one may need to modify its prerequisite phase(s). Our main objective could therefore be broken into three parts.

First, adaptation of an advanced segmentation algorithm instead of the basic classification module of ray casting.

Second, establishment of representative transfer functions that properly manifest classification complications.

Third, adjustment of the rendering pipeline for optimum implementation of transfer functions.

1.5 Contributions

Having mentioned our motivation, the actual problem we are trying to solve and our objectives we can now elaborate on our contributions. The first part of this work's contributions fulfills our first objective which is boosting classification. To boost classification we needed to come up with our own creative segmentation design. This work's segmentation design is based on K-means clustering algorithm. K-means algorithm in itself is an unreliable segmentation mechanism and has serious limitations. We have however managed to yield a reasonably reliable segmentation design by carefully resolving the K-means algorithm's limitations. The second part of this work's contributions fulfills our second and third objectives by targeting the limitations of ray casting. A tagged rendering design is proposed in order to rectify the limitations of ray casting. Figure 1.4 provides an elaborate list on the limitations of both K-means algorithm as well as ray casting. Please refer to Chapter three for comprehensive details on our solutions to the enlisted limitations.

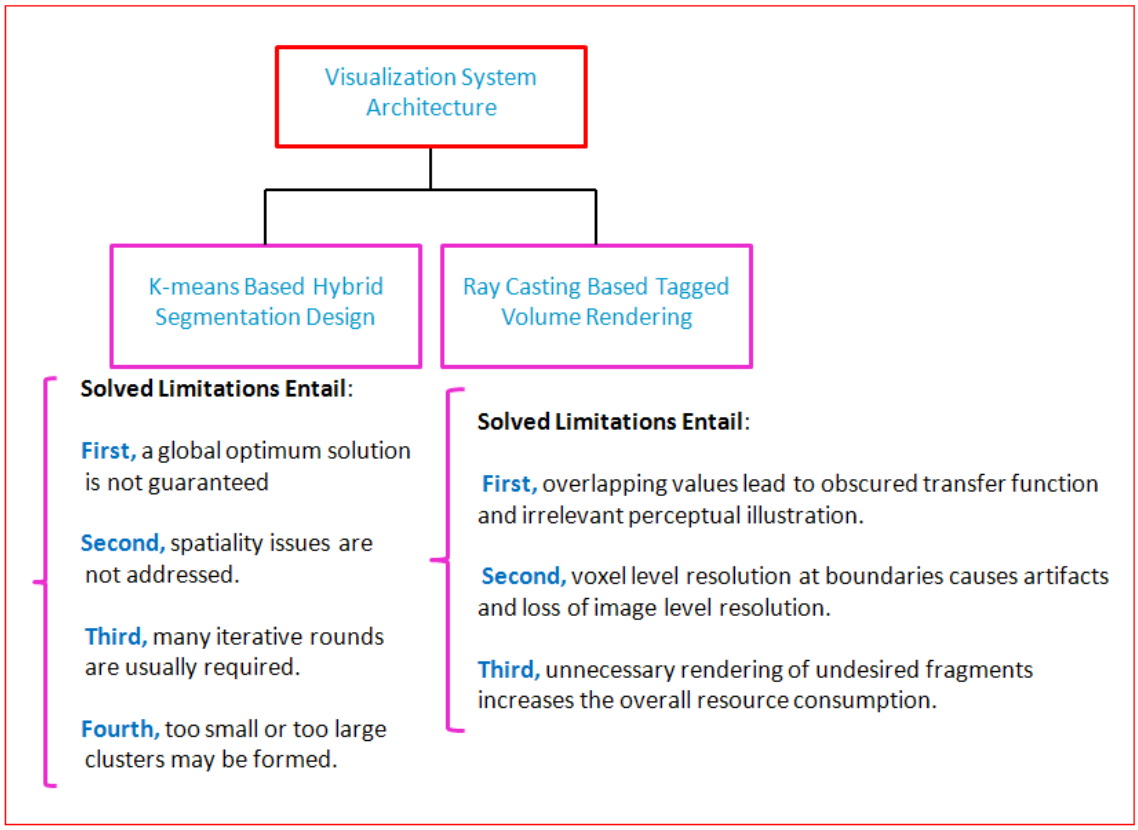

Figure 1.4: Enlistment of contributions.

1.6 Background

Visualization is a necessary means for making sense of the information (data). Depending on the nature of data, one could divide visualization technology into scientific and information categories. The term scientific visualization is used to categorize spatial-temporal data which must contain three spatial coordinates and the time dimension [Schroeder et al. 2002]. On the other hand information visualization terminology refers to data in higher-dimensional spaces or abstract spaces. Scientific visualization is used in science and engineering fields while information visualization deals with financial and marketing data. Due to the overlap among computer graphics, imaging and visualization, there is confusion about their actual differences. Computer graphics is the process of image creation via a computer. It includes 2D and 3D paint and draw techniques. Imaging or image processing

examines 2D images and learns from them. However, the concept of 3D imaging is publicized as well [Lukac et al. 2005]. It includes techniques to transform (ex. rotate, scale and shear), extract information from, analyze and enhance images [Schroeder et al. 2002]. Visualization explores, transforms and views data as images in order to enhance the data understanding process. Figure 1.5 attempts a conceptual demonstration of interlinks among imaging, computer graphics and visualization. As may be noticed, imaging and computer graphics are the two complementary components of a visualization system. Their inputs and outputs are interdependent. Depending on how one manages each of these components the visualization outcome could be weak and non-representative or strong and representative.

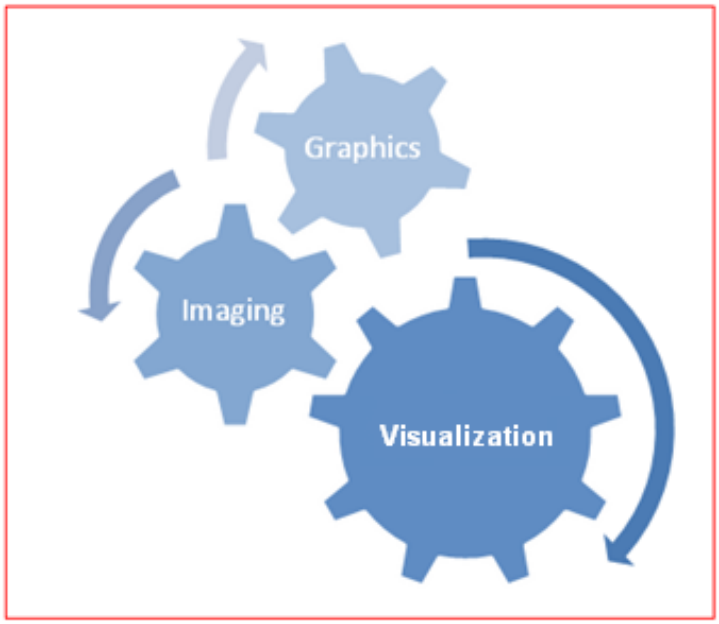

Figure 1.5: Correlation among computer graphics, imaging and visualization.

Further elaboration on a typical visualization process which encompasses both computer graphics and imaging is best depicted by Figure 1.6. As can be seen, data enrichment refers to repeated transformation process for information deduction,

derivation and enhancement. For instance, if we have sparse and accurate data we can interpolate to create a model but if we have sparse and flawed data we should approximate. Once a model is formed it should be converted (mapped) into a format understandable by graphic hardware. There are numerous computer graphic techniques which could generate feasible formats for different kinds of data model. For example, iso-surfacing technique uses triangles as a geometric means for scalar value representation. Visual properties (ex. Color) are then assigned to geometrical objects in order to fabricate an image. Volume rendering technique is yet another example which uses the cubical vertices directly in order to composite an image.

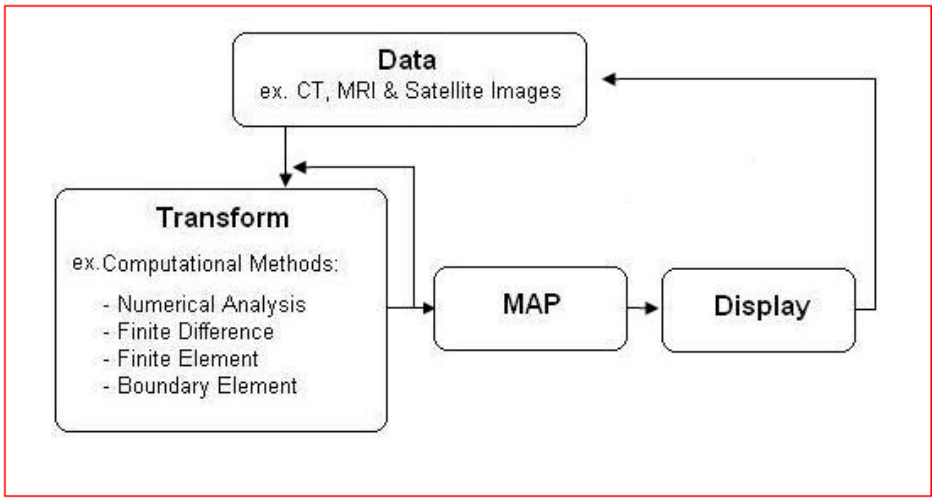

Figure 1.6: A typical visualization course. Data is recursively enriched by various transformation methods until a model is formed. It is then mapped to a graphic system for display [Schroeder et al. 2002].

Volume rendering encompasses a variety of computer graphics techniques for 2D projection of discretely sampled 3D datasets [Lichtenbelt et al. 1998]. A 3D dataset is usually represented by a group of regular 2D image slices forming a volumetric grid. By regularity we mean equal distances among the slices (ex. 1

millimeter distance between every 2D image slice scanned by CT or MRI scanners) and identical number of pixels on each slice. A regular volume is comprised of cubical elements (voxels) that in turn contain eight vertices. Value of a particular voxel may be derived by manipulation of its surrounding vertices. A volume could be rendered by extracting surfaces of equal values or directly as a block of data [Lichtenbelt et al. 1998]. Iso-surfacing techniques such as marching cubes form polygonal surfaces within voxels using threshold values. Direct volume rendering (DVR) techniques however deal with wholesome volume considerations through transfer function(s). Generally, Transfer function defines RGBA (red, green, blue, alpha) value for every voxel. Alpha is a measure of opacity and is associated with color values. Now, ray casting, splatting, shear-warp and texture mapping are some of the popular direct volume rendering techniques [Schroeder et al. 2002].

Ray casting computation emanates from output image rather than input volume data therefore, it is image based rather than object based. As may be noticed from Figure 1.7, there are three different layers to ray casting operation. A camera model which constitutes the center of ray projection (the eye point), the volume to be rendered and a virtual image plane that includes an array of pixels and floats between the camera and volume. Ray(s) is generated at camera layer for any desired number of pixels on the image plane. It passes through the volume, carrying out sampling at regular or adaptive intervals. Data is interpolated at each sampling point. Interpolation is essentially an averaging operation among vertices (each vertex holds a RGBA value generated earlier by transfer function) of the voxel being sampled. As ray(s) moves through and exits the volume, sampled points along the ray are composited (summed up) into a single RGBA value.

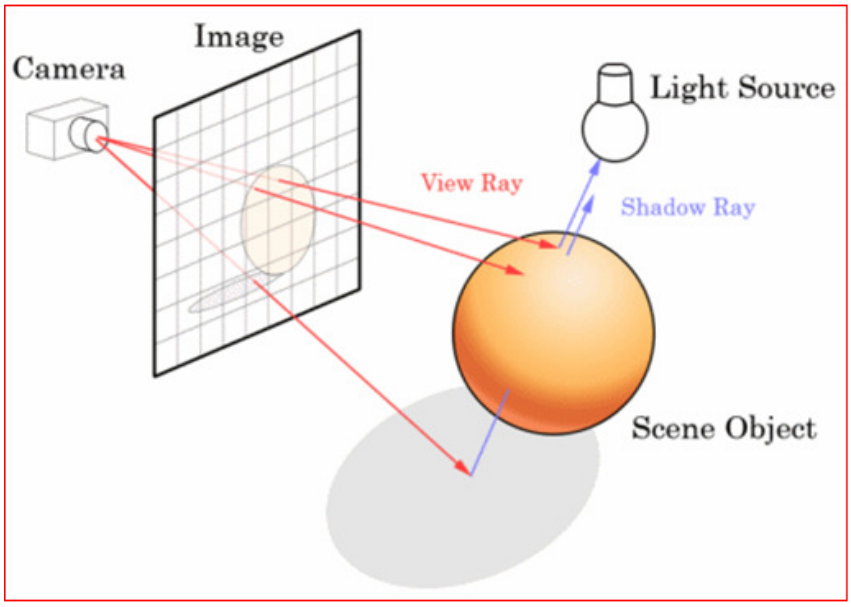

Figure 1.7: Ray casting based direct volume rendering [Camargo 2009].

In contrast to ray casting splatting sprinkles volume's voxels on the image plane like snow balls in a back to front order. The splats are rendered as disks whose properties (color and opacity) vary diametrically in Gaussian manner. Shear-warp was originally developed for speedy implementation of DVR at the cost of less accurate sampling and potentially poor rendering quality. It factorizes volume data at a predetermined voxels to pixels ratio, cutting out only the required (relevant) set of slices [Schulze and Lang 2002]. This is regarded as 3D shear. Then, an intermediate rendering process projects the sheared piece onto a 2D image plane and warps it back to the current display orientation. Texture mapping can produce acceptable rendering result given a graphic card that supports 3D Textures [Schroeder et al. 2002]. Volume data is mapped to appropriate color and opacity in order to form a texture map. Textured volume is sliced like loaf bread along the viewing angle. Each slice is a square and is parallel to other slices and the projection plane.

Segmentation is a key step towards derivation of semantics from digital images. The crucial need of numerous high level applications to certain level of

semantics is a fact that has drawn considerable attention to the field of image processing [Irani and Belaton 2009]. Segmentation refers to the process of partitioning a digital image into multiple regions (sets of pixels), where each region is different from the other regions in respect to some characteristics [Gonzalez and Woods 2006]. For instance, color, intensity, texture and spatiality. The goal of segmentation in general is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze [Shapiro and Stockman 2001]. Many general-purpose methods have been developed for image segmentation. However, since there is no general solution to the image segmentation problem, these methods often have to be combined with domain knowledge in order to effectively solve an image segmentation problem for a particular domain [Irani and Belaton 2009].

An interesting general-purpose segmentation algorithm is clustering. Clustering refers to the process of grouping pixels of an image such that pixels which are in the same group (cluster) are similar among them and are dissimilar to the pixels which belong to the other groups (clusters) [Irani and Belaton 2009]. Similarity is measured by distance and defined by an N dimensional feature space. Feature distance calculation differs from spatial distance calculation. Feature distance calculation is based on features such as color or intensity and texture while spatial distance calculation is based on X, Y (width, height) coordinates. Devising an appropriate distance calculation method is an important task since it greatly affects final clustering result [Irani and Belaton 2009]. Distance calculation method should be able to effectively compromise among multiple parameters. Furthermore, defining similarity parameters based merely on feature space may lead to clustering errors. For instance, two objects with similar color and texture properties but different

shapes will be grouped under the same cluster. Errors of this kind could be fatal in a content based retrieval system where shape is of substantial importance in production of appropriate search results. Clustering algorithms may be generally classified into four main categories which are hierarchical, overlapping, probabilistic and exclusive [Matteucci 2007]. Hierarchical clustering algorithms are based on union between two nearest clusters. They start by setting every pixel as a cluster and progress until final result (few desired clusters) reached [Fung 2001]. Overlapping clustering algorithms are based on fuzzy sets. Each pixel may belong to two or more clusters with different degrees of membership [Cleuziou 2008]. Final result is produced either in a ranked manner or by selecting an appropriate degree of membership for each pixel [Cleuziou 2008]. Probabilistic clustering algorithms are entirely based on probability. They develop probabilistic models (each model is represented by a particular cluster) and attempt to optimally match between the patterns that pixels may form and the models [Fung 2001]. Exclusive clustering algorithms exclusively group pixels, such that if a pixel belongs to a particular cluster then it could not belong to any other cluster [Fung 2001]. K-means is an instance of exclusive clustering algorithms and is used in this work's methodology. K-means algorithm starts clustering by determining K initial central points, either at random or using some heuristic data [Fahim et al. 2006]. It then groups each image pixel under the central point it is closest to. Next, it calculates new central points by averaging the pixels grouped under each central point [Fahim et al. 2006]. The two former algorithmic steps are repeated alternately until convergence (central points no longer change by averaging).

1.7 Outline

Chapter one provides the motivation of this research work, it defines the problems that are to be solved and proposes creative solutions. It also familiarizes the reader with basic but important concepts.

In Chapter two the reader is guided through a steady understanding and criticism of the research literature. Section 2.7 highlights the particularly pertinent literature.

Chapter three proposes a well defined solution. It strives to rectify the limitations of ray casting in the context of an integrated visualization frame work.

Chapter four and five elaborate on the solution proposed in Chapter 3 and carry out objective testing and comparative study. Five cases with extensive datasets are studied.

Chapter six is a wrap up. It concludes the past five chapters and insinuates the future work.

1.8 Chapter Summary

This chapter raises a substantial issue in the computer assisted visualization domain. It observes the current direct volume rendering techniques and specifically ray casting as insufficient means for fine visualization and proposes a union between image processing merit and computer graphics for an enhanced visualization effect. Ray casting is a direct volume rendering technique, suitable for volumetric visualization of sampled data. However, it suffers from elementary classification process and inferior transfer function design. An image processing based approach could therefore be used to advance the ray casting's classification process and improve its transfer function based object distinction capabilities.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

The basic idea of this work is to incorporate the advantageous aspects of image segmentation and volume rendering within an integrated frame work in order to improve volumetric visualization effect. Among volume rendering methods in computer graphics field we have chosen direct volume rendering (DVR) which in itself includes at least four concurrent techniques. Ray casting, shear-warp, splatting and texture mapping are the four DVR techniques that employ similar rendering principles in different manners. To be clear, by similar rendering principles we mean apparatus such as transfer function based texture assignment, sampling, composition, volumetric ray traversal and projection on a 2D plane. Among the mentioned DVR techniques, ray casting forms the basis of this work's proposed solution. As depicted by Table 2.1, this chapter is divided into 4 main parts.

Table 2.1: The literature review structure.

Section(s)	Review Domain	Description
2.2 - 2.5	concurrent DVR techniques including ray casting, shear-warp, splatting and texture mapping	characteristics, general improvements and applications
2.6	multidimensional transfer functions	rectification of transfer function design with no reference to segmentation
2.7	segmentation based direct volume rendering (ray casting in particular)	rectification of transfer function design and other rendering components with reference to segmentation
2.8	K-means based image segmentation	construction of a valid segmentation design by resolving the main limitations of k-means clustering

First, Sections 2.2 through 2.5 study the characteristics, general improvements and applications of DVR techniques. Depending on how DVR techniques use rendering principles their outcome could be fast or slow and high or low in quality and smoothness. General improvements enable non-fundamental but effective modifications to DVR techniques. DVR applications could embrace a wide range including for instance, visualization of human organs, real time virtual reality scenes and hyper dimensional information visualization.

Second, Section 2.6 studies the inherently imprecise transfer function design in direct volume rendering. It identifies the available solutions that merely emphasis on transfer function rectification with no regards for segmentation. The basic idea that is shared among literature presented in Section 2.6 is adding one or more dimensions to the otherwise standard single dimensional transfer function design. The goal of multidimensional treatment is to improve feature distinction capabilities and facilitate 3D object distinction process. However, there are limitations on the number of added dimensions because each dimension should preserve a compromising balance with other dimensions.

Third, in contrast to Section 2.6, Section 2.7 does not ignore classification as the original cause of imprecise transfer function design. It studies the solutions that consider the rendering process with reference to segmentation.

Fourth, Section 2.8 identifies the segmentation solutions that employ K-means clustering as their basis. It elaborates on how different research works have yielded a reasonably reliable segmentation design via resolving the serious limitations of K-means clustering.

2.2 Ray Casting

It is known that an increase in sampling rate reduces artifacts and improves the overall rendering quality through smooth reconstruction of scalar values [Lee et al. 2010]. An increase in sampling rate however sacrifices interactive volume rendering due to intensive computational demand. A virtual sampling method based on cardinal splines (Catmull-Rom type in particular) is proposed for fast and computationally efficient shrinking of sampling intervals [Lee et al. 2010].

The sampling points of each conventional ray segment are used as control points in order to construct curves with c^1 continuity (piecewise curve structure) [Lee et al. 2010]. The eager to retain early ray termination privilege has dismissed the use of a wholesome cubic spline with c^2 Continuity [Lee et al. 2010]. In early ray termination technique once sufficiently dense material was encountered, further samples will make no significant contribution to the final result and therefore can be neglected. A spline with c^2 continuity requires all samples and rules out the possibility of early termination employment. As can be notice from Figure 2.1, while the ray traverses through the voxels, every four intensity samples assume a spline such that the two middle points are the Section endpoints and the other two points are used for slope calculations [Lee et al. 2010]. The virtual sampling takes place at one-fourth of conventional sampling interval by polynomial evaluations. Simple arithmetic tasks for polynomial evaluations predict further sampling points without actual need to picking them [Lee et al. 2010].

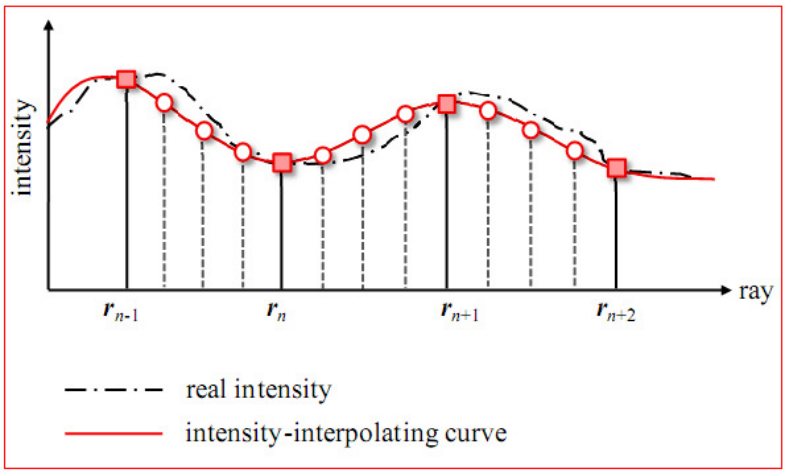

Figure 2.1: Virtual sampling. Red rectangles display the actual samples. Circles refer to virtual samples calculated in between the actual samples in order to obtain a curve with better uniformity and accuracy as compared to original curve represented by dark dashed line [Lee et al. 2010].

An axis aligned boundary box based solution is advocated to reduce sampling frequency and exempt empty voxels from ray casting process [Qing et al. 2010]. A target object is separated by fixing a bounding box encompassing a range between min and max coordinate points. Interpolation and sampling tasks are performed only if ray(s) crosses through the box [Qing et al. 2010].

Ray casting algorithm is modified for accurate results [Tian et al. 2008]. Volume data is initially divided into foreground and background via a threshold comparison. The background is leaped as irrelevant space and the foreground is sampled in reverse order such that classified optical attributes (ex. color and opacity) are only assigned after sampling. The sampling takes place based on LOD (level of detail) technique. For instance, a combination of nearest neighbor interpolation and tri-linear interpolation is used [Tian et al. 2008]. Degree of freedom for opacity measurement is broadened in order to enrich human visual perception.

A real-time volume search algorithm for fine object detection is introduced [Coatrieux et al. 2006]. As Ray traverses, each time a voxel that satisfies a predefined contrast threshold is encountered it is referenced as a seed point for region growing and all its connected neighboring voxels are analyzed. Those neighboring voxels that satisfy the threshold value are counted and stored in a stack. If the stack size reaches a size threshold the ray traversal is stopped and a region of interest is highlighted. A 3D linear interpolation is then performed on the neighborhood and refined detection is obtained [Coatrieux et al. 2006].

A trade-off between image order and object order algorithms is claimed in order to produce an accurate yet interactive solution [Mora et al. 2002]. Identical treatment of rays for volumetric cell projection on the image plane only enables fixed sampling through a translation vector while finer rendering may result from shifted sampling or in other words incoherency among the rays [Mora et al. 2002]. To achieve this, pre-computed rays are utilized. A square made of four neighboring pixels is partitioned and a list of coordinates corresponding to the cell projection is associated with each sector [Mora et al. 2002]. Figure 2.2 demonstrates the overall idea. Notice that shifted samples refer to shift in location rather than interval.

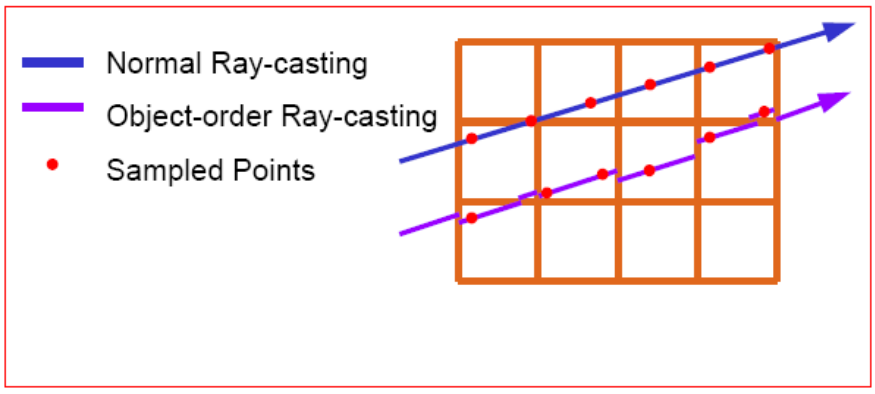

Figure 2.2: Shifted sampling in object order ray casting [Mora et al. 2002].

So far all the above approaches assumed regular grids for scalar data representation, allowing a gap for the review of disparate scalar fields. Due to complications caused by non-uniform data resolution (large regions of space having law field variation while small regions have high field variation) irregular grids are paid comparatively less attention. Curvilinear and non-curvilinear are the two common formats for irregular grids [Silva and Mitchell 1997]. The latter is an unstructured grid with no implicit connectivity information. The former is a twisted (bent) structured grid which retains the basic topology of regular grids [Silva and Mitchell 1997]. A lazy sweep ray casting algorithm is proposed for rendering irregular grids. It is based on sweep plane model and enables careful exploitation of spatial uniformity even when image resolution is significantly different from object space resolution. The sweep plane paradigm is a standard in computational geometry. As a 2D sweep plane is swept across 3D space a data structure stores the sweep status (upon encountering a discrete set of points data structure is updated). The main idea is to localize the problem and solve it within a lower dimensional (2D) space [Silva and Mitchell 1997].

2.3 Shear-Warp

According to [Chen et al. 2011] the shear-warp technique comprises of four main components which are view matrix (M_{view}), permutation (P), 3D shear (M_{shear}) and 2D warp $\left(M_{warp}\right)$. The shear-warp based rendering can be written as:

$$\boldsymbol{M_{view} = M_{shear} \times M_{warp} \times P} \qquad \text{Equation 2.1}$$

P transposes the coordinate system such that the principal view axis changes to Z, M_{shear} cuts a part of the volume and forms a sheared object space and M_{warp} converts the sheared object coordinates into image space.

M_{shear} could be written as:

$$\boldsymbol{M_{shear}} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \boldsymbol{S_x} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \boldsymbol{S_y} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \qquad \text{Equation 2.2}$$

The S_x and S_y are the translating factors for X and Y axis respectively. They can be derived from M_{view} matrix as below:

Let the upper left 3 × 3 sub-matrix of M_{view} be:

$$\boldsymbol{M_{view}} = \begin{bmatrix} \boldsymbol{m_{11}} & \boldsymbol{m_{12}} & \boldsymbol{m_{13}} \\ \boldsymbol{m_{21}} & \boldsymbol{m_{22}} & \boldsymbol{m_{23}} \\ \boldsymbol{m_{31}} & \boldsymbol{m_{32}} & \boldsymbol{m_{33}} \end{bmatrix} \qquad \text{Equation 2.3}$$

Then values of S_x and S_y can be derived as:

$$\boldsymbol{S_x} = \frac{\boldsymbol{m_{22}m_{13} - m_{12}m_{23}}}{\boldsymbol{m_{11}m_{22} - m_{21}m_{12}}} \qquad \text{Equation 2.4}$$

$$\boldsymbol{S_y} = \frac{\boldsymbol{m_{11}m_{23} - m_{12}m_{13}}}{\boldsymbol{m_{11}m_{22} - m_{21}m_{12}}} \qquad \text{Equation 2.5}$$

Given M_{view}, P and M_{shear} as known parameters we can easily obtain M_{warp} as below:

$$\boldsymbol{M_{warp} = M_{view} \times M_{shear}^{-1} \times P} \qquad \text{Equation 2.6}$$

[Chen et al. 2011] then proposes Lagrange and spline interpolation methods as an effective improvement that could reduce the staircasing problem of shear-warp based rendering. Considering only six sampling points represented by (X_k, Y_k) such that $X_k$and Y_k correspond to adjacent sampling points and their scalar values respectively, sampling points can be written as:

$$\left\{\begin{matrix} \boldsymbol{x | x = k, k+1, k+2, k+3, k+4, k+5} \\ \boldsymbol{k \epsilon N \leftarrow slice\ number} \end{matrix}\right\} \qquad \text{Equation 2.7}$$

The above defined sampling points are used within formulation of both Lagrange and spline interpolation methods. Lagrange interpolation polynomial is $P(x)$ of degree $\leq$ (n-1) that passes through n points $\big((X_1, Y_1 = f(x_1)), (X_2, Y_2 = f(x_2)), \ldots\ldots (X_n, Y_n = f(x_n))\big)$. It could be written as:

$$\boldsymbol{P(x) = \sum_{j=1}^{n} P_{j(x)}} \qquad \text{Equation 2.8}$$

Where:

$$P_j(x) = y_j \prod_{\substack{k=1 \\ k \neq j}}^{n} \frac{x - x_k}{x_j - x_k} \qquad \text{Equation 2.9}$$

Therefore:

$$P(x) = \frac{(x - x_2)(x - x_3) \ldots\ldots (x - x_n)}{(x_1 - x_2)(x_1 - x_3) \ldots\ldots (x_1 - x_n)} y_1 + \cdots + \frac{(x - x_1)(x - x_2) \ldots\ldots (x - x_{n-1})}{(x_n - x_1)(x_n - x_2) \ldots\ldots (x_n - x_{n-1})} y_n \qquad \text{Equation 2.10}$$

[Chen et al. 2011] formulates spline interpolation method as below:

For each segment ($[X_{j-1}, X_j]$) the cubic spline function is $S(x)$ and should be based on the following format:

$$S(x) = S_j(x) = a_j x^3 + b_j x^2 + c_j x + d_j \qquad \text{Equation 2.11}$$

In order to derive a formulation, $S(x)$ must meet three conditions, which are:

First, $S(x_j) = f(x_j)$ $(j = 0,1, \ldots, n)$.

Second, S(x) is not more than a third degree polynomial in every segment ($[X_{j-1}, X_j](j = 0,1, \ldots, n)$).

Third, $S^{"}(x)$ exists while it is continuous on the interval (x_0, x_n).

Now, the cubic spline function formula could be written as:

$$S(x) = \frac{(x - x_{k+1})^3}{6h_k} m_k + \frac{(x - x_k)^3}{6h_k} m_{k+1} - \left(y_k - \frac{h_k^2}{6} m_k \right) + \left(y_{k+1} - \frac{h_k^2}{6} m_{k+1} \right) \frac{x - x_k}{h_k} \quad \text{Equation 2.12}$$

High quality rendering is traded-off with speed in shear-warp algorithm [Li et al. 2010]. Specifically, aliasing takes effect as a result of alternative sampling distances (a range between 1 and $\sqrt{3}$) for different viewing angles. Larger sampling distances at particular angels (45◦) may cause stair-casting impact [Li et al. 2010]. To reduce aliasing, interpolation between two adjacent slices is proposed. An interpolation method based on multi-quadric radial basis function is employed. The interpolation is performed in three main steps.

First, a curve is constructed from the four original and neighboring sampling points along the main viewing angle.

Second, the coefficients of the curve are obtained [Li et al. 2010].

Third, using the curve new re-sampling points are calculated based on which new slices are constructed and fitted into original slices.

Shear-warp based rendering is modified to accommodate wavelet data encoding technique and trivariate spline model in order to introduce a new approach for fast and interactive volume visualization [Schlosser et al. 2005]. Standard wavelet expansions are used to represent the dataset grid. Such form of representation allows employment of octree organization to hierarchically guide the local reconstruction of