CORA Cork Open Research Archive
Cartlann Taighde Oscailte Chorcaí

Title	Uncertainty visualization in 3D scalar data
Authors	Ma, Ji
Publication date	2014
Original Citation	Ma, J. 2014. Uncertainty visualization in 3D scalar data. PhD Thesis, University College Cork.
Type of publication	Doctoral thesis
Rights	© 2014, Ji Ma - http://creativecommons.org/licenses/by-nc-nd/3.0/
Download date	2026-11-06 04:42:22
Item downloaded from	https://hdl.handle.net/10468/1535



UCC
University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

Uncertainty Visualization In 3D Scalar Data

Ji Ma, M. Sc.



A thesis submitted to the National University of Ireland, Cork

in fulfilment of the requirements for the degree of

Doctor of Philosophy in the Faculty of Science

April 2014

Research Supervisors: Prof. Gregory Provan

Mr. David Murphy

Prof. Cian O'Mathuna

Head of Department: Prof. Barry O'Sullivan

Department of Computer Science and Tyndall National Institute,

National University of Ireland, Cork.

Acronyms

CT	Computed Tomography
MRI	Magnetic Resonance Imaging
PET	Positron Emission Tomography
HVS	Human Visual System
MR	Multiresolution
IVR	Indirect Volume Rendering
DVR	Direct Volume Rendering
MC	Marching Cubes
PC	Parallel Coordinates
LVIS	Linked Views and Interactive Specification
ELVIS	Extended Linked Views and Interactive Specification
TF/TFs	Transfer Function/Transfer Functions
RGBα	Red, Green, Blue and alpha
ROI	Region of Interest
PDF	Probability Density Function
HSL	Hue, Saturation and Lightness
UCC	University College Cork
MANOVA	Multivariate Analysis of Variance
ANOVA	Analysis of Variance
RM	Repeated Measurements
SPSS	Statistical Product and Service Solutions

OpenGGCM	Open Geospace General Circulation Model
GPU	Graphics Processing Unit
0D	Zero-dimensional
1D	One-dimensional
2D	Two-dimensional
3D	Three-dimensional
4D	Four-dimensional
P	Point
S	Scalar
V	Vector
T	Tensor

Abstract

One problem in most three-dimensional (3D) scalar data visualization techniques is that they often overlook to depict uncertainty that comes with the 3D scalar data and thus fail to faithfully present the 3D scalar data and have risks which may mislead users' interpretations, conclusions or even decisions.

Therefore this thesis focuses on the study of uncertainty visualization in 3D scalar data and we seek to create better uncertainty visualization techniques, as well as to find out the advantages/disadvantages of those state-of-the-art uncertainty visualization techniques. To do this, we address three specific hypotheses: (1) the proposed Texture uncertainty visualization technique enables users to better identify scalar/error data, and provides reduced visual overload and more appropriate brightness than four state-of-the-art uncertainty visualization techniques, as demonstrated using a perceptual effectiveness user study. (2) The proposed Linked Views and Interactive Specification (LVIS) uncertainty visualization technique enables users to better search max/min scalar and error data than four state-of-the-art uncertainty visualization techniques, as demonstrated using a perceptual effectiveness user study. (3) The proposed Probabilistic Query uncertainty visualization technique, in comparison to traditional Direct Volume Rendering (DVR) methods, enables radiologists/physicians to better identify possible alternative renderings relevant to a diagnosis and the classification probabilities associated to the materials appeared on these renderings; this leads to improved decision support for diagnosis, as demonstrated in the domain of medical imaging.

For each hypothesis, we test it by following/implementing a unified framework that consists of three main steps: the first main step is uncertainty data modeling, which clearly defines and generates certainty types of uncertainty associated to given 3D scalar data. The second main step is uncertainty visualization, which transforms the 3D scalar data and their associated uncertainty generated from the first main step into two-dimensional (2D) images for insight, interpretation or communication. The third main step is evaluation, which transforms the 2D images generated from the

second main step into quantitative scores according to specific user tasks, and statistically analyzes the scores. As a result, the quality of each uncertainty visualization technique is determined.

Acknowledgements

Many thanks to:

* My supervisors and examiners: Prof. Gregory Provan, Mr. David Murphy, Prof. Cian O'Mathuna, Mr. Michael Hayes, Prof. Paul McKevitt and Dr. Sabin Tabirca.

* The Irish Research Council for Science, Engineering and Technology (IRCSET) for the financial support of my PhD.

* My colleague Gareth Young in the Department of Computer Science for his generous help with proof reading.

* My friends Chris Murphy, Zixiang Cong, Xiao Chen, Xingcan Wang, Wenqin Xu, Li Liu, Mei Jin, Zhaolin Liu, Lin Jing, Joan Maguire, Fuhua Huang, Yaqi Wan and Xiaowen Zhang for their companying in Ireland.

* My colleagues in the Department of Computer Science for providing valuable information for PhD study, as well as making work more enjoyable: Tracey Mehigan, Alie Mady, Marion Behrens, Weiping Song, Menouer Boubekeur, Ann O'Brien and Xiuchao Wu.

* My colleagues in the Tyndall National Institute: Zhiheng Quan, Nicolas Holubowitch, Andreas Wieczorek, Jiadi Lu, Ciaran Cleary, Weiwei Yu, Aidan Daly, Sanjay Patil, Vanessa Smet, Hongjia Mo and Wei Han.

* Prof. Claes Lundstrom from the Center for Medical Image Science and Visualization (CMIV), Linkoping University and Mr. Andrew Foulks from University of New Hampshire, for their generously offering some test data sets used in this thesis.

* Those who volunteered to participate the user studies for my research.

* Mom, Dad, Grandmother and my girlfriend for unwavering support.

This thesis is dedicated to my wonderful parents

Hongyan Mi & Zhiqiang Ma,

and my dear grandmother

Fuhua Jia.

Thank you all for everything.

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institution of learning.

Signed:

Ji Ma

Contents

Acronyms i

Abstract iii

Acknowledgements v

Declaration vii

List of Tables xiii

List of Figures xiv

Chapter 1 Introduction 1

1.1 Uncertainty Visualization in 3D Scalar Data 1

1.1.1 Motivation 2

1.1.1.1 Pervasive Uncertainty in 3D Scalar Data 2

1.1.1.2 The Need for Uncertainty Visualization in 3D Scalar Data 5

1.1.2 Challenges 8

1.1.3 Research Hypotheses 11

1.1.4 Contributions 11

1.2 Thesis Outline 12

Chapter 2 Background and Related Work 15

2.1 Taxonomy and Notation of Uncertainty Visualization Techniques 15

2.2 Review of Uncertainty Visualization Techniques in Scalar Data 16

2.2.1 U_0^S 17

2.2.2 U_1^S 18

2.2.3 U_2^S 20

2.2.4 U_3^S 23

2.2.5 Multivariate Uncertainty Data 28

2.2.5.1 U_0^{kS} 28

2.2.5.2 $U_{>0}^{kS}$ 29

2.3 Evaluation of Uncertainty Visualization Techniques in Scalar Data 30

2.4 Summary 31
Chapter 3 Thesis Technical Foundations and Framework 32

3.1 Foundations 32
3.1.1 3D Scalar Data 32
3.1.2 Volume Visualization 34
3.1.2.1 Isosurface Rendering – IVR 34
3.1.2.2 DVR 38
3.2 Framework 42
3.2.1 Uncertainty Data Modeling 44
3.2.2 Uncertainty Visualization 46
3.2.3 Evaluation 50
3.3 Summary 51
Chapter 4 Uncertainty Visualization in Isosurface Rendering Using Textures 53

4.1 Introduction 53
4.2 MR Modeling and Quantification for Univariate 3D Scalar Data 54
4.2.1 Motivation 54
4.2.2 MR Modeling 55
4.2.2.1 Haar Wavelet Transform 55
4.2.2.2 Decimation 57
4.2.3 Quantification 58
4.3 Texture Uncertainty Visualization Technique 58
4.3.1 Extended Marching Cubes Algorithm 59
4.3.2 Two Types of Textures 60
4.3.2.1 Blurred Textures 60
4.3.2.2 Glyph Textures with Different Number of Edges 61
4.3.3 Case Study 61
4.3.3.1 Data Set and Pre-processing 61
4.3.3.2 Results and Discussion 63
4.4 Evaluation of Perceptual Effectiveness 66
4.4.1 Study Design 67
4.4.1.1 Uncertainty Data Modeling 67
4.4.1.2 Six Uncertainty Visualization Techniques Chosen for Evaluation 68
4.4.1.3 Participant Pool 72
4.4.1.4 User Study Tasks 72

4.4.1.5 Interface Design 73
4.4.1.6 Participant Training 74
4.4.1.7 The Trial Run 74
4.4.1.8 The Main Study 75
4.4.2 Initial Findings 75
4.4.2.1 Data Identification 75
4.4.2.2 Error Identification 76
4.4.2.3 Visual Overload 77
4.4.2.4 Brightness 78
4.4.3 Method of Analysis 79
4.4.4 Results and Discussion 79
4.4.4.1 Data Identification 79
4.4.4.2 Error Identification 83
4.4.4.3 Visual Overload 84
4.4.4.4 Brightness 85
4.5 Evaluation of Execution Time 86
4.6 Summary 87
Chapter 5 Uncertainty Visualization in DVR Using LVIS 89
5.1 Introduction 89
5.2 MR Modeling and Quantification for Multivariate 3D Scalar Data 90
5.2.1 MR Modeling 91
5.2.1.1 Haar Wavelet Transform 91
5.2.1.2 Decimation 92
5.2.2 Quantification 93
5.3 LVIS Uncertainty Visualization Technique for Univariate 3D Scalar Data 94
5.3.1 Workflow 95
5.3.2 Visualization Components 96
5.3.2.1 Scalar/Error Data Object-space View 96
5.3.2.2 Scatter Plot Feature-space View 97
5.3.2.3 Linking between Views and Depth Sample Extraction 97
5.3.3 Case Study 98
5.3.3.1 Original Data Set and Pre-processing 98
5.3.3.2 Results and Discussion 99
5.4 Evaluation of Perceptual Effectiveness 105
5.4.1 Study Design 105

5.4.1.1 Uncertainty Data Modeling 105
5.4.1.2 Five Uncertainty Visualization Techniques Chosen for Evaluation 106
5.4.1.3 Participant Pool 110
5.4.1.4 User Study Tasks 111
5.4.1.5 Interface Design 113
5.4.1.6 Participant Training 114
5.4.1.7 The Main Study 115
5.4.2 Method of Analysis 115
5.4.3 Results and Discussion 118
5.5 Evaluation of Execution Time 122
5.6 ELVIS Uncertainty Visualization Technique for Multivariate 3D Scalar Data 122
5.6.1 Overview of Functionalities 123
5.6.2 Functionality 1: Visualization of Statistical Information and Correlations 124
5.6.2.1 Scalar/Error Data Views 124
5.6.2.2 Scalar/Error Histogram Views 124
5.6.2.3 Parallel Coordinates View 125
5.6.2.4 Linking between Views and Depth Sample Extraction 125
5.6.3 Functionality 2: Visualization of Data Features 126
5.6.3.1 Scatter Plot Views 126
5.6.3.2 Linking between Views 127
5.6.4 Functionality 3: Feature Localization 128
5.6.4.1 Logical Scatter Plot Views 128
5.6.4.2 Feature Localization View 130
5.6.4.3 Linking between Views 130
5.6.5 Case Study 131
5.6.5.1 Original Data Set and Pre-processing 131
5.6.5.2 Results and Discussion 132
5.7 Summary 141
Chapter 6 Uncertainty Visualization of Material Intensity Classification in DVR Using Probabilistic Query Method 143
6.1 Introduction 143
6.2 Explicitly Probabilistic TFs 145
6.3 Probabilistic Query 148
6.3.1 Overview of Workflow 148

6.3.2 Query Statements 149
6.3.2.1 Probability Inequality 150
6.3.2.2 Approximate Matching 152
6.3.2.3 Weight of Query 155
6.3.2.4 Priority of Query 157
6.3.3 Query Color Specification 158
6.3.4 Query Volume Rendering 161
6.3.5 Single Material Query and Multiple Material Query 162
6.3.6 Case Studies 163
6.3.6.1 Case Study 1: Thyroid Tumor Assessment 164
6.3.6.2 Case Study 2: Stenosis Assessment 166
6.3.6.3 Case Study 3: Breast Tumor Assessment 168
6.4 Summary 176
Chapter 7 Conclusions and Future Work 178
7.1 Conclusions 178
7.2 Future Work 182
References 185

List of Tables

Table 2.1: Classification of uncertainty visualization techniques. The grey part indicates the scope of this thesis. 16
Table 4.1: Multivariate tests[d] table. 80
Table 4.2: Tests of within-subjects effects on data identification. 81
Table 4.3: Statistically significant difference between the 6 techniques on the aspect of data identification. 82
Table 4.4: Tests of within-subject effects on error identification. 82
Table 4.5: Statistically significant difference between the 6 techniques on the aspect of error identification. 83
Table 4.6: Tests of within-subjects effects on visual overload. 84
Table 4.7: Statistically significant difference between the 6 techniques on the aspect of visual overload. 85
Table 4.8: Tests of within-subjects effects on brightness. 86
Table 4.9: Execution time of the 6 evaluated uncertainty visualization techniques. 86
Table 5.1: Tests of between-subjects effects on accuracies. 116
Table 5.2: Results for one-way ANOVA on accuracies. 117
Table 5.3: Tests of between-subjects effects on time. 118
Table 5.4: Univariate tests on time from the first 4 one-way ANOVAs. 118
Table 5.5: Univariate tests on time from the second 5 one-way ANOVAs. 119
Table 5.6: Statistical significance of accuracies between the 5 uncertainty visualization techniques 120
Table 5.7: Statistical significance of the completion time between the 5 uncertainty visualization techniques for the task of identifying the maximum error. 120
Table 5.8: Statistical significance of the completion time between the 5 uncertainty visualization techniques for the task of identifying the minimum error. 121
Table 5.9: Statistical significance of the completion time between the 4 types of searching tasks with respect to the LVIS technique. 121
Table 5.10: Execution time of the 5 evaluated uncertainty visualization techniques. 122

List of Figures

Figure 1.1: The visualization pipeline shows the introduction of data uncertainty 3
Figure 1.2: Two similar 3D scalar data sets 5
Figure 1.3: A brain tumor data set from CT scan 6
Figure 1.4: (a) A photograph of a Mayan temple 8
Figure 1.5: An example of uncertainty adding an extra dimension 10
Figure 2.1: Tukey box plot 17
Figure 2.2: Uncertainty visualization techniques for 1D case 19
Figure 2.3: Uncertainty visualization techniques for 2D case 20
Figure 2.4: Increasing irregularity in a texture pattern 20
Figure 2.5: Two frames from an animation to reveal the uncertainty 22
Figure 2.6: A 2D annotation technique 22
Figure 2.7: The contour technique 23
Figure 2.8: The multiple linked Views technique 23
Figure 2.9: Isosurface rendering of a cadaver head data set 23
Figure 2.10: Uncertainty in the real tumor data set 24
Figure 2.11: Isosurface rendering of lobster data 24
Figure 2.12: Uncertainty visualization of ocean data with uncertainty 25
Figure 2.13: Uncertainty visualization of magnetohydrodynamics 26
Figure 2.14: Uncertainty visualization of data classification uncertainty 27
Figure 2.15: Uncertainty visualization consists of a PC control interface 29
Figure 3.1: Typical grid types in two dimensions 33
Figure 3.2: An example of 3D scalar data set with regular grids 34
Figure 3.3: Classification of volume visualization 34
Figure 3.4: Visualization pipeline of (a) isosurface rendering vs, (b) DVR 35
Figure 3.5: A labelled cell 36
Figure 3.6: The cubeIndex 36
Figure 3.7: By symmetry there are 15 possible ways 37
Figure 3.8: Edge 2, 3 and 11 intersect this cell 37
Figure 3.9: A typical TF 40
Figure 3.10: Rendering the 3D tooth scalar data 40
Figure 3.11: A diagram for volume ray-casting 40
Figure 3.12: Compositing from front to back 42
Figure 3.13: The framework of this thesis 42
Figure 3.14: Workflow of Texture uncertainty visualization technique 42
Figure 3.15: Workflow of LVIS uncertainty visualization technique 43
Figure 3.16: Workflow of ELVIS uncertainty visualization technique 43
Figure 3.17: Workflow of Probabilistic Query uncertainty visualization technique 44
Figure 4.1: An example of a MR hierarchy 57
Figure 4.2: A cell of 3D scalar data with 8 voxels distributed at the 8 corners 59
Figure 4.3: Blurred textures used to depict the degree of errors 61
Figure 4.4: Glyph textures with different number of edges 61
Figure 4.5: The MR hierarchy of a CT scan's lobster 62
Figure 4.6: The results by using the blurred textures technique 64

Figure 4.7: The results of using the glyph textures with different number of edges 66
Figure 4.8: Errors are mapped to the hue 68
Figure 4.9: Errors are mapped to the blurred textures 69
Figure 4.10: Errors are mapped to the glyph textures 70
Figure 4.11: Errors are mapped to the transparency 70
Figure 4.12: Errors are mapped to the transparency with enhanced grid 71
Figure 4.13: Errors are mapped to the texture opacity 72
Figure 4.14: The user interface 73
Figure 4.15: Average scores of the 6 uncertainty visualization techniques on the data identification aspect 76
Figure 4.16: Average scores of the 6 uncertainty visualization techniques on the error identification aspect 77
Figure 4.17: Average scores of the 6 uncertainty visualization techniques on the visual overload aspect 78
Figure 4.18: Average score of the 6 uncertainty visualization techniques on the brightness aspect 78
Figure 5.1: An example of the qualitative uncertainty visualization 89
Figure 5.2: An example of a MR 92
Figure 5.3: Overview of the LVIS workflow 96
Figure 5.4: An example of a MR hierarchy generated from the Haar Wavelet Transform ... 99
Figure 5.5: The LVIS uncertainty visualization 101
Figure 5.6: The LVIS uncertainty visualization is applied 102
Figure 5.7: The LVIS 103
Figure 5.8: The LVIS uncertainty visualization is applied to (a) 104
Figure 5.9: The LVIS uncertainty visualization is 104
Figure 5.10: The transparency technique 107
Figure 5.11: The discontinuities technique 107
Figure 5.12: The overlays technique 108
Figure 5.13: The hue technique 109
Figure 5.14: The LVIS technique 110
Figure 5.15: The user interface 114
Figure 5.16: Typical correlations that could be shown in a PC 125
Figure 5.17: An example about how to acquire the corresponding features of a ROI for multivariate data 128
Figure 5.18: An example of a compound ROI specification 128
Figure 5.19: A MR hierarchy for the density variable (left column) 134
Figure 5.20: The results after a ROI of the spatial domain is specified 138
Figure 5.21: The results after the multivariate data's certain features are specified 139
Figure 5.22: The results after the multivariate data's certain features 141
Figure 6.1: (a) A rendering based on one classification scheme 144
Figure 6.2: An explicitly probabilistic TF 146
Figure 6.3: A specific example about how a concrete explicitly probabilistic TF is used to compute the material probability 148
Figure 6.4: Overview of the Probabilistic Query workflow 149
Figure 6.5: The effects of three singular queries on the MR kidney data set with different target probability 151
Figure 6.6: The fuzzy matching scores with different dropoff parameters 154

Figure 6.7: The effects of three singular queries on the MR kidney data set with different dropoff parameters 155
Figure 6.8: The effects of three singular queries on the MR kidney data set with different weight parameters 156
Figure 6.9: The effects of three compound queries on the MR kidney data set with different combination of priority parameters 159
Figure 6.10: An example about how to specify the query color 161
Figure 6.11: Two examples of extracting multiple features from the same material 163
Figure 6.12: The Probabilistic Query uncertainty visualization technique 165
Figure 6.13: Two renderings of the CT thyroid tumor data set 166
Figure 6.14: A rendering of a MR renal angiography from the traditional DVR 166
Figure 6.15: The Probabilistic Query uncertainty visualization technique applied to the MR renal angiography 167
Figure 6.16: The Probabilistic Query uncertainty visualization applied to examine the breast tumor extent in relation to the breast 170
Figure 6.17: The Probabilistic Query uncertainty visualization applied to examine the correlations between the tumor and its surrounding vessels 171
Figure 6.18: The Probabilistic Query uncertainty visualization applied to reveal results 175
Figure 6.19: An example to show the Probabilistic Query uncertainty visualization's capability for complex feature localization 176

Chapter 1 Introduction

1.1 Uncertainty Visualization in 3D Scalar Data

3D scalar data, also known as 3D scalar fields, or volumetric scalar data, are a common type of data used in many application areas, for example, medical imaging, biology, geophysics, industry, molecular systems, meteorology, and computational fluid dynamics (Celebi, 2013). They are often acquired from instrumental measurements, for example, computed tomography (CT), magnetic resonance imaging (MRI), positron emission tomography (PET), ultrasound and confocal microscopy or scientific simulations, for example, stormy predication and climate change simulation. The visualization of such data types are known as volume visualization (Brodlie, 2006), which is now well accepted as a powerful means to allow scientists to explore large data sets, and to present their results to a wider audience (Brodlie, Osorio and Lopes, 2012).

However, one main problem existing in most volume visualization techniques is that they are developed under an assumption that the data sets to be visualized are perfect, without incorporating any uncertainty. This is rarely the case in the real world as uncertainty is an inherent part of all data (Sanyal et al., 2009) and the sources of uncertainty are throughout the scientific process (Potter, 2010a)(Potter, 2011). For example, starting with the data acquisition stage, nearly all data sets are generated with some extent of uncertainty. For data measurements, no matter whether they are taken by scientists or instruments, there will always be statistical variation in these measurements. For scientific simulations, the model and its parameters are usually decided by a domain specialist, and are inherently a simplification of the system being modeled. In the data transformation stage, data may be rescaled, resampled, quantized, etc. and thus have the potential to introduce uncertainty. Even in the visualization stage, uncertainty may also be introduced into the data due to the operations such as interpolation, resampling, classification etc. that are built in to the visualization algorithms.

Overlooking the depiction of uncertainty in visualization will result in an unfaithful representation of data. More importantly, it leaves the visualization running in a risk that misleads viewers' interpretation of data, or drawing incorrect conclusions or decisions from data. Because of this, uncertainty visualization has been recognized as one of top research problems in visualization literature (Johnson, 2004)(Chen, 2005)(Johnson et al., 2006)(Laramee and Kosara, 2007), and great emphasis has been placed on its research (Johnson and Sanderson, 2003):

"We urge the scientific visualization research community to take the next step and make visually representing errors and uncertainties the norm rather than exception."

Thus, this thesis is about uncertainty visualization in 3D scalar data and we present our approaches to tackle this issue.

1.1.1 Motivation

There are two main reasons that motivate the research of this thesis. This section will discuss them in details.

1.1.1.1 Pervasive Uncertainty in 3D Scalar Data

Uncertainty is pervasive in all data and this is also the case in 3D scalar data. We can understand this by looking at the sources of uncertainty in a typical visualization pipeline, as proposed by (Pang, Wittenbrink and Lodha, 1997)(Pang, 2008) and illustrated in Figure 1.1. This figure shows that there are three stages involved in the visualization pipeline. First, there is a data acquisition stage where data may be collected from measurements, field observations, or numerical models. Then these data undergo a data transformation stage where a series of transformations are involved *i.e.* unit conversion, refine, feature extraction, etc. Finally, the results of data from the data transformation stage are fed into a visualization stage where they are mapped to visual parameters. This figure illustrates that uncertainty can be introduced at any stage in the pipeline, even in the stage of visualization itself.

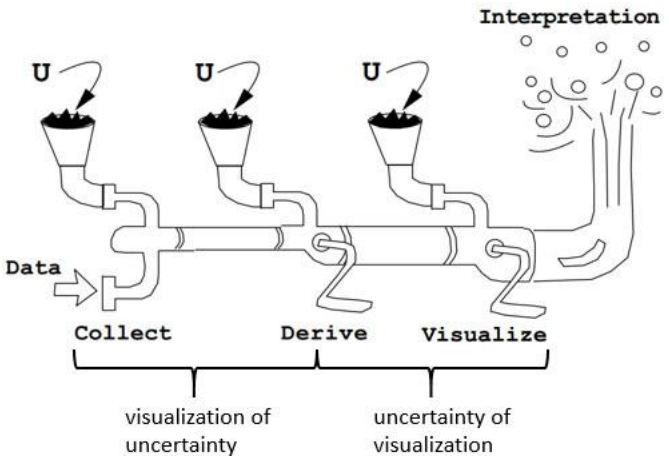


Figure 1.1: The visualization pipeline shows the introduction of data uncertainty (U) (Pang, Wittenbrink and Lodha, 1997)(Pang, 2008).

Uncertainty in data acquisition: data may be collected from observations and measurements or scientific simulations. In either case, uncertainty may be introduced into data due to a variety of factors such as miscalibration in instruments, or over simplification of mathematical models. For data measurement, there is an experimental variability associated with every measurement, whether a machine or a scientist takes it. The more times the measurement is taken, the more confident the measurement is. But there will always be a statistical variation in these measurements. For scientific simulations, the model and its parameters are usually decided by a domain expert, inherently a simplification of the system being modeled. In addition to model simplification and sensitivity of these models to input parameters, numerical calculations performed on these models may also introduce uncertainty due to the choices of integration algorithms and the limited precision of computing machinery (Pang, Wittenbrink and Lodha, 1997). Likewise, uncertainty can also be introduced into data by human observations as there is a difference in perception among individuals.

Uncertainty in data transformation: quite often, raw data will undergo further transformations before a final image is generated. These data transformation operations can be as simple as unit conversation, or may involve some more complex processes such as mapping classified intensity data to optical properties, that is normally done the Transfer Functions (TFs). These data transformations may happen as early as the data acquisition stage, or later as part of the visualization stage. As the

data transformations change the data from their original forms, there is potential to introduce uncertainty.

Uncertainty in data visualization: uncertainty can also be introduced during the visualization stage itself. For example, filtering and mapping are two common steps in many visualization techniques, and both of them have the potential to introduce uncertainty. For the filtering, it usually involves the operation of interpolation, which intends to build a continuous empirical model from the acquired discrete data. Uncertainty will of course be introduced into data during interpolation because we are only making plausible inferences of the data behaviours between sampled voxels. Take the volume ray-casting rendering as a more specific example. When we reconstruct the value of each interior sampling point in a cell along a casted ray, different interpolation algorithms can be employed *i.e.* the nearest neighbor interpolation, trilinear interpolation, cubic convolution and B-spline interpolation (Meibner et al., 2000), with each one making different assumption of data behaviour. Consequently uncertainty to different extents could be introduced. As for the mapping step, it involves the process to create a geometric model, and uncertainty can be introduced into the data when the geometric model is approximated *i.e.* the curved isosurface is approximated by triangles in the Marching Cubes (MC) algorithm (Lorensen and Cline, 1987).

Figure 1.1 shows that in particular, two types of uncertainty can be distinguished, depending on the stage where they are generated. “Visualization of uncertainty” focuses on the data acquisition stage or data transformation that occurs as early as the data acquisition stage, while “uncertainty of visualization” focuses on the visualization stage or data transformation that occurs within the visualization stage.

Normally, we view the visualization pipeline as a left-to-right output process, and uncertainty could be accumulated and propagated when data pass through the pipeline. In interactive work, we also traverse the pipeline from right-to-left, and uncertainty needs to be borne in minds as we pass back along the pipeline (Brodlie, Osorio and Lopes, 2012).

In any case, the uncertainty is pervasive in 3D scalar data and it should not be disregarded in the volume visualization.

1.1.1.2 The Need for Uncertainty Visualization in 3D Scalar Data

A simple but very common need for uncertainty visualization in 3D scalar data comes from the task to compare possible differences or errors between two similar 3D data sets. Figure 1.2 illustrates such an example and two similar 3D data sets are volume rendered on the left and right. While it is possible to tell the partial differences between the two 3D data sets by our Human Visual System (HVS) (Alexandre and Tavares, 2010)(Hansen and Johnson, 2004) *i.e.* the lower edge on the right figure is slightly coarser than the one on the left figure, it is impossible to make an accurate and comprehensive distinction of their differences. Thus, a visualization technique that could not only represent the characteristics of the 3D data, but also their differences are expected. This exactly falls into the research of uncertainty visualization in 3D scalar data.

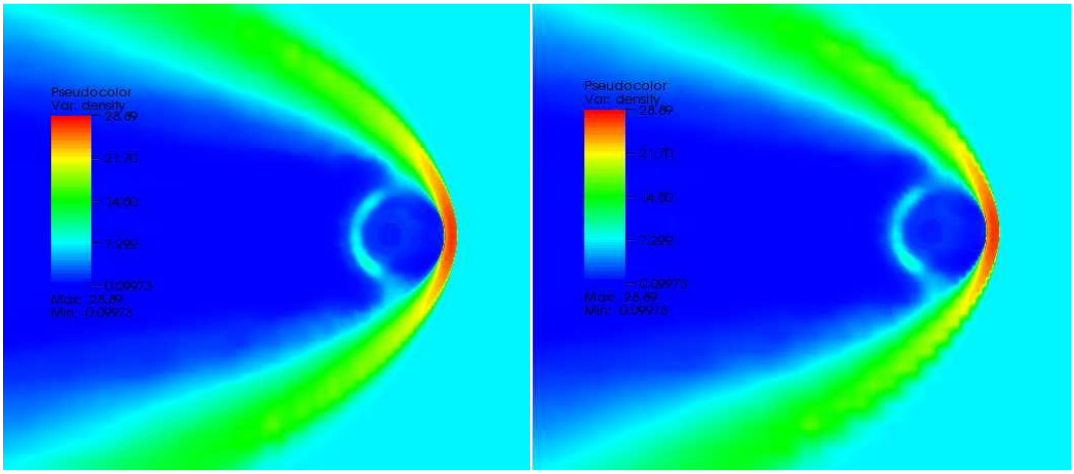


Figure 1.2: Two similar 3D scalar data sets are volume rendered on the left and right (Foulks and Bergeron, 2009).

The second example (Potter, 2010b) comes from the field of Medical Imaging where radiologists, physicians and surgeons have routinely used volume visualization techniques to diagnose and assess patients' data acquired from CT, MRI, Ultrasound or PET (Meibner et al., 2000). Figure 1.3 presents a CT scan data set of a human brain with a brain tumor, and two different volume renderings of the tumor's boundary are shown on the left and right. While there are no huge differences between the

two representations of the tumor boundary, decisions made on either of them may have a significantly different impact on the patient. This is because this tumor grows within the brain's white and grey matter, damage of its surrounding tissues can cause serious consequences to the patient. If we have removed the tumor with excessive tissues, this may result in serious injury of the patient. However, if we have removed the tumor incompletely, the tumor could regrow and repeated and costly surgeries are required. In any case, an advanced uncertainty visualization technique that can inform the surgeons the possible boundary positions of the tumor, as well as their confidence level could be very helpful. By inspecting and understanding the possible boundary positions of the tumor as well as their associated confidence levels, surgeons could improve their surgery planning, or perform more accurate surgery. The patient could also benefit from this by having a safer surgical operation, or a faster recovery time.

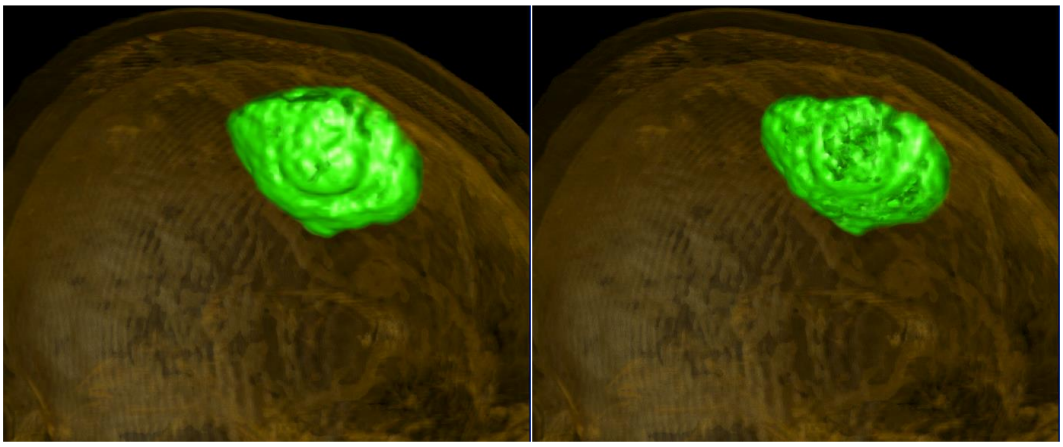


Figure 1.3: A brain tumor data set from CT scan. Two different volume renderings of the possible boundaries of the tumor are shown on the left and right (Potter, 2010b).

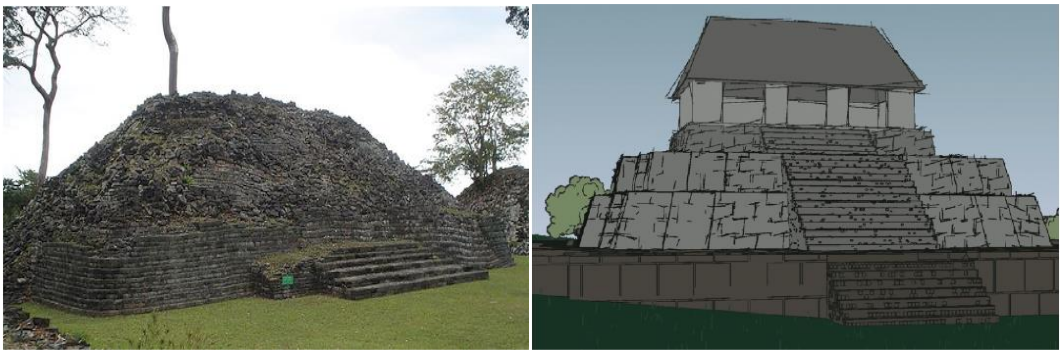
A third example (Potter, 2010b)(Potter, 2003) is from the interdisciplinary field of architecture or archaeological reconstruction, in which one of the biggest problems is the difficulty in communicating the ideas concerning the speculative look of a destroyed site. While verbal communication is always obscure and hard to depict the details, fully realistic renderings of the reconstructed mock-up may mislead the interpretation of viewers. This is because a realistic rendering-based image always tries to be as close as possible to be a photograph, which suggests that it could have been taken from a camera. It may lead viewers to make inappropriate assumptions

about the validity, accuracy as well as the degree of certainty of the speculative reconstruction. Thus, advanced visualization techniques that could reflect the status or certainty information of the reconstructed site are needed. Fortunately, such a problem can be addressed by the research of uncertainty visualization in 3D scalar data, and we prove this by giving a concrete uncertainty visualization solution to this issue. In Figure 1.4(a) we show a photograph of an incomplete Mayan temple. And in Figure 1.4(b) we render a reconstructed 3D model of the temple with its uncertainty information being indicated by line textures.

Figure 1.4(a) shows that the actual temple consists of two levels: a lower base and an upper pile of stone rubble. The reconstruction restored the rubble pile as another level of stone work, with a wooden hut on top of it, as shown in Figure 1.4(b). The uncertainty visualization technique expresses the levels of confidence in the reconstruction by means of line textures. The base structure of the temple is rendered as solid, straight lines. This is due to the fact that the base structure of the temple is still in existence, and thus has a high level of confidence. The next level of work is rendered as a less connected and more random line textures. This is used to express the fact that we only know that this level of work are constructed by stone, as evidenced by the rubble pile, but its actual structure is only inferred from knowledge of nearby temples. Finally, the top wooden hut is rendered as a very sketchy style with empty line textures, which is used to reflect the fact that the actual existence of the hut is not known, and we only assumed its existence based on our findings of wooden fragments from the rubble.

Such an uncertainty visualization technique not only depicts the appearance of the Mayan temple, but also indicates the certainty information of its reconstruction. This will help architecture or archaeological experts to accurately communicate their ideas concerning the speculative look of the temple without making inappropriate assumptions or interpretation.

From the three examples mentioned above it is clear that there is a demand for research of uncertainty visualization in 3D scalar data.



(a) *(b)*

Figure 1.4: (a) A photograph of a Mayan temple. (b) An uncertainty visualization of the possible look of the Mayan temple after reconstruction (Potter, 2010b)(Potter, 2003).

1.1.2 Challenges

Although the problem of uncertainty visualization has been identified in the visualization community (Johnson, 2004)(Chen, 2005)(Johnson et al., 2006)(Laramee and Kosara, 2007)(Johnson and Sanderson, 2003), it is taking a long time to solve this problem, especially for 3D or higher dimensional data. Why is it so hard? Brodlie, Osorio and Lopes (2012) summarize some possible reasons for uncertainty visualization in 3D scalar data.

Uncertainty definition is complex: uncertainty, by its nature, is a complex subject. Indeed as David and Keller (1997) note, even the terminology itself is often unhelpful: "the self-referential problem of uncertainty about uncertainty terminology has been a notable stumbling block in this avenue of inquiry."

A useful step forward here is the definition for uncertainty in geospatially referenced information presented by Thomson et al. (2005). They distinguish nine categories in their definition: accuracy/error, precision, completeness, consistency, currency/timing, credibility, subjectiveness, interrelatedness, and finally, lineage. Although the definition is specific to geovisualization, it would be useful to extend it to the wider area of scientific data visualization.

Uncertainty can be presented in different ways: each voxel in 3D scalar data could correspond to one or multiple scalar function values that are sampled on it, as

explained in Section 3.1.1. In addition to these "certain" data, each voxel could also correspond to some types of "uncertain" data that is presented in different ways:

•*As a singular scalar data*: this is the simplest case in which uncertainty at each voxel is depicted as a singular scalar value. We could consider the uncertainty as the same as the certain data except from its meaning.

•*As multivariate data*: this is an extended version of previous case and instead of each voxel corresponding to a singular scalar uncertainty value, it now corresponds to multiple scalar uncertainty values.

•*As bounded data*: sometimes the uncertainty is depicted as continuous values falling into finite bounds, rather than one or multiple discrete values as described in the previous two cases. Olston and Mackinlay (2002) refer this to bounded uncertainty.

•*As a Probability Density Function (PDF)*: the uncertainty at each voxel can also be supplied as a random variable, say F, with its PDF, $g(f)$, as shown in formula 1.1:

$$prob(a \leq F \leq b) = \int_a^b g(f)df \qquad (1.1),$$

where $prob$ refers to probability, and a is min. and b is max.

Uncertainty adds an extra dimension to the visualization: this can be illustrated by a simple example. In figure 1.5(a), a singular data (2, 8.8) is presented as a marker. Since it only depicts a singular point, it is zero-dimensional (0D). Now assuming that there is an uncertainty associated to the data in the y direction, say 8.8 ± 0.3, then both the data as well as its uncertainty can be presented as an error bar, which is a vertical line and thus is one-dimensional (1D), as shown in Figure 1.5(b). From the example it is clear that uncertainty adds an extra dimension to the visualization, as a marker becomes a line. The same conclusion can be applied to higher dimensional data *i.e.* isolines with uncertainty become areas, and isosurfaces with uncertainty become volumes. Here the word "dimension" is not only limited to the spatial dimensions. It actually refers to any channels that could be used to depict data *i.e.* color.

For low-dimensional data this may cause fewer problems as the data to be visualized only occupies few dimensions and there are many other dimensions available for depiction of uncertainty. But for 3D or higher dimensional data, this poses a big challenge: many dimensions have been used to represent "certain" data, and there may not be enough dimensions available to depict uncertainty. In addition, uncertainty visualization in 3D or higher dimensional data can easily cause visual clutter and complexity, as there are too many dimensions used simultaneously to represent both data and its uncertainty. Consequently it may decrease the viewers' understanding of the data.

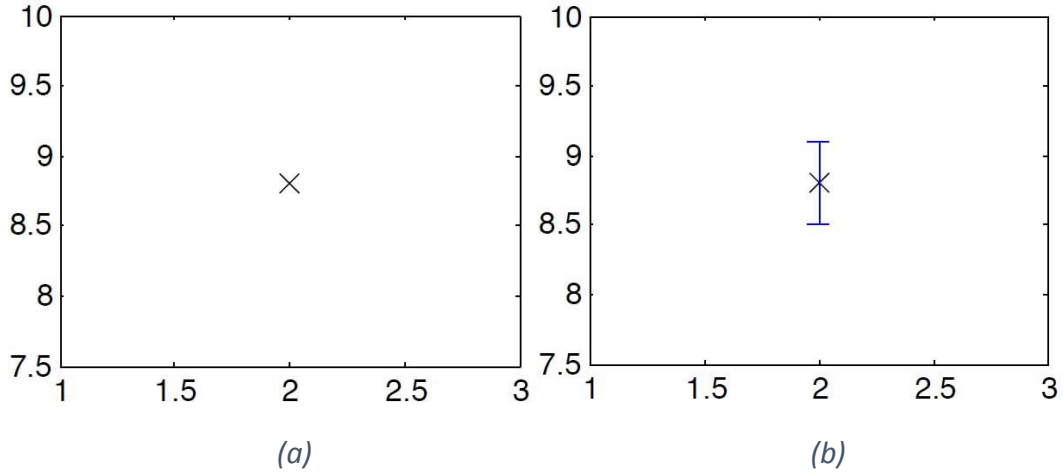


Figure 1.5: An example of uncertainty adding an extra dimension to the visualization (Brodlie, Osorio and Lopes, 2012).

Uncertainty may obscure data presentation: a simple example can illustrate this. For example, assuming we utilize transparency to depict the data with uncertainty. The small transparency is used to represent the data with low uncertainty, while the big transparency is used to represent the data with high uncertainty. As a result, it may be very difficult for the viewers to identify those data with very high uncertainty, as they are too transparent to be observed.

Uncertainty adds another discipline: quite often, a multidisciplinary team that includes domain scientists, visualization scientists, numerical analysts and artists produces the best volume visualization. And now as we consider uncertainty in the volume visualization, a further discipline that deals with uncertainty, statistics, must

be added. We need to increase our knowledge in statistics and enhance collaborate with statisticians in order to improve the rigour of uncertainty visualization.

1.1.3 Research Hypotheses

Based on the state of the art of uncertainty visualization research, three new uncertainty visualization techniques have been proposed in this thesis. These three uncertainty visualization techniques formulate the following three hypotheses:

• **Hypothesis 1:** the proposed Texture uncertainty visualization technique enables users to better identify scalar and error data and provides reduced visual overload and more appropriate brightness than four state-of-the-art uncertainty visualization techniques, as demonstrated using a perceptual effectiveness user study.

• **Hypothesis 2:** the proposed LVIS uncertainty visualization technique enables users to better search max/min scalar and error data than four state-of-the-art uncertainty visualization techniques, as demonstrated using a perceptual effectiveness user study.

• **Hypothesis 3:** the proposed Probabilistic Query uncertainty visualization technique, in comparison to traditional DVR methods, enables radiologists and physicians to better identify possible alternative renderings relevant to a diagnosis and the classification probabilities associated to the materials appeared on these renderings; this leads to improved decision support for diagnosis, as demonstrated in the domain of medical imaging.

In particular, the uncertainty visualization techniques involved in the first two hypotheses focus on the uncertainty generated from data transformation that occurs in the data acquisition stage, thus they belong to the visualization of uncertainty, as described in Section 1.1.1.1. The Probabilistic Query uncertainty visualization technique developed in the third hypothesis focuses on the uncertainty generated from data transformation that occurs within the visualization stage, therefore it belongs to the uncertainty of visualization.

1.1.4 Contributions

Addressing the three hypotheses mentioned above, this thesis provides the following contributions:

• propose a Texture uncertainty visualization technique (applicable to two types of textures), which is empirically shown through an evaluation to have some utilities in certain perceptual aspects, but not as good as the existing hue and texture opacity uncertainty visualization techniques. In addition, the perceptual effectiveness of the 4 existing uncertainty visualization techniques are revealed. These provide useful guidance for future uncertainty visualization design.

• propose a LVIS uncertainty visualization technique, which is shown empirically through an evaluation to have better perceptual effectiveness than other evaluated techniques. In addition, the perceptual effectiveness of the 4 existing uncertainty visualization are revealed. These provide useful guidance for future uncertainty visualization design. Moreover, because of the superiority of the LVIS technique, it is extended to become the Extended Linked Views and Interactive Specification (ELVIS), which is intended to visualize uncertainty in multivariate 3D scalar data.

• propose a Probabilistic Query uncertainty visualization technique, which provides radiologists and physicians with better decision support for diagnosis than traditional DVR. This is demonstrated by three concrete case studies in the domain of medical imaging.

1.2 Thesis Outline

The remainder of this thesis is organized as follows:

Chapter 2 introduces the background and related work to our research. First, a taxonomy and notation of uncertainty visualization techniques are presented. This could give us a better understanding about the scope of this research. Second, a review of uncertainty visualization techniques in scalar data is given based on the taxonomy. Third, some works about evaluation of the perceptual effectiveness of uncertainty visualization techniques are discussed.

Chapter 3 presents the technical foundations and framework of this thesis. First, the foundations of this thesis are introduced, which include some common concepts and

visualization algorithms used in the field of visualization. Second, the framework exploited in this thesis is presented. This includes three main steps: uncertainty data modeling, uncertainty visualization and evaluation, which are defined and explained in great detail.

Chapter 4 presents the corresponding work for hypothesis 1 – the Texture uncertainty visualization technique, which is intended to visualize uncertainty in isosurface rendering. It consists of three main parts. First, the uncertainty data modeling method is introduced, and it is used to generate the data for testing and evaluating the Texture uncertainty visualization technique. Second, the actual Texture technique that includes two types of textures and is applied to the data generated from the first part is presented. Third, the evaluation work that is used to compare the perceptual effectiveness of the Texture technique with other 4 existing uncertainty visualization techniques is shown.

Chapter 5 shows the related work to hypothesis 2 – the LVIS uncertainty visualization technique, which is intended to visualize uncertainty in DVR. This involves four main parts. First, the uncertainty data modeling method exploited to generate the data for testing and evaluating the LVIS uncertainty visualization technique is introduced. Second, the actual LVIS technique, which includes three components and is applied to the data generated from the first part is presented. Third, its evaluation work with respect to perceptual effectiveness in comparison to other 4 existing uncertainty visualization techniques is shown. Finally, its extended work towards visualizing uncertainty in multivariate 3D scalar data is shown.

Chapter 6 shows the related work to hypothesis 3 – the Probabilistic Query uncertainty visualization technique, which is intended to visualize the material intensity classification issue involved in DVR, and thus improves DVR as a better decision making tool. Two sub works are incorporated in this chapter. First, the explicitly probabilistic TF model is introduced, which is used to model the material intensity classification probability and provides the foundation for the Probabilistic Query technique. Second, the actual Probabilistic Query technique is presented. It is applied to three concrete case studies in the domain of medical imaging to illustrate

its ability of providing better decision support for diagnosis, in contrast to traditional DVR.

Finally, Chapter 7 summarizes the entire thesis and points out the limitations existed in these works. An outline of future work related to this thesis is also provided.

Chapter 2 Background and Related Work

This chapter presents the background and related work of this thesis, which includes three main parts. First, a simple but effective taxonomy of uncertainty visualization techniques and its notation are introduced. Second, a review of a variety of uncertainty visualization techniques in scalar data is presented according to the taxonomy. Third, the evaluation works of these uncertainty visualization techniques in scalar data are presented. Note that the review discusses a wider range of works than the 3D case focused in this research. In such a way, we have opportunities to explore the possible extensions of those low-dimensional techniques to our research.

2.1 Taxonomy and Notation of Uncertainty Visualization Techniques

This section introduces taxonomy of uncertainty visualization techniques. This will not only give a clearer concept of the focused scope of this thesis, but also facilitate a better organization of the following review work. Over the past few years, a few taxonomies (Pang, Wittenbrink and Lodha, 1997)(MacEachren et al., 2005)(Skeels et al., 2010)(Thomson et al., 2005) have been proposed, ranging from information visualization to scientific visualization. In this thesis, we introduce a simple but effective uncertainty visualization taxonomy that is proposed by (Brodlie, Osorio and Lopes, 2012) who proposed this taxonomy based on their early work (Brodlie et al., 1992) for the classification of scientific visualization algorithms.

This taxonomy is based on two aspects of data. The first aspect is the dimension of the independent variables, which define the observation space (Theisel, 2005) and often refer to space or time. The second aspect is the type of dependent variables, which are collected from the observation space (Theisel, 2005) *i.e.* temperature or

density and are often known as point (P), scalar (S), vector (V) or tensor (T). It can be expressed by the following notation:

$$U_d^{kt} \quad (2.1),$$

where U refers to the uncertain data; the subscript d refers to the dimensions of the independent variable, which can be 0D, 1D, 2D, 3D or four-dimensional (4D) (space + time); t indicates the type of dependent variable, which can be P, S, V or T; and k indicates the number of multivariate at each voxel. Thus for example, a notation U_3^Scan be explained as the uncertainty in the data consists of a univariate's scalar function of three independent variables.

Based on this taxonomy, the uncertainty visualization techniques can be organized into Table 2.1. Here the temporal dimension is not considered in this table as it usually has different features and resolution than the spatial dimensions (Sanyal et al., 2009). Also, according to (Brodlie, Osorio and Lopes, 2012)(Potter, Rosen and Johnson, 2012), no research has been conducted to explore the uncertainty in the temporal dimension. In particular, the grey part of Table 2.1 indicates the scope of this thesis, which concerns uncertainty visualization in 3D scalar data, for both singular and multiple variables at each voxel.

Table 2.1: Classification of uncertainty visualization techniques. The grey part indicates the scope of this thesis.

Dim.	U^P	U^S	U^{kS}	U^V	U^T
U_0					
U_1					
U_2					
U_3					

2.2 Review of Uncertainty Visualization Techniques in Scalar Data

This section presents various uncertainty visualization techniques in scalar data. For those techniques in 0D, 1D and 2D cases, their possible extensions to a 3D case are discussed. For the techniques in the 3D case, their advantages or disadvantages are discussed.

2.2.1 U_0^S

In this case, the uncertainty data quantified are not associated to any spatial or temporal variables and they are typically presented with many observations of a single scalar variable (Brodlie, Osorio and Lopes, 2012). A classic technique to visualize such uncertainty data is the error bar, which can effectively reveal the upper and lower boundaries of the data by the length of the bar. Another popular technique to visualize this type of uncertainty data is the box plot proposed by Tukey (1977), which can reveal the uncertainty of the data by five summary statistics: the maximum value, the minimum value, the upper quartile, the lower quartile and the median, as illustrated in Figure 2.1.

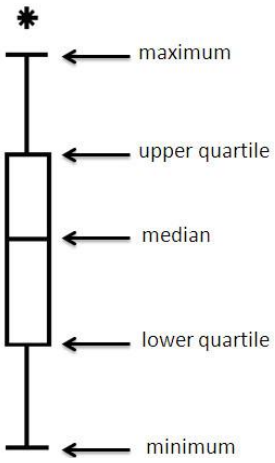


Figure 2.1: Tukey box plot.

There have been many works (McGill, Tukey and Larsen, 1978)(Tufte, 1983)(Benjamini, 1988)(Hintze and Nelson, 1998)(Esty and Banfield, 2003)(Choonpradub and McNeil, 2005) reported on the extensions of modifications of the box plot technique. In a recent paper, a review of these works is presented by Potter et al. (2010), who suggested a summary plot, which incorporates further descriptive statistics such as skew and kurtosis, in addition to the traditional box plot.

Although these techniques mentioned above have been proved to be effective to help quickly summarize the uncertainty characteristics of a data set, and provide a straightforward way to compare different data sets, they cannot be directly integrated into DVR to visualize uncertainty in 3D data (Dinesha, Adabala and Natarajan, 2012). This is because these techniques require adding at least one spatial

dimension to indicate the uncertainty, and this is not possible in DVR where all the three spatial dimensions have been totally occupied for the display of certainty data. Imposing them into DVR will quickly overload the visualization and produce visual clutter (Coninx et al., 2011). However, they may be integrated into the isosurface rendering techniques *i.e.* MC to indicate the uncertainty in 3D data. Also, they could be used accompanied with DVR to reveal the uncertainty *i.e.* summarize the local uncertainty characteristics of the drill down information that is interesting to users.

2.2.2 U_1^S

In this case, the uncertainty data are quantified over one spatial variable and there is a variety of ways to indicate such type of uncertainty data. A series of techniques for this case are summarized and a formal user study to compare their perceptual effectiveness are performed by Sanyal et al. (2009). Figure 2.2 illustrates some of these techniques.

Figure 2.2(a) shows that uncertainty can be depicted by the scaled sizes of a glyph. The bigger the size of the glyph, the higher the value of uncertainty will be. Conversely, the smaller the size of the glyph, the lower the value of uncertainty will be. Figure 2.2(b) presents an image where the uncertainty is depicted by the color attribute of a glyph. While the lighter color of the glyph indicates a higher uncertainty, the darker color of the glyph indicates a lower uncertainty. The same conclusion can be drawn from the technique illustrated in Figure 2.2(c). However, instead of using an extra glyph's color to depict uncertainty, this technique directly uses the data lines' color to depict it. Figure 2.2(d) presents an image which uses the error bars to depict uncertainty. While the long error bars indicate high uncertainty, the short error bars indicate low uncertainty. Consequently the error bar technique is extended successfully for the 1D case. Finally Figure 2.2(e) illustrates an image which applies the animation technique to the 1D data lines to depict uncertainty. While the large amplitude of the animation indicates high uncertainty, the small amplitude of the animation indicates low uncertainty.

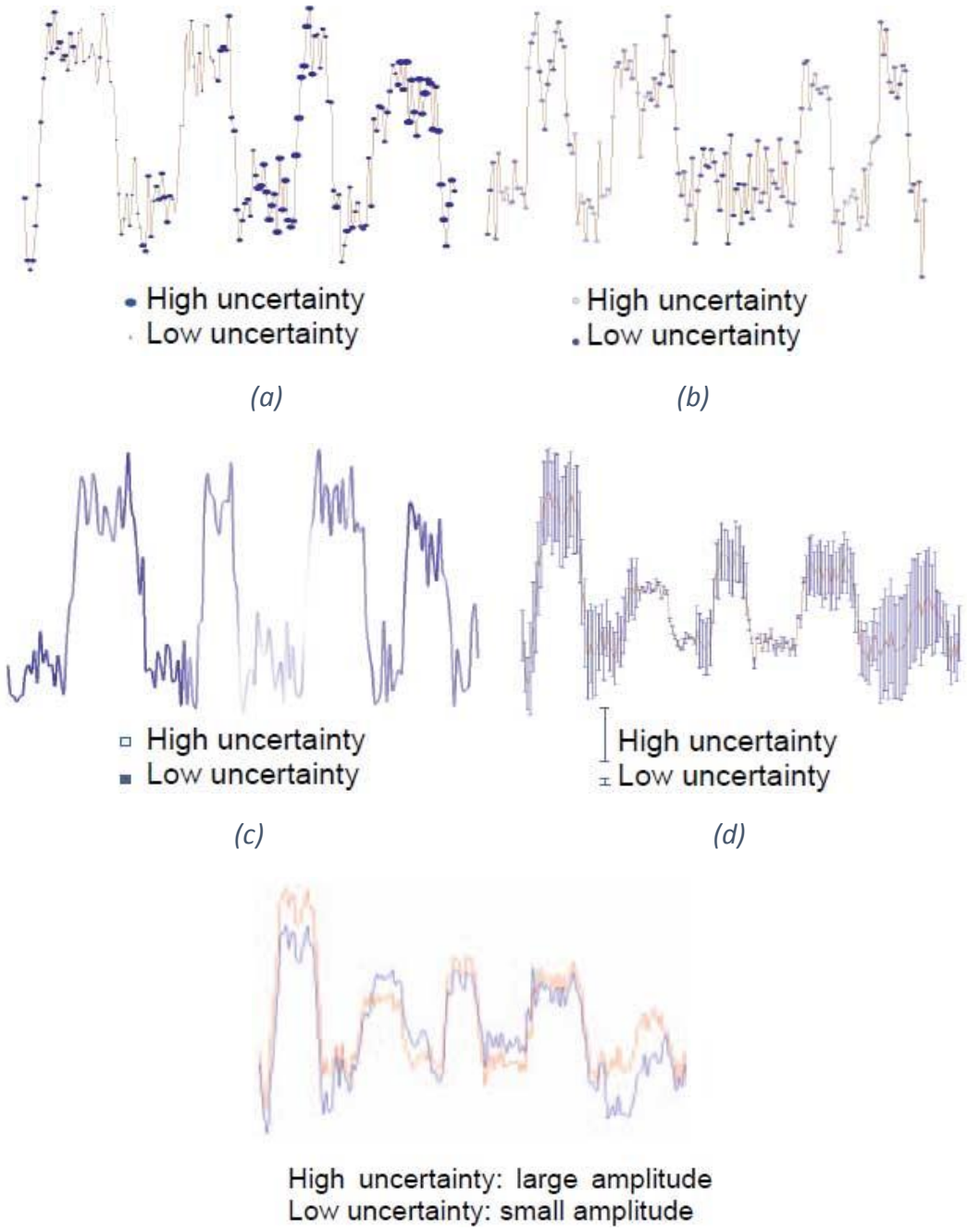


Figure 2.2: Uncertainty visualization techniques for 1D case (Sanyal et al., 2009).

Although the above-mentioned two glyph techniques and the error bars technique are useful for uncertainty visualization for the 1D case, they cannot be directly integrated into DVR for the 3D case. This can be attributed to either or both of the following two reasons. First, as explained in the previous section, they require adding at least one spatial dimension to the existing DVR for the indication of uncertainty, and this will quickly overload the visualization and produce visual clutter (Coninx et

al., 2011). Second, they typically involve some modification of geometric primitives and their associated attributes *i.e.* color to indicate the uncertainty; unfortunately DVR does not produce any intermediate geometric primitives for them to modify (Djurcilov et al., 2001)(Djurcilov et al., 2002). However, these techniques may be integrated into the isosurface rendering techniques or used combined with DVR to show uncertainty in 3D data. In terms of the above-mentioned color-mapping technique and the animation technique, it is possible to integrate both into DVR for uncertainty visualization, as discussed in Section 2.2.4.

2.2.3 U_2^S

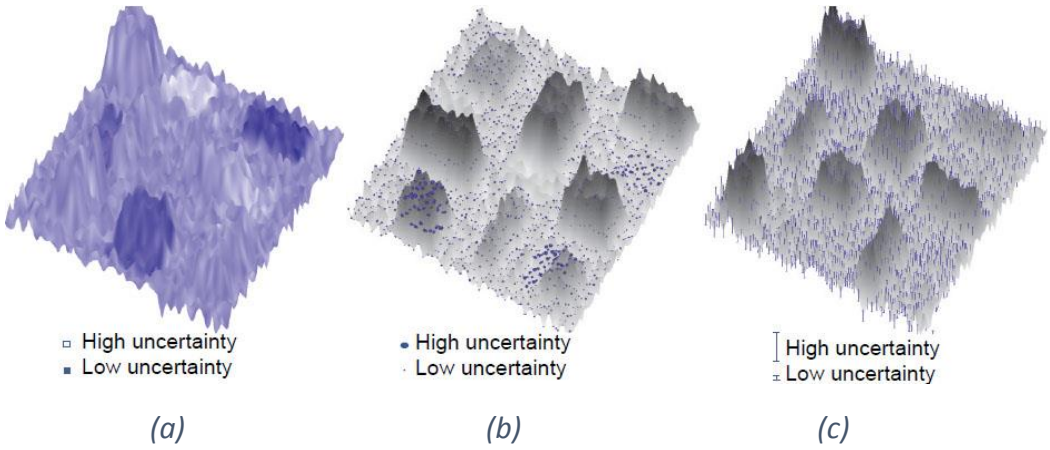


Figure 2.3: Uncertainty visualization techniques for 2D case. (a) color-mapping data with uncertainty, (b) the scaled size of a glyph, (c) traditional error bars (Sanyal et al., 2009).

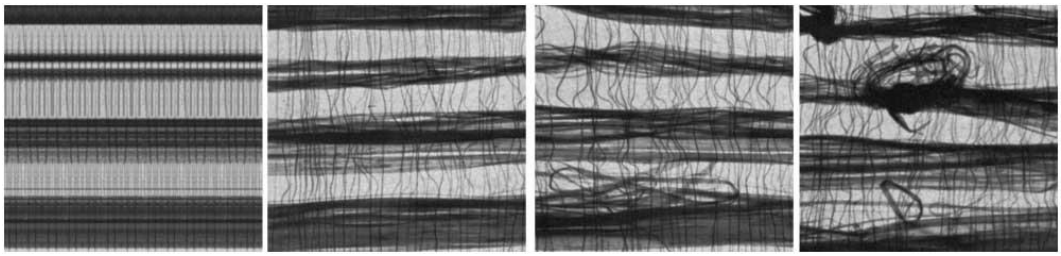


Figure 2.4: Increasing irregularity in a texture pattern can intuitively represent increasing levels of uncertainty (Interrante, 2000).

This is the case where uncertainty data are quantified in terms of two spatial variables. A variety of techniques have been proposed for this case. Probably the most common technique for this case is color-mapping (Sanyal et al., 2009)(Lodha et al., 1996)(Wittenbrink, Pang and Lodha, 1995)(Potter et al., 2009)(Sanyal et al., 2010)

where the degree and presence of uncertainty data are indicated by the color, as illustrated in Figure 2.3(a). Another common technique for this case are glyphs (Sanyal et al., 2009)(Pang, Wittenbrink and Lodha, 1997)(Lodha et al., 1996)(Wittenbrink, Pang and Lodha, 1995)(Potter et al., 2009)(Zehner, Watanabe and Kolditz, 2010), where the uncertainty data are indicated by the size, shape, length, color etc. attributes of the glyphs, as illustrated in Figure 2.3(b). The error bar technique is also obviously applicable for this 2D case and it is extended by Sanyal et al. (2009) from the 1D case to the 2D case, as illustrated in Figure 2.3(c). A research that makes use of the carefully selected texture patterns to depict different levels of uncertainty is performed by Interrante (2000), as illustrated in Figure 2.4. Animation is a commonly used technique in visualization and two works that apply it to indicate the uncertainty are suggested by Brown (2004) and Ehlschlaeger, Shortridge and Goodchild (1997). Figure 2.5 illustrates an example of this technique where two frames are selected from an animation and the uncertainty is indicated by the oscillated height in the dotted bordered rectangle. A 2D annotation work that superimposed a grid on top of an image and the uncertainty is indicated by subtle modifications of the grid lines is suggested by Cedilnik and Rheingans (2000), as illustrated in Figures 2.6(a) and (b). More recently, there have been some works (Osorio and Brodlie, 2008)(Pothkow and Hege, 2011) that extend the concept of a 2D contour and show the uncertainty regions across the spatial domain by modifying the contours' color, thickness and opacity. Figure 2.7 illustrates one example of this technique where each highlighted pixel in this image indicates that there is a 65% or greater probability of the pixel having a value close enough to the contour value h = 0. Also, there are two works (Potter et al., 2009) and (Sanyal et al., 2010) that combine various visualization techniques into multiple linked views to reveal the clear characteristics of uncertainty, as illustrated in Figure 2.8.

Because the above-mentioned techniques of color-mapping, glyphs, error bars and animation are extended from the 0D or 1D case and their expansibilities to the 3D case have been introduced in the previous two sections, here we avoid repeated discussions. For the 2D annotation technique, its advantage is that the data representation has minimal interference induced by the uncertainty data. This is very

useful for those cases where it is more important to display data rather than uncertainty information. Although it is possible (Cedilnik and Rheingans, 2000) to extend this technique into 3D cases, no research has been performed towards such extensions. As for the multiple linked views technique, it is significant. Because it could not only yield a clearer data presentation by displaying diverse of information in multiple views, but also combine multiple visualization techniques together for quantitative data analysis. Unfortunately, no research has been reported to extend this technique for the 3D case. For the texture technique, it is possible to extend it to the 3D case, but only a few investigations (Rhodes et al., 2003)(Djurcilov et al., 2002) have been conducted for this study. In terms of the remaining contour technique, some works have reported to extend it to the 3D case, and they will be reviewed in the next section.

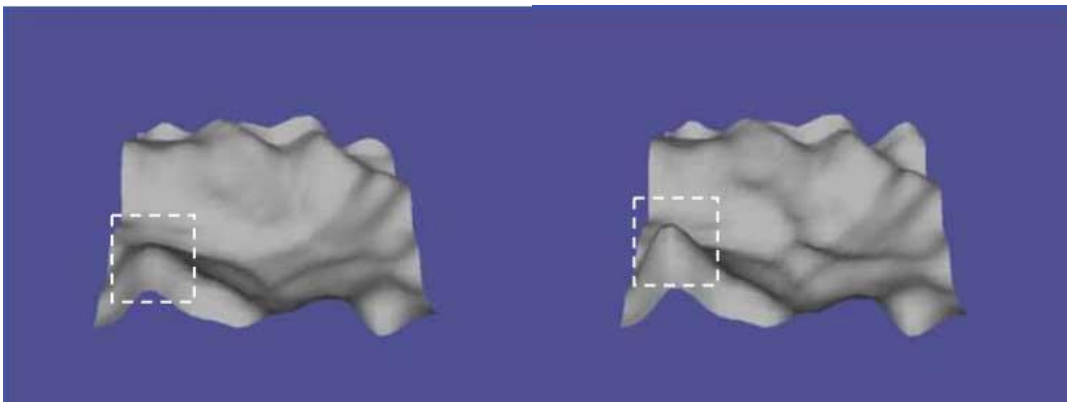
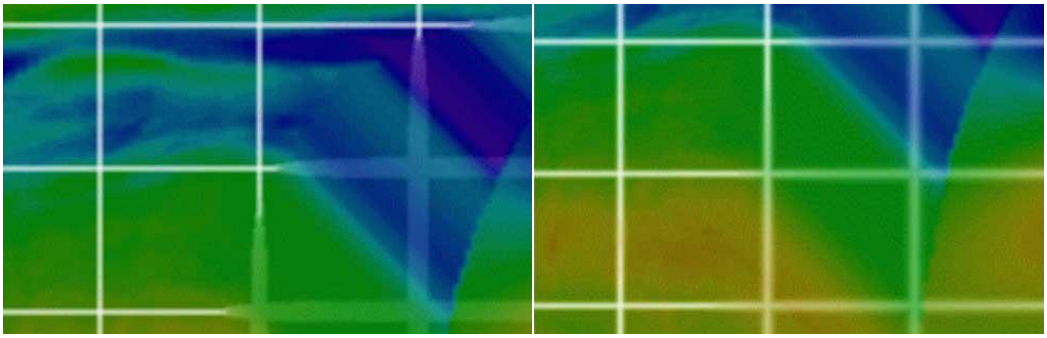


Figure 2.5: Two frames from an animation to reveal the uncertainty that is indicated by the oscillated height in the dotted bordered rectangle (Brown, 2004).



(a) *(b)*

Figure 2.6: A 2D annotation technique with uncertainty is indicated by (a) the width variation of the grid lines, (b) the exponential sharpness of the grid lines (Cedilnik and Rheingans, 2000).

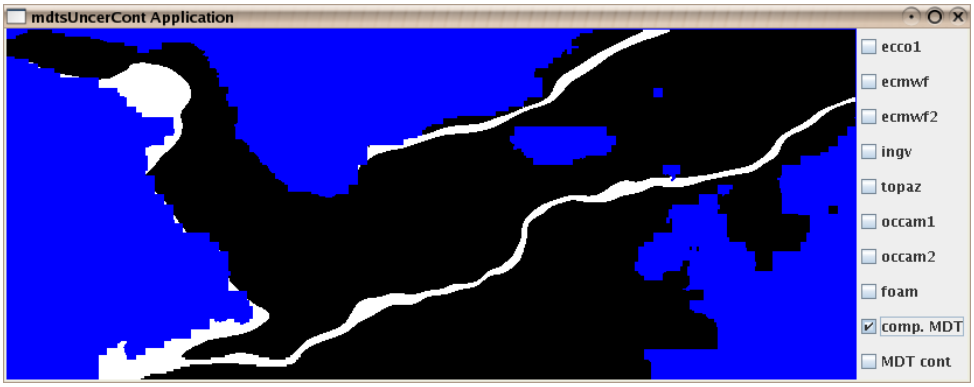


Figure 2.7: The contour technique with uncertainty is indicted by contour color (Osorio and Brodlie, 2008).

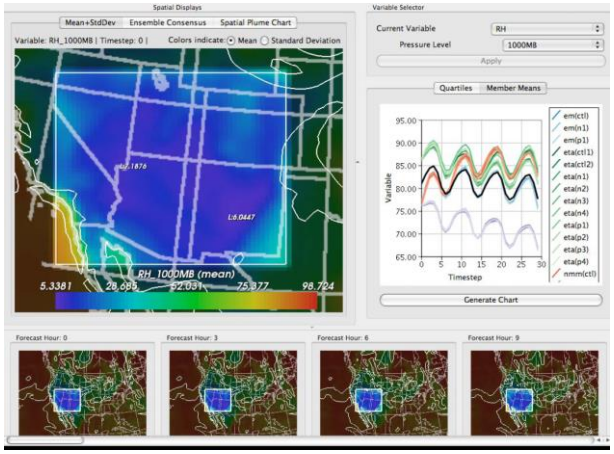
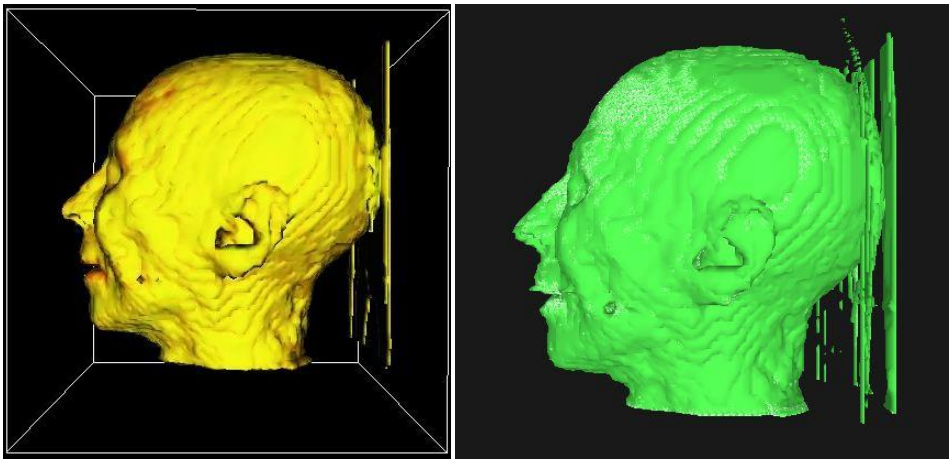


Figure 2.8: The multiple linked Views technique (Potter et al., 2009).

2.2.4 U_3^S



(a) *(b)*

Figure 2.9: Isosurface rendering of a cadaver head data set with uncertainty is indicated by (a) hue, (b) texture opacity (Rhodes et al., 2003).

This is the case where the scalar uncertainty data are depicted by three spatial variables. When moving to the 3D case, the visualization channels available for uncertainty data have been significantly decreased due to the fact that the 3D space has been in use to visualize the "certain" data, either by isosurface rendering or DVR. This poses huge challenges for uncertainty visualization.

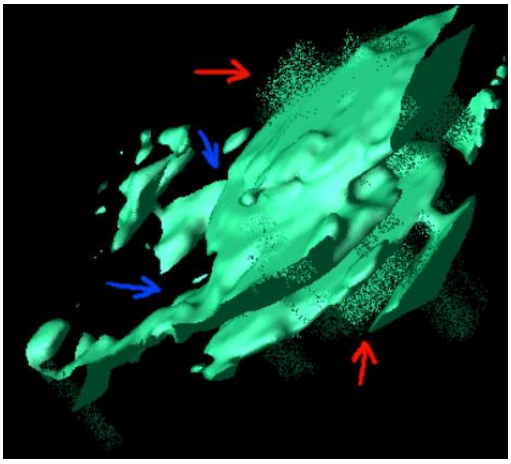
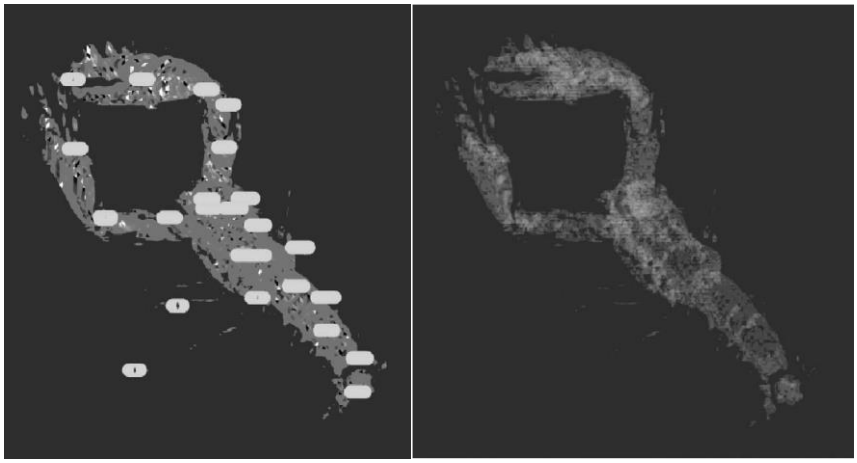


Figure 2.10: Uncertainty in the real tumor data set is indicated by the displacement distance of surface points, in the direction of the surface normal (Grigoryan and Rheingans, 2002)(Grigoryan and Rheingans, 2004).

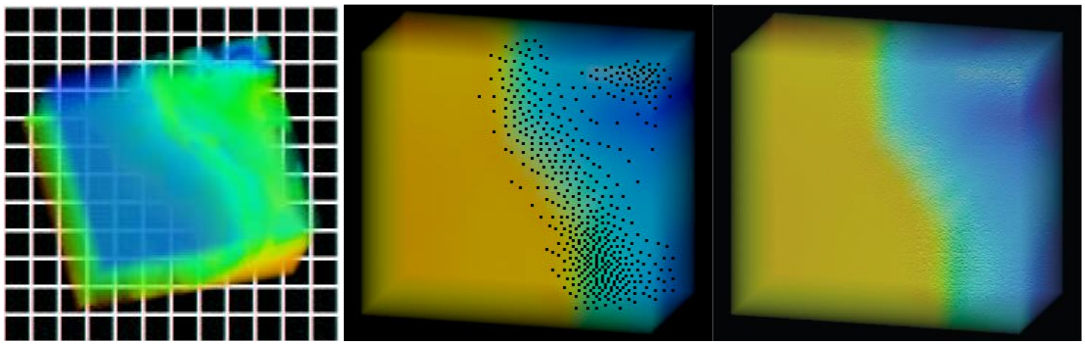


(a) *(b)*

Figure 2.11: Isosurface rendering of lobster data with local uncertainty is represented by (a) cylinder glyphs, (b) transparency (Newman and Lee, 2004).

Isosurface: there is research (Rhodes et al., 2003) that uses hue and texture opacity to indicate uncertainty, as illustrated in Figures 2.9(a) and (b). A technique which uses displacement of the surface points to indicate uncertainty is suggested by Grigoryan and Rheingans (2002)(2004). As illustrated in Figure 2.10, the areas pointed to by red

arrows indicate high uncertainty with big displacement of the surface points, while those pointed to by blue arrows indicate low uncertainty with small displacement. A work which involves using glyphs and transparency to depict uncertainty is presented by Newman and Lee (2004), as illustrated in Figures 2.11(a) and (b). In contrast to the techniques mentioned above which visualize uncertainty existing within data, several methods that address positional uncertainty that is caused by isosurface rendering itself are proposed by (Pothkow and Hege, 2011)(Pothkow, Weber and Hege, 2011).

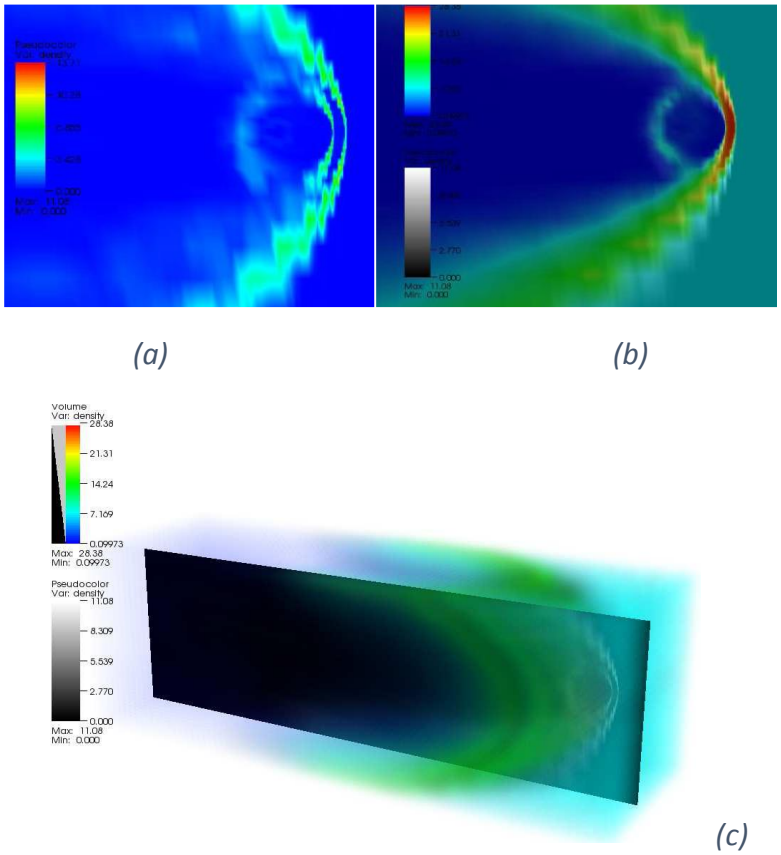


(a) *(b)* *(c)*

Figure 2.12: Uncertainty visualization of ocean data with uncertainty. Uncertainty as indicated by (a) transparency, (b) density of holes and (c) texture contrast (Djurcilov et al., 2001)(Djurcilov et al., 2002).

DVR: uncertainty visualization gets even harder in this case because all three spatial variables plus color and transparency have been in use for the "certain" data, and the visualization channels left for depiction of uncertainty are less. Early work by Djurcilov et al. (2001)(2002) explored a series of techniques including using transparency, holes and textures to visualize uncertainty in 3D data. According to whether these techniques can be directly integrated into DVR or not, they classify them into two general methods, namely inline and post-process. The transparency technique involves mapping uncertainty directly into the volume rendering equation, and Figure 2.12(a) illustrates its result. The holes technique involves compositing the DVR of data with the dithered DVR of uncertainty, and Figure 2.12(b) illustrates its result. As for the texture technique, it involves adjusting each pixel's brightness of the DVR of data to the corresponding pixel's brightness from the texture map, according to its uncertainty value, and Figure 2.12(c) illustrates the corresponding result. There is research (Foulks and Bergeron, 2009) that uses color-mapping,

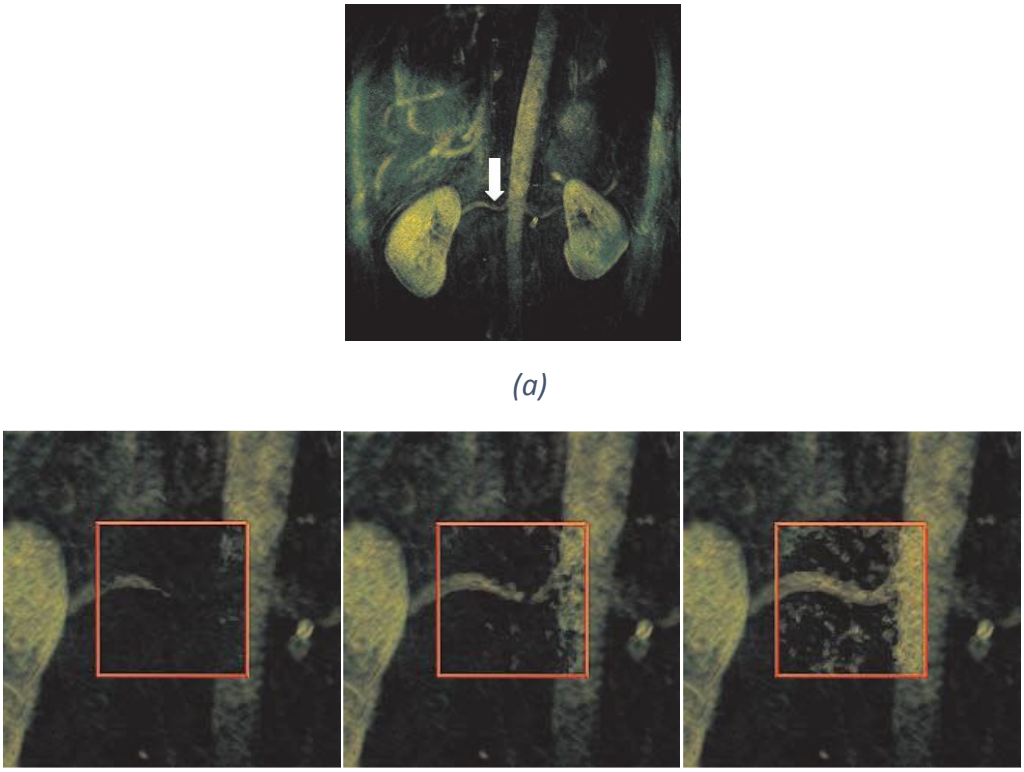
overlay and hybrid renderings to reveal uncertainty in 3D data. The color-mapping technique is mapping uncertainty data directly to the color and the “exact” data are not represented in the final image, as illustrated in Figure 2.13(a). The overlay technique involves superimposing a semi-transparent colorized DVR of data with a semi-transparent gray DVR of uncertainty, as illustrated in Figure 2.13(b). The hybrid renderings technique involves compositing the semi-transparent colorized DVR of data with a gray slice rendering of uncertainty, as illustrated in Figure 2.13(c). Not surprisingly, the time dimension can be very useful to depict uncertainty in this case. (Lundstrom et al., 2007)(Lundstrom, 2007) integrated the animation technique into DVR and used it to indicate the data classification uncertainty caused by traditional TFs. Figure 2.14 illustrates a specific example of this technique. In Figure 2.14(a), a traditional DVR of a renal angiograph is presented, and we can see from the arrow that a vessel stenosis may occur. Figures 2.14(b), (c) and (d) present three frames from the proposed animation technique, and from them we can see that there is no stenosis in the vessel.



(a) *(b)*

(c)

Figure 2.13: Uncertainty visualization of magnetohydrodynamics with uncertainty (Foulks and Bergeron, 2009).



(a)

(b) *(c)* *(d)*

Figure 2.14: Uncertainty visualization of data classification uncertainty. (a) A traditional DVR with a suspected vessel stenosis as denoted by the arrow. (b)-(d): Three frames from the animation technique indicate there is no stenosis in the vessel (Lundstrom et al., 2007)(Lundstrom, 2007).

Although these techniques can intuitively depict uncertainty without requiring extra training of users, they all suffer from one or more drawbacks. The disadvantage of the above-mentioned displacement technique of surface points, as shown in Figure 2.10, is that the appearance of data may become unrecognizable to users when the amount of uncertainty increases. The disadvantage of the glyphs technique, as shown in Figure 2.11(a), is that partial data representation is always occluded by the glyphs and this is not good for users to identify the data. Similarly, the disadvantage of the transparency technique, as shown in Figure 2.11(b) and Figure 2.12(a), is that the data presentation is always weakened by the transparency that is used to indicate uncertainty. As for the holes technique as shown in Figure 2.12(b), its disadvantage is that data representation is missing in the regions of holes. For the color-mapping techniques shown in Figure 2.13(a), its disadvantage is that the data presentation is completely invisible for users' analysis. In terms of the animation technique as shown in Figures 2.14(b), (c) and (d), its disadvantage is that the

movement and flickering in the animation can easily cause visual fatigue to users. Moreover, a main problem in all the techniques mentioned above for the 3D case is that they only give qualitative indications rather than quantitative indications of uncertainty data. This may be sufficient for some cases where only relative distinctions of uncertainty are needed. However, this will be less useful for those cases where quantitative distinctions of uncertainty are required *i.e.* in the medical imaging filed, radiologists and physicians want to quantitatively assess the uncertainty caused by data classification during DVR so that they can draw more appropriate conclusions for the diagnosis of patients.

2.2.5 Multivariate Uncertainty Data

While most works in the field of uncertainty visualization have focused on univariate uncertainty data, as reviewed in the previous four sections, little research has been reported for multivariate uncertainty data (Xie et al., 2006)(He, Yang and Xu, 2011). This section reviews the works that are related to visualization of multivariate uncertainty data.

2.2.5.1 U_0^{kS}

This is the case where a set of uncertainty data have been quantified for multiple variables, but they are not related to space and time. Uncertainty visualization of this type of data usually extends existing techniques in multivariate visualization (Wong and Bergeron, 1994)(Chan, 2006), which is a typical research topic in the information visualization (Spence, 2007)(Dix, 2013).

There is research (Xie et al., 2006) that extends three well-known multivariate visualization techniques of parallel coordinates (PC), scatter plot matrices and star glyphs, to incorporate uncertainty information. Furthermore, different visual encodings *i.e.* line width, point size, hue, saturation and lightness (HSL) for the indication of uncertainty are compared. Two works that augment PC to incorporate uncertainty by means of blurring, opacity and color are reported by (Feng et al., 2010)(Feng, 2010) and Ge et al. (2009). Again, these techniques tend to use certain visual channels, *i.e.* color and line width, to depict uncertainty and thus can only make qualitative indications.

2.2.5.2 $U_{>0}^{kS}$

This is the case where a set of uncertainty data are quantified for multiple variables, over 1D, 2D or 3D spatial domains. It is already an identified research problem (Johnson, 2004) to represent multiple separate variables into a single image over spatial domains, and the addition of uncertainty data makes the problem even harder.

The only research reported for this case is from (Haroz, Ma and Heitmann, 2008) (Haroz and Heitmann, 2008), who combined a PC control interface together with a spatial view to visualize the uncertainty between different simulation codes in cosmological particle simulations (Heitmann et al., 2005)(Heitmann et al., 2007) (Fryxell et al., 2000)(Warren and Salmon, 1993), as illustrated in Figure 2.15. A very important advantage of this technique is that it allows quantitative exploration of uncertainty by using the PC as a query interface.

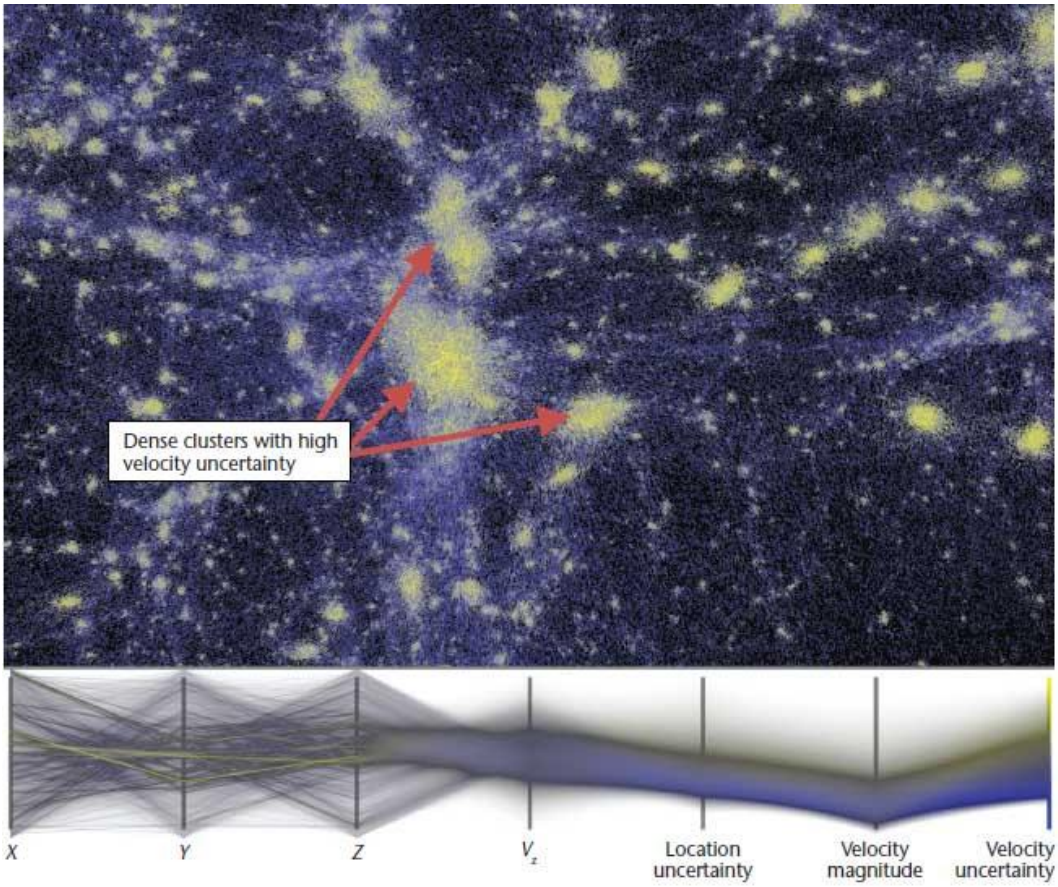


Figure 2.15: Uncertainty visualization consists of a PC control interface plus a spatial view (Haroz, Ma and Heitmann, 2008)(Haroz and Heitmann, 2008).

2.3 Evaluation of Uncertainty Visualization Techniques in Scalar Data

An often overlooked aspect in the field of visualization is evaluation, and this is also the case in uncertainty visualization (Potter, Rosen and Johnson, 2012). Lack of evaluation will not benefit future uncertainty visualization design and development, as no valuable knowledge and guidelines are summarized and formulated from the existing solutions.

Sanyal et al. (2009) presented a user study that involved four user tasks to evaluate the perceptual effectiveness of four commonly used uncertainty visualization techniques, including traditional error bars, scaled size of glyphs, color-mapping on glyphs and color-mapping on the data surface in 1D and 2D data, and there are four valuable findings from their study. First, they found a significant difference in user performance between searching for locations of high and searching for locations of low uncertainty. Second, they found that the error bars technique consistently underperformed throughout the experiment. Third, they found that the scaled size of glyphs technique and the color-mapping on the data surface technique performed reasonably well. Finally, they found that the efficiency of most of these techniques are highly dependent on the tasks performed. Newman and Lee (2004) evaluated four aspects of perceptual effectiveness for four new and four existing isosurface rendering-based uncertainty visualization in 3D data. Their study found that while each technique has some utility in their scenario, the new multi-point glyph and the existing ball and arrow glyph techniques appear to be most advantageous.

Although the above-mentioned two evaluations involve data range from 1D to 3D and their findings are very valuable for future uncertainty visualization design, no evaluation has been reported for DVR-based uncertainty visualization techniques. In addition, the isosurface rendering-based uncertainty visualization evaluation proposed by Newman and Lee (2004) covered most non-integrated techniques rather than those integrated techniques. Thus, more research is needed to evaluate the perceptual effectiveness for both DVR and isosurface-based uncertainty visualization techniques.

2.4 Summary

This chapter introduces the background and related work of this thesis, which includes three main parts. First, a simple but effective taxonomy of uncertainty visualization techniques and its notation are introduced. Second, a review of a variety of uncertainty visualization techniques in scalar data is presented according to the taxonomy. Third, the evaluation works of these uncertainty visualization techniques in scalar data are presented.

In the next chapter, we will introduce this thesis' technical foundations and unified framework used for testing each hypothesis.

Chapter 3 Thesis Technical Foundations and Framework

This chapter consists of two main parts. Section 3.1 introduces the technical foundations of this thesis to facilitate a better understanding of our research. This includes some commonly used concepts in the field of visualization such as 3D scalar data, isosurface rendering, DVR, etc. Section 3.2 presents the thesis' framework that is used throughout our research and includes three main steps: uncertainty data modeling, uncertainty visualization and evaluation. Each main step will be clearly defined and explained with respect to how it is implemented in every hypothesis.

3.1 Foundations

This thesis builds upon some common concepts and algorithms used in the visualization community, including 3D scalar data, volume visualization, isosurface rendering/indirect volume rendering, MC, DVR, volume ray-casting. The following 3 subsections will describe them in detail.

3.1.1 3D Scalar Data

A univariate 3D scalar data set, or often referred to as univariate volumetric scalar data set, ideally is considered to represent a continuous scalar function in a three-dimensional space, formulated mathematically as (Lundstrom, 2007):

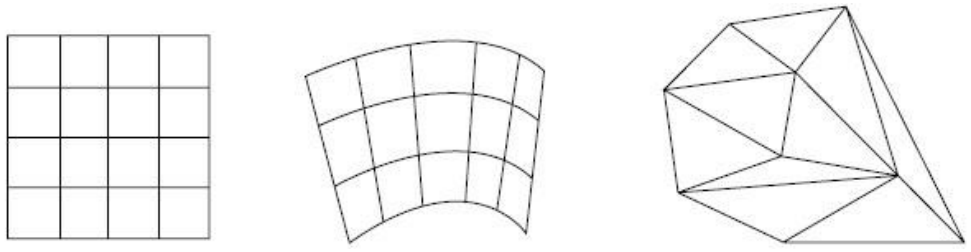
$$\mathrm{f}: R^3 \rightarrow R \qquad (3.1).$$

However, data sets acquired from measurements or scientific simulations in the real world do not have continuous values, and they are limited to the points where measured or simulated values have been sampled or collected. In general, these points of data sets can be sampled at purely random locations in space and thus construct to different grid types. Figure 3.1 illustrates some typically grid types in two dimensions. A common case among them is the regular Cartesian gird, where the

data sets sampled on the grid consist of cubes. In other words, data sets are sampled along three directions x, y and z, and in each direction, the distance d_i where d refers to distance, and i refers to any a direction of x, y and z, between any two adjacent points are equal, but d_x, d_y and d_z are not necessarily equal (Tang, 1999). This is the type of data exploited in this thesis, and data sets acquired from biomedical scanners such as CT, MRI, PET, ultrasound and many scientific simulations typically belong to this type. Figure 3.2 illustrates a specific example of them. In such a data set, every cube is called a cell, such as those highlighted in red. Every sample point distributed at the corner of the cell, as indicated by the red rectangle, is called a voxel, a name stemming from its 2D counterpart pixel. As each cell has 8 corners, it includes 8 voxels.

The value sampled at every voxel can represent many different physical properties or variables. In medical imaging, the property usually refers to the material intensity obtained from the biomedical scanners, with different intensities corresponding to different materials of human body. Examples of properties from measurements and scientific simulations in other domain include temperature, density and electrostatic potential, etc. If each voxel of a data set only has one property, this type of data set is known as univariate 3D scalar data set. Otherwise, if each voxel of a data set has more than one property, it is known as multivariate 3D scalar data set.

During the process of visualization, these discrete data sets are often enriched through interpolation. In this way, an approximation of the continuous scalar functions they represent can be achieved.



Regular *Curvilinear* *Unstructured*

Figure 3.1: Typical grid types in two dimensions.

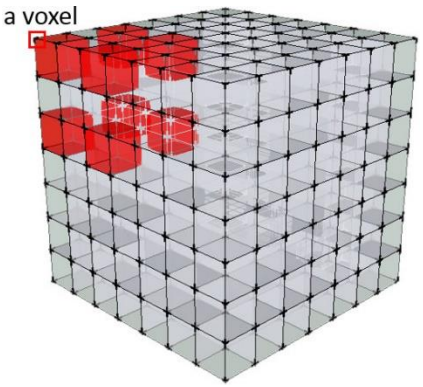


Figure 3.2: An example of 3D scalar data set with regular grids.

3.1.2 Volume Visualization

Volume visualization has been the most active sub-area of research during last 20 years in scientific visualization (Celebi, 2013), and it involves the process of mapping 3D scalar data to 2D image so that users can visualize 3D scalar data on 2D computer screen. Roughly speaking, volume visualization can be classified into two main approaches, as illustrated in Figure 3.3 (for a more detailed classification, please refer to (Jurgen et al., 1995)(Dachille, 1997)(Swan, 1998)(Yagel, 1996)(Law, 1996)). The first approach is called isosurface rendering, or indirect volume rendering (IVR), while the second approach is called DVR. Both rendering approaches have been used in this thesis. The following two sections will introduce these two approaches in greater detail.

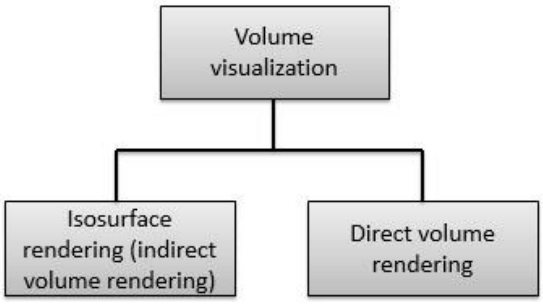


Figure 3.3: Classification of volume visualization.

3.1.2.1 Isosurface Rendering – IVR

Isosurface rendering refers to a set of algorithms or techniques that can extract and display surfaces whose scalar value equals to a given constant from 3D scalar data. By isolating the interested isosurface from other irrelevant features within 3D scalar data, it enables users to more quickly obtain insight or make important decisions

(Johnson and Huang, 2009). The reason that it is referred to as IVR can be explained through its visualization pipeline, as illustrated in Figure 3.4(a). We can see that in comparison to the one-step DVR pipeline which directly renders the input 3D scalar data into a 2D image, as shown in Figure 3.4(b), the isosurface rendering's pipeline does not directly render them. Instead, it first extracts relevant surfaces from 3D scalar data according to a given constant, and then renders these surfaces into a 2D image. Therefore, it is deemed as IVR.

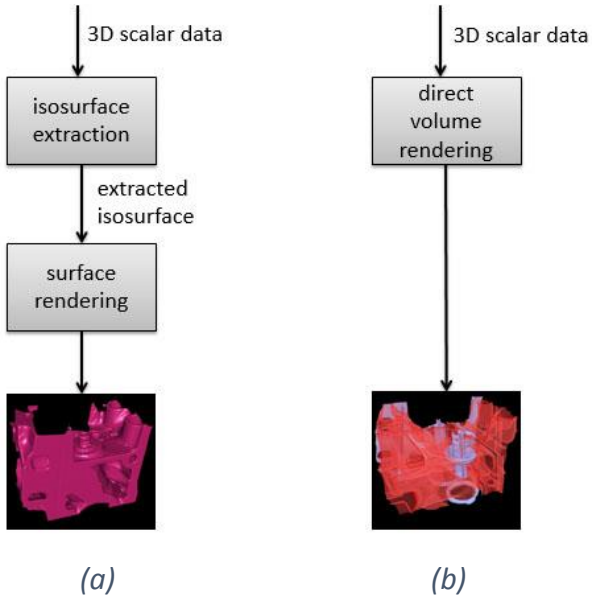


Figure 3.4: Visualization pipeline of (a) isosurface rendering vs, (b) DVR.

MC (Lorensen and Cline, 1987) is probably the most classic algorithm to implement isosurface rendering. Its basic idea is just like implied by its name, which is to march through every cell of a 3D scalar data set, and within every cell create triangle meshes according to the user given constant value *isovalue* to construct part of the isosurface. By connecting all these triangle meshes constructed within every cell together, an approximation of the desired isosurface from the 3D scalar data set is formed. More specifically, it involves the following 8 steps (Moller, 2010):

1. Consider a cell of a 3D scalar data set, as highlighted in Figure 3.2. To facilitate explanation, we label this cell here in the following manner, as shown in Figure 3.5. The 8 red numbers correspond to indices of the 8 voxels, beginning with 0, and the blue 12 numbers correspond to indices of the 12 edges, also beginning with 0.

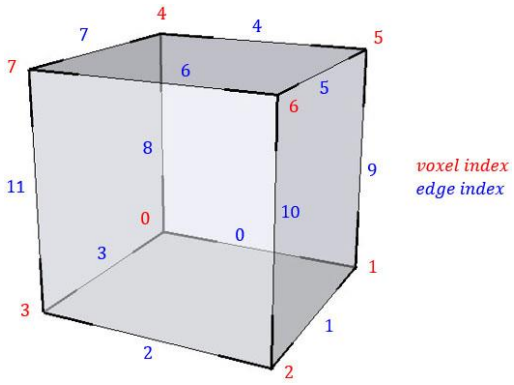


Figure 3.5: A labelled cell.

2. Classify every voxel of the cell according to whether its scalar value is greater than the user given *isovalue*, or less than or equal to *isovalue*. If its scalar value is greater than *isovalue*, we say that this voxel is outside the isosurface. Otherwise we say that this voxel is inside the isosurface.

3. Build the cell's *cubeIndex* according to every voxel's binary classification. This *cubeIndex* can be implemented by a single byte including 8 bits, with each bit representing this classification status of a voxel of this cell. If a voxel is outside the isosurface, its corresponding bit will be set to 0. Otherwise its corresponding bit will be set to 1. For example, assuming that the value at voxel 3 is below *isovalue*, and all values at other voxels are above it. Then the *cubeIndex* for this case can be illustrated in Figure 3.6.

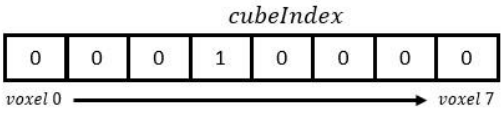


Figure 3.6: The cubeIndex for this case where the scalar value at voxel 3 is below isovalue, and all scalar values at other voxels are above isovalue.

4. Get a list of edges at which the isosurface intersects with the cell from a pre-defined table using the *cubeIndex* (table[cubeIndex]). This pre-defined table stores all 2^8=256 (as every voxel has two classification status, and a cell includes 8 voxels) possible ways in which the isosurface can intersect the cell. Therefore through this table it is possible to look up a specific edge list for a given *cubeIndex*. By symmetry these 256 ways can be reduced to 15 cases, which are illustrated in Figure 3.7. Continue with the above example, by

looking up this table we can obtain that the edge list for the *cubeIndex* as shown in Figure 3.6 should be 2, 3 and 11, as illustrated in Figure 3.8.

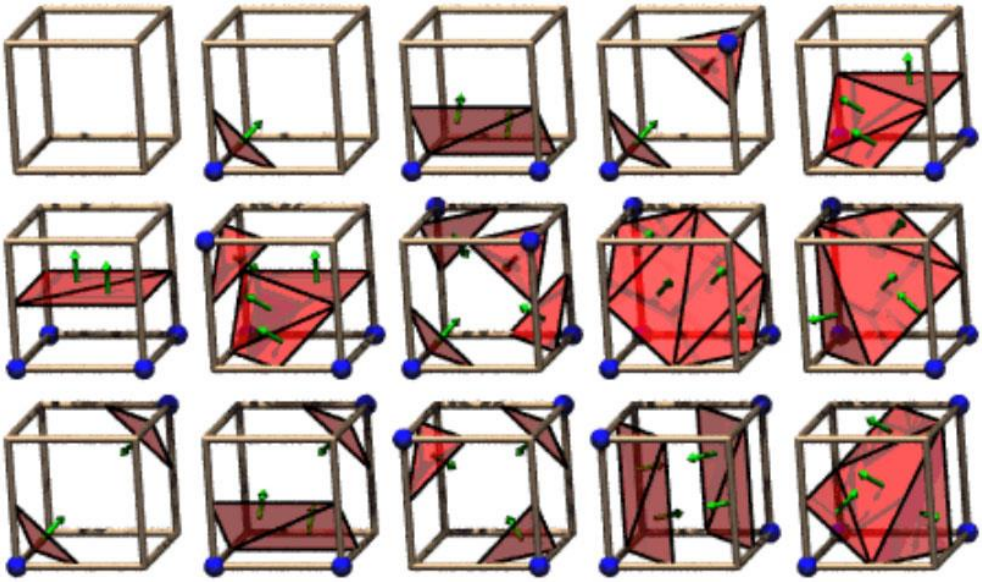


Figure 3.7: By symmetry there are 15 possible ways in which the isosurface can intersect a cell (Moller, 2010).

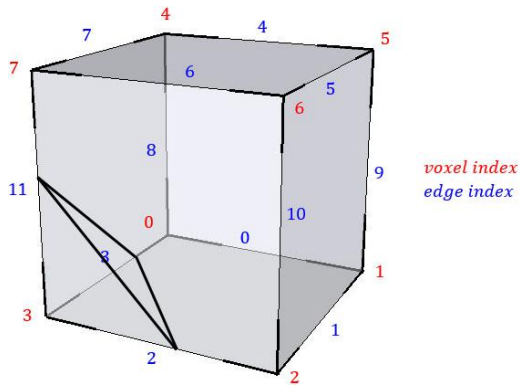


Figure 3.8: Edge 2, 3 and 11 intersect this cell when voxel 3 is inside the isosurface, and other voxels are outside the isosurface.

5. Find the concrete intersection location through linear interpolation. Step 4 only provides information about which edges are intersected by the isosurface for the specific *cubeIndex*, but does not give information about the exact intersection location. Thus linear interpolation has to be exploited to find the exact intersection location along the intersected edge. For example, to find the intersection location along edge 2, as illustrated in Figure 3.8, the following formula is used:

$$l = l_2 + \frac{(l_3 - l_2)(isovalue - s_2)}{(s_3 - s_2)} \qquad (3.2),$$

where l represents the intersection location along edge 2 (isovalue is its scalar value); l_2 and s_2 represent the location and scalar value for voxel 2; l_3 and s_3 represent the location and scalar value for voxel 3.

The same linear interpolation can be used to calculate the other two intersections along edge 3 and edge 11. Consequently the exact intersections that form a triangle mesh can be found.

6. Compute the normal at each intersection for shading triangle meshes. This can be divided into two steps. First, compute the normal, which is also the gradient at each voxel of this cell using central difference. Second, use the same linear interpolation as described in step 5 to compute the normal at each intersection.

7. Go to the next cell and repeat step 1 to step 6, until all cells of a 3D scalar data have been processed. As a result, an approximation of the isosurface using many small triangle meshes are extracted and constructed from the 3D scalar data.

8. A final step is to render these constructed triangle meshes so that the desired isosurface can be displayed in the form of 2D image on screen. This can be easily done by standard graphics techniques with hardware support (Brodlie, 2006)(Papaioannou, 2010).

3.1.2.2 DVR

DVR refers to a set of algorithms or techniques that aim to map an entire 3D scalar data set directly into a 2D image without an intermediate step of isosurface extraction. Thus it is called DVR and Figure 3.4(b) shows its visualization pipeline. In contrast to the isosurface rendering, DVR has two main advantages (Hansen and Johnson, 2004). First, it is capable of capturing the entire characteristics of a 3D scalar data set, including its interior structure that is essentially lost in the isosurface rendering. Second, it could represent certain 3D entities that do not have tangible surfaces and edges, for example, clouds, fog and fire, and thus are not appropriate to be represented by the isosurface rendering.

Volume ray-casting (Drebin, Carpenter and Hanrahan, 1988)(Levoy, 1990a)(Levoy, 1990b) is probably the most classic algorithm to implement DVR. Its basic idea is to cast a ray from every pixel of the final 2D image into a 3D scalar data set, and take samples along every ray within the 3D scalar data set. By compositing all samples along every ray, it forms a final color for every pixel. Consequently the entire 3D scalar data set is directly mapped to a 2D image. In particular, it involves the following 7 steps:

1. Classification. This is the first step where the raw 3D scalar data set is mapped to optical properties, typically color and opacity. Its purpose is to enable users to visually distinguish different scalar values of the 3D scalar data in the final 2D image. This classification task is typically achieved by defining a TF, as illustrated in Figure 3.9, with two objectives: defining which scalar values should share a visual characteristics and how the visual characteristics should appear in the final 2D image. For example, the 3D scalar data set acquired from CT scanners often incorporates a large amount of intensities, with different intensity range maybe corresponding to different materials *i.e.* air, fat, tissue and bone. To clearly distinguish these materials in the final 2D image, a TF must define which intensities belong to, say the bone, and how this bone should appear. Normally, a TF is designed as an interactive interface that users can interactively control how the 3D scalar data should appear in the final 2D image. More precisely, it provides a 1D mapping between a scalar value s and a color $\mathbf{c}$:

$$\mathbf{c} = \mathcal{T}(\mathrm{s}), \mathcal{T}: \mathrm{R} \to R^4 \qquad (3.3),$$

where a color $\mathbf{c}$ refers to a four-component vector consisting of red, green and blue radiance as well as opacity. This is often known as $\mathrm{RGB}\alpha$ (from here and onwards, unless otherwise stated, a color $\mathbf{c}$ refers to the $\mathrm{RGB}\alpha$). By changing the TF, a 3D scalar data set can be rendered as completely different 2D images, as shown in Figure 3.10.

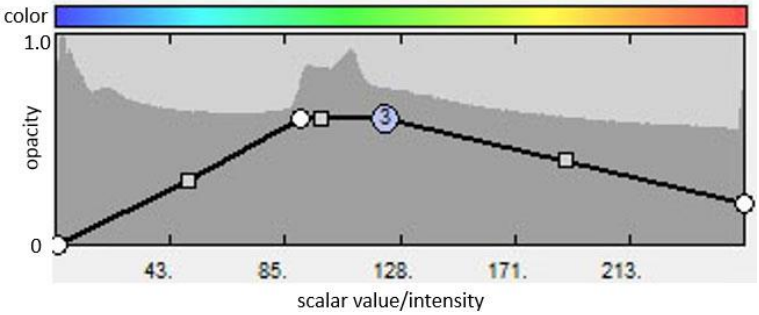


Figure 3.9: A typical TF that maps a scalar value to an opacity plus color, often known as $RGB\alpha$.

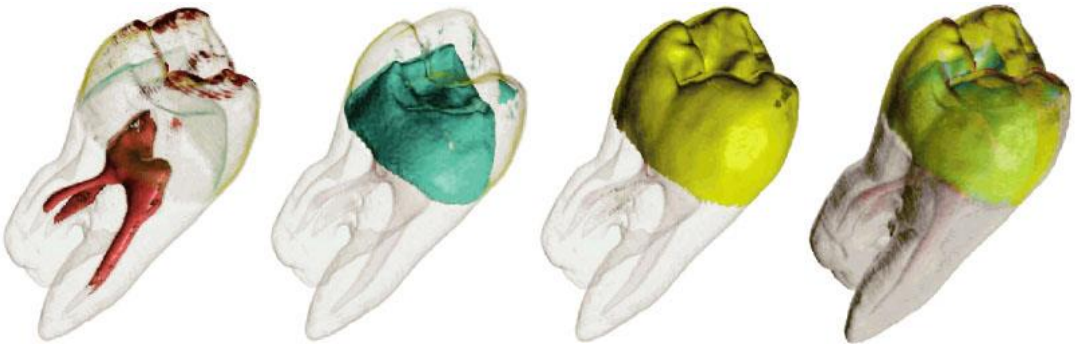


Figure 3.10: Rendering the 3D tooth scalar data (Roettger, 2012) using different TFs (Moller, 2010).

2. Ray casting. Considering a pixel in the final 2D image, cast a ray from this pixel along the observation direction into the 3D scalar data. This process can be shown in Figure 3.11.

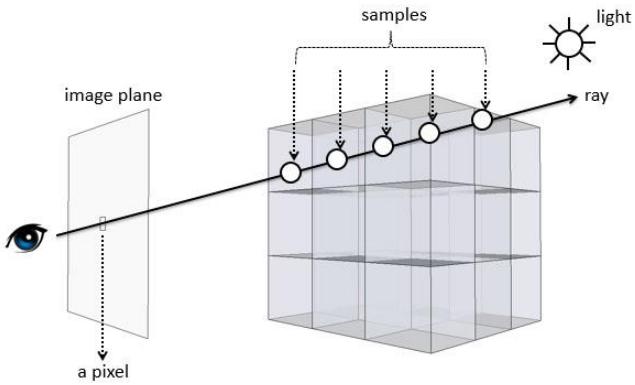


Figure 3.11: A diagram for volume ray-casting.

3. Sampling. Take samples along this ray at regular unit intervals, as illustrated in Figure 3.11. As these samples typically do not coincide with the voxels, linear interpolation is needed to compute their scalar values within their corresponding cells.

4. Shading. Every sample along this ray needs to be shaded accordingly to obtain its correct color $\mathbf{c}'$, which reflects perception of depth, provided that its initial color $\mathbf{c}$ and its normal, as well as the location of the light source are known. The light location is easily known by definition. Every sample's initial color $\mathbf{c}$ can be obtained by looking up a given TF designed in step 1 for classification according to its scalar value, computed in step 3. Its normal can be computed through two steps. First, compute the normal, which is also the gradient, for every voxel of the 3D scalar data set using central difference. Second, compute the normal of every sample using linear interpolation.

5. Compositing. The correct color $\mathbf{c}'$ of every sample along this ray are composited to form the final color value for the pixel that is currently being processed. The composition is typically calculated through the volume rendering integral (Lundstrom, 2007)(Engel et al., 2006), which is derived from the optical model, accounting for emission and absorption, simulating how a ray of light is affected when travelling through the volume. In practice, the composition is typically computed along the ray from front to back (or from back to front) using the discrete form of the volume rendering integral, as shown in formula 3.4:

$$\begin{aligned} C_{out} &= C_{in} + (1-\alpha_{in})\alpha_{now}C_{now} \\ \alpha_{out} &= \alpha_{in} + (1-\alpha_{in})\alpha_{now} \end{aligned} \quad (3.4).$$

   Figure 3.12 illustrates this process, where each dot represents a sample along a ray. C_{now} and α_{now} represent the RGB and opacity values for the sample located in the middle. C_{in} and α_{in} represent the composition's RGB and opacity values before passing through this sample. C_{out} and α_{out} represent the composition's RGB and opacity values after passing through this sample. Using this formula, it iteratively passes through every sample along the ray, from the entry sample to the exit sample, to compute a color value. The final obtained value is the color of the corresponding pixel.

6. Go to next pixel of the final 2D image and repeat step 2 to step 5 until all pixels have been processed. As a result, the final 2D image is constructed.

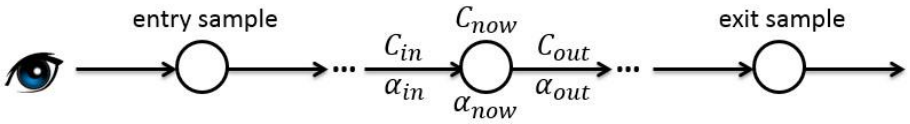


Figure 3.12: Compositing from front to back.

7. A final step is to render and display this 2D image on the screen.

3.2 Framework

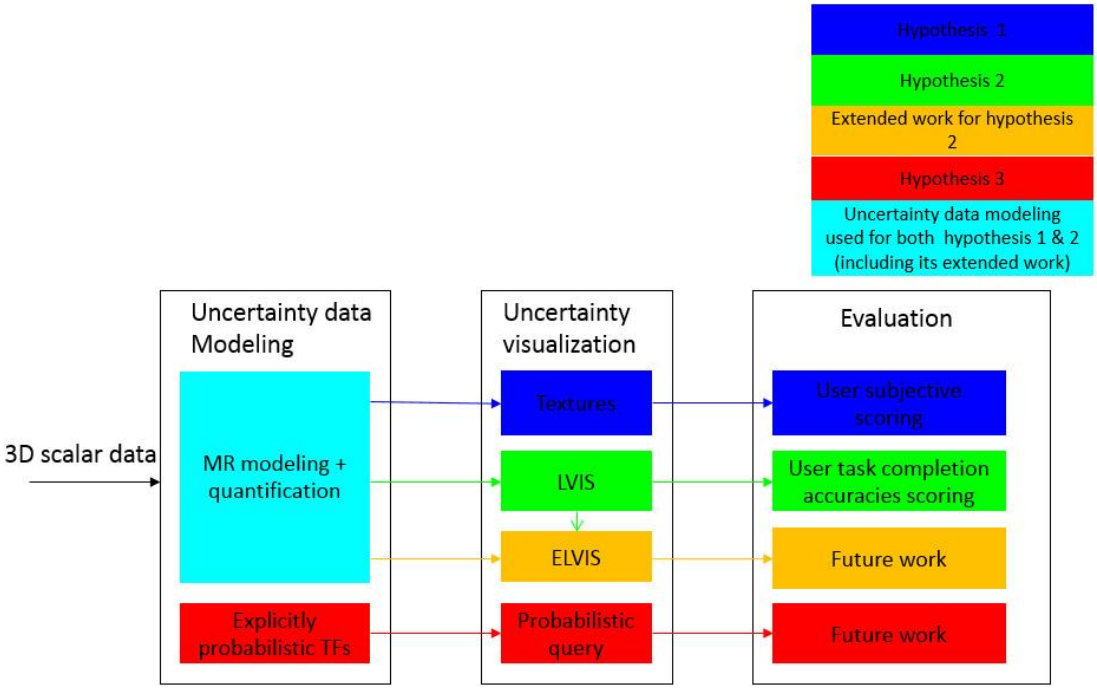


Figure 3.13: The framework of this thesis.

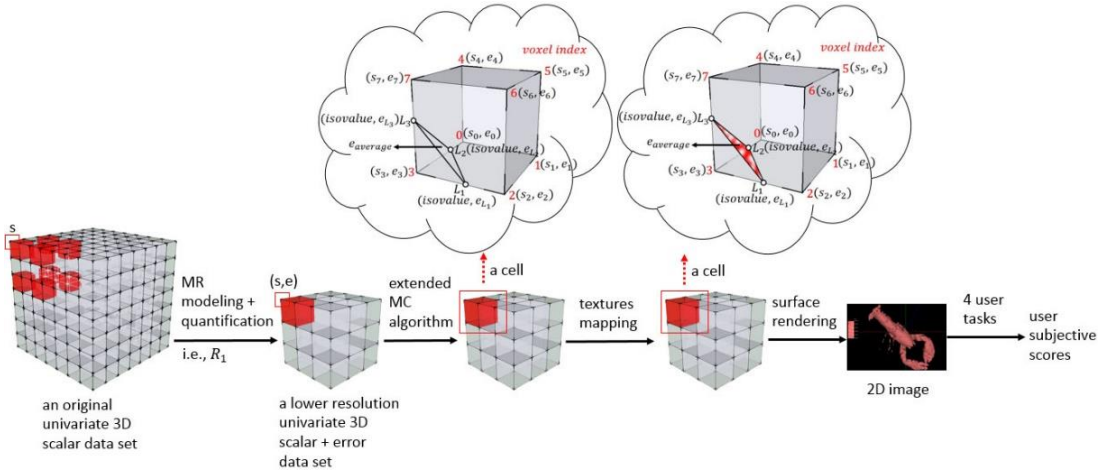


Figure 3.14: Workflow of Texture uncertainty visualization technique.

Figure 3.13 illustrates our proposed framework used in this thesis. This framework consists of three main steps, which are uncertainty data modeling, uncertainty visualization and evaluation, respectively. Every main step implements a specific functionality and includes two or four concrete techniques that are used or developed for different hypotheses to achieve that functionality. In addition, this figure shows how each of the 3 hypotheses that are highlighted in blue, green and red, falls into this framework. Moreover, it shows that there is a technique, which is highlighted in yellow and pointed by the LVIS uncertainty visualization technique. This refers to the extended work for LVIS technique (called ELVIS).

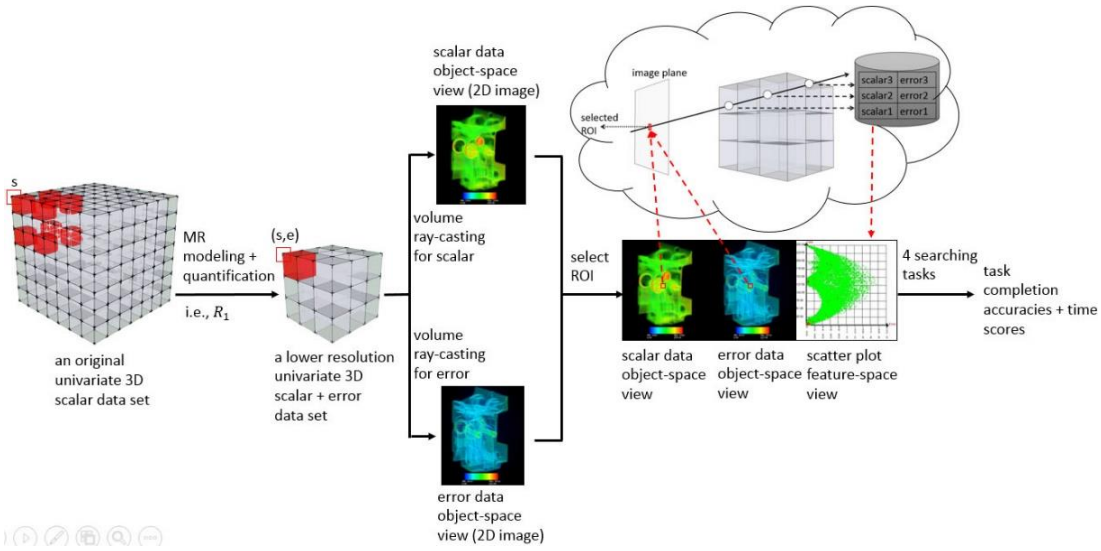


Figure 3.15: Workflow of LVIS uncertainty visualization technique.

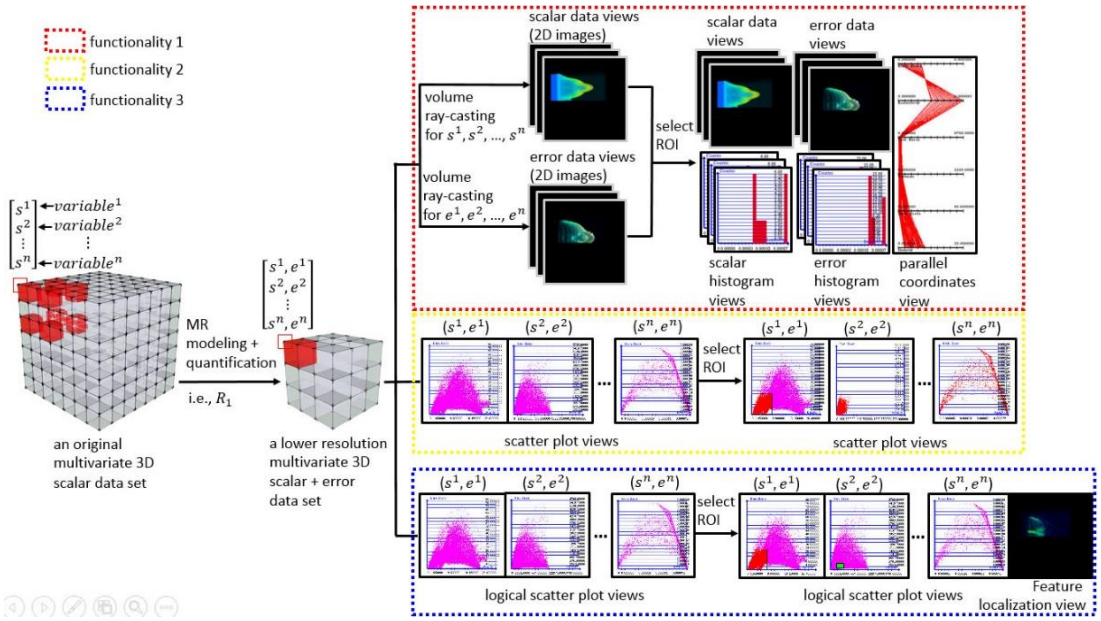


Figure 3.16: Workflow of ELVIS uncertainty visualization technique.

The remaining of this chapter will define the specific functionalities of the three main steps of this framework and explain how they are implemented in each of these concrete works (3 hypotheses + 1 extended work). To give readers a clear overview of each work as well as for ease of later reference, Figure 3.14 to Figure 3.17 illustrates the entire workflow for the Texture uncertainty visualization (hypothesis 1), LVIS uncertainty visualization (hypothesis 2), ELVIS uncertainty visualization (extended work for hypothesis 2) and Probabilistic Query uncertainty visualization (hypothesis 3), respectively.

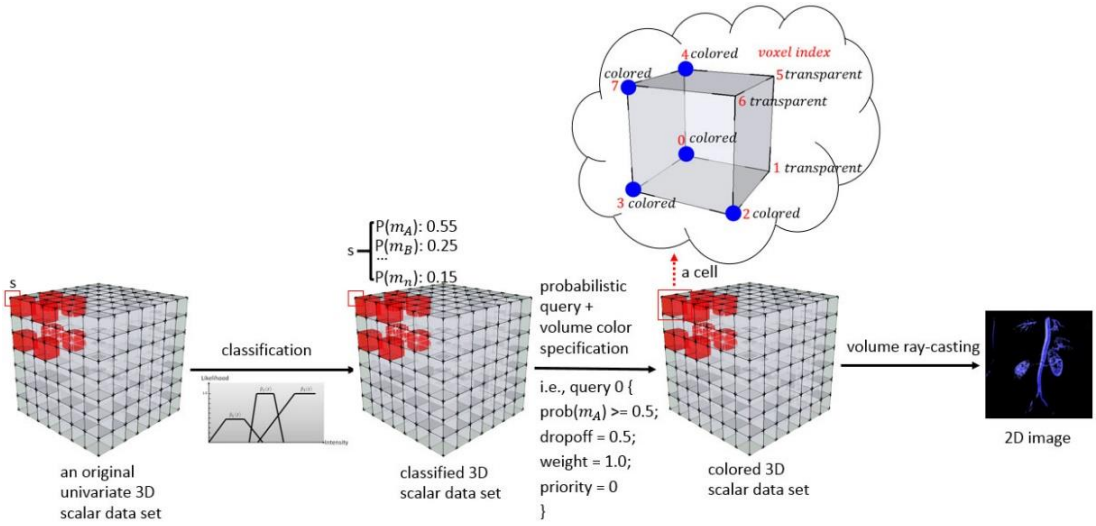


Figure 3.17: Workflow of Probabilistic Query uncertainty visualization technique.

3.2.1 Uncertainty Data Modeling

The uncertainty may refer to various quantities associated with data including error, accuracy, variability, noise, or completeness of data, etc. (Dinesha, Adabala and Natarajan, 2012). An initial issue before trying to explore and investigate uncertainty visualization techniques is to clearly define (Sanyal et al., 2009) and generate a certain type of uncertainty, if uncertainty is not directly available from instrumental measurements or scientific simulations. This is exactly what the framework's first main step – uncertainty data modeling does. It typically takes 3D scalar data as input and computes a certain type of uncertainty data associated to them. As a result, it outputs both 3D scalar data (which may be different from the input one because of some transformations) and their associated uncertainty, which serve as input data for the framework's second main step – uncertainty visualization.

In particular, we can see from Figure 3.13 that two types of uncertainty have been defined and modeled in this thesis:

The first type of uncertainty is defined as errors, which are modeled through a two-step approach and are introduced in Section 4.2 and Section 5.2 in great detail. First, a Multiresolution (MR) modeling technique including either Haar Wavelet Transform or Decimation is applied to an original 3D scalar data set to generate its a set of lower resolution 3D scalar data sets. Second, a quantification step is applied to compute the errors between the original 3D scalar data set and any one (determined by the user) of its lower resolution 3D scalar data sets. These errors are measured at every voxel of the determined lower resolution 3D scalar data set using standard deviation. As a result of this process, a lower resolution 3D data set whose each voxel includes both a scalar value s and the s' associated error values e is generated.

We can see from Figure 3.13 that the Texture, LVIS and ELVIS uncertainty visualization techniques exploit the two-step modeling approach to construct their input data, and Figure 3.14 to Figure 3.16 shows their workflows about how this approach transforms the original data (assuming that the user determined to use the lower resolution data set after one time MR modeling, and we call this data set R_1). We can see from both Figure 3.14 and Figure 3.15 that for both the Texture and LVIS uncertainty visualization techniques, given an original $8 \times 8 \times 8$ univariate 3D scalar data set, whose each voxel includes a scalar value s, as indicated by the red rectangle, the two-step modeling approach transforms it to a $4 \times 4 \times 4$ lower resolution univariate 3D data set, whose each voxel includes both a scalar value s and an error value e associated to s. We can see from Figure 3.16 that for the original $8 \times 8 \times 8$ multivariate 3D scalar data set whose each voxel includes multiple scalar values $s^1, s^2, \cdots, s^n$ that are from n variables, used by the ELVIS uncertainty visualization technique, the two-step modeling approach transforms it to a $4 \times 4 \times 4$ lower resolution multivariate 3D data set, with its each voxel including both scalar values $s^1, s^2, \cdots, s^n$ and their associated error values $e^1, e^2, \cdots, e^n$.

It is worth mentioning that the two-step modeling approach can be deemed as a data transformation process that occurs in the data acquisition stage. Therefore the three

uncertainty visualization techniques including the Texture, LVIS and ELVIS, developed based on it can be classified as visualization of uncertainty, as introduced in Section 1.1.1.1.

The second type of uncertainty is defined as the material intensity classification probability that inherently exists when the 3D scalar data used in the medical imaging domain are classified. This is caused because different materials may share the same intensity. Also, there are variations of the intensity values between different patients (Lundstrom et al., 2007)(Lundstrom, 2007). It is modeled through the explicitly probabilistic TFs, as introduced in Section 6.2, which maps a sampled intensity value s to (1) one or multiple materials, (2) the likelihood associated to each material. As a result of the explicitly probabilistic TFs, the original 3D scalar data set is enriched with more information, with each intensity-specific voxel corresponding to (1) one or multiple materials, (2) the probability (normalized likelihood) associated to each material.

Figure 3.13 shows that the Probabilistic Query uncertainty visualization technique exploits the explicitly probabilistic TFs, and Figure 3.17 shows its workflow about how an explicitly probabilistic TF transforms its original data set. For the original univariate 3D scalar data set with each voxel including a scalar value s used by the Probabilistic Query technique, the explicitly probabilistic TF enriches it to a new 3D data set, called classified 3D scalar data set, with its each intensity-specific voxel corresponding to (1) one or multiple materials such as m_A, m_B, $\cdots$, m_n, (2) the probability $\mathrm{P}(m_A)$, $\mathrm{P}(m_B)$, $\cdots$, $\mathrm{P}(m_n)$ that are associated to these materials.

It is worth mentioning that the intensity classification probability modeled by the explicitly probabilistic TFs is actually occurring in the visualization stage of the visualization pipeline. Therefore the Probabilistic Query uncertainty visualization technique developed based on it can be classified as uncertainty of visualization as introduced in Section 1.1.1.1.

3.2.2 Uncertainty Visualization

This is the framework's second step. It takes the 3D scalar data and their associated error data that are output from the first main step as input and transforms them into

2D images for users' insight. These resulting 2D images can be then used as input for the framework's last main step – evaluation.

In particular, we can see from Figure 3.13 that in total 4 uncertainty visualization techniques have been developed:

The first uncertainty visualization technique is called Texture, which aims to make use of carefully designed textures to depict the degree of uncertainty. More specifically, two types of textures have been designed. The first type of texture is called blurred texture, which is intended to use the blurred degree of textures to convey uncertainty. The second type of texture is called glyph texture with different number of edges, which is intended to use the different number of edges of glyphs being embedded on textures to convey uncertainty. Figure 3.14 shows the specific workflow about how the Texture uncertainty visualization technique transforms its input, which is the lower resolution univariate 3D scalar + error data set, with its each voxel including both a scalar value s and an error value e associated to s, to the final 2D image. We can see from this figure that this process can be divided into 3 steps. First, an extended version of the MC algorithm, as introduced in Section 4.3.1, is applied to the input data set. What it does is that it not only marches each cell and constructs small triangles within a cell based on the cell's 8 voxels' scalar values, but also computes the error value at each vertex of these small triangles based on the cell's 8 voxels' error values. Furthermore, it computes an average error value $e_{average}$ for each small triangle based on the error value at its each vertex. Now that the average error value $e_{average}$ for each small triangle which is part of the isosurface is readily available, the second step – texture mapping, which maps a corresponding texture that is determined by the $e_{average}$ to each small triangle can be conducted. Third, these small triangles with different textures mapped are rendered by standard graphics techniques with hardware support (Brodlie, 2006)(Papaioannou, 2010). As a result, the final 2D image with extracted isosurface is obtained, and the errors of the isosurface are indicated by different textures.

The second uncertainty visualization technique is called LVIS, which aims to enable users to more accurately explore the 3D scalar data set and its associated errors.

Figure 3.15 shows the specific workflow about how it transforms the input, which is the lower resolution univariate 3D scalar + error data set, with its each voxel including both a scalar value s and an error value e associated to s, to final 2D images consisting of 3 visualization components, namely scalar data object-space view, error data object-space view and scatter plot feature-space view, respectively. This process can be summarized as two steps. First, the standard volume ray-casting, as described in Section 3.1.2.2, is applied to both the lower resolution univariate 3D scalar data set and its associated errors, respectively, to obtain two object-space views: scalar data object-space view which represents scalar values and error data object-space which represents error values. The design of separately rendering the scalar and error values is good, as this could give users a clearer data presentation of scalar and error characteristics. Second, based on inspection on the scalar data object-space view and error data object-space view, the user could select a Region of Interest (ROI) on either of these two views and further observe the scalar and error characteristics within the ROI in the scatter plot feature-space view. In this way, a quantitative visualization of the scalar and error values of the input data set can be achieved. One thing that is required to be mentioned here is that the scalar and error values extracted from the ROI are not simply pixel values acquired from scalar/error data object-space views. They are actually corresponding scalar and error sample values extracted from the input 3D data set, as indicated within the cloud of Figure 3.15 (Section 5.3.2.3 will discuss this in greater detail).

The third uncertainty visualization technique is named ELVIS, and it extended the LVIS technique to visualize uncertainty in multivariate 3D scalar data. Figure 3.16 shows the specific workflow of how it transforms the input data set, which is the lower resolution multivariate 3D scalar + error data set, with its each voxel including multiple scalar values s^1, s^2, $\cdots$, s^n and their associated error values e^1, e^2 ,$\cdots$, e^n, into final 2D images that can be grouped into 3 different functionalities, as illustrated by the red, yellow and blue dotted lines in Figure 3.16. The transformation for functionality 1 is derived from the one for LVIS technique, and therefore it can be summarized as two steps. First, the standard volume ray-casting is applied to separately render the lower resolution multivariate 3D scalar values and their

associated error values. As a result, a set of scalar data views and error data views are generated, with each scalar data view representing scalar values of a specific variable, and each error data view representing the error values associated to the scalar values of the variable. The design of separately rendering each variable's scalar values and their associated error values is good, as this could offer users a clearer data presentation. Second, based on inspection on the scalar/error data views, the user could select a ROI on any of them and further observe the statistical information of each variable's scalar values or error values within the ROI through the scalar/error histograms views. Also, the user could observe the correlations between multivariate scalar values and their associated error values within the ROI through the PC view. It is worth mentioning that the multivariate scalar values and their associated error values extracted from the ROI are not simply pixel values acquired from the scalar/error data views. They are actually corresponding multivariate scalar and error sample values extracted from the input 3D data set (Section 5.6.2.4 will introduce it in greater detail). For functionality 2, the transformation can be summarized as two steps. First, the input data set is drawn into n (as there are n variables) scatter plot views, with each one representing a variable's scalar values and their associated error values. Second, based on inspection on these scatter plot views, the user could select a ROI of a variable's scalar values and their associated error values on any one of them, and then observe the corresponding characteristics of other variables' scalar values and their associated error values in other scatter plot views. For functionality 3, the transformation can also be described as two steps. First, the input data set is drawn into n (as there are n variables) logical scatter plot views, with each one representing a variable's scalar values and their associated error values. These logical scatter plot views are different from the scatter plot views mentioned in functionality 2 in that they can be combined as a logical AND or OR operation. Second, based on inspection on these logical scatter plot views, the user can specify one or more ROIs (for one or more variable's scalar values and their associated error values) on any of them and then observe and localize the feature which meets these ROIs in the feature localization view.

The last uncertainty visualization technique developed is named Probabilistic Query, which aims to enable users to better explore the material intensity classification issue involved in DVR and therefore improve DVR as a better decision making tool. Figure 3.17 shows the specific workflow of how it transforms the input classified 3D scalar data set into a final 2D image, and we can see from this figure that this process can be divided into two steps. First, the user formulates either a single or a compound query to specify which voxels he wants to extract and visualize from the classified 3D scalar data set. For example, Figure 3.17 shows a specific single query named $\mathrm{query}\ 0$, which indicates that the user wants to extract and visualize those voxels which (1) include material m_A, (2) the m_A's probability is greater than or equal to 0.5. Additionally, the user specifies RGB colors to different query results (Section 6.3.3 will introduce this in greater detail), and this results in each voxel being assigned with a concrete RGB color. For example, for voxels that meet $\mathrm{query}\ 0$, they will be assigned with the color blue. For voxels that do not meet $\mathrm{query}\ 0$, they will be assigned with the black/background color. As a result of this step, a new data set called colored 3D scalar data set is generated, with each voxel being assigned with a concrete color $\mathrm{RGB}\alpha$, depending on whether or not it meets the user specified query. Second, the volume ray-casting, as described in Section 3.1.2.2, is applied to the colored 3D scalar data set to generate the final 2D image, which corresponds to the user specified query. One thing that requires mentioning here is that during the volume ray-casting rendering, the nearest neighbor interpolation is exploited instead of the most commonly used trilinear interpolation to avoid introducing non-specified intermediate colors, as introduced in Section 6.3.4.

3.2.3 Evaluation

The last main step of this framework is evaluation, which takes the 2D images outputted from the uncertainty visualization as input, and converts them into quantitative scores according to specific user tasks. As a result, the "quality" of uncertainty visualization techniques can be determined. This is a relatively new research topic in the field of uncertainty visualization and is often overlooked by most research (Potter, Rosen and Johnson, 2012).

The effectiveness of a visualization technique to convey meaningful information to HVS is known as its perceptual effectiveness, and it has been exploited by some leading researchers (Sanyal et al., 2009)(Newman and Lee, 2004) to evaluate new or existing uncertainty visualization techniques. Following these works, we also performed two works in this thesis, which evaluated the perceptual effectiveness of the proposed Texture and LVIS uncertainty visualization techniques, as illustrated in Figure 3.13.

The first evaluation work is conducted to determine perceptual effectiveness of the Texture uncertainty visualization technique, and Figure 3.14 shows the specific workflow about how it transforms the 2D image into the final scores. This process involved 4 carefully designed user tasks, which are to ask participants to subjectively rate 4 perceptual aspects (1. Ease of scalar identification, 2. Ease of error identification, 3. Visual overload, 4. Brightness) with respect to uncertainty visualization for the 6 techniques, with each perceptual aspect having a 10-point scale. The final scores about the Texture uncertainty visualization techniques can be directly collected from the participants. We also collected scores for other 4 existing techniques, and Section 4.4 will introduce this evaluation in great detail.

The second evaluation is conducted to determine the perceptual effectiveness of the LVIS uncertainty visualization technique, and Figure 3.15 shows the specific workflow about how it transforms the 2D images to the final scores. This process can be divided into two steps. First, each participant is asked to complete four searching tasks, which are to search a spot, which may include (1) the max. error value, (2) the min. error value, (3) the max. scalar value, (4) the min. scalar value within a randomly generated ROI. Second, depending on the participants' task completion accuracies and time, their task completion results are mapped to the final task completion accuracy scores and time scores. We also computed the task completion accuracy scores and time scores for other 4 existing techniques, and Section 5.4 will introduce this evaluation in greater detail.

3.3 Summary

This chapter introduces this thesis' technical foundations and unified framework, both aims to enable readers to better understanding this thesis.

In the next three chapters, we will present our three works, with each one being used to address one of the three proposed hypotheses, respectively.

Chapter 4 Uncertainty Visualization in Isosurface Rendering Using Textures

4.1 Introduction

This chapter will present our work on the Texture uncertainty visualization technique (hypothesis 1), as illustrated in Figure 3.13.

There are three reasons that motivate us to explore such a technique. First, although making use of texture mapping to depict the degree of uncertainty for MC-based isosurface rendering seems a very straightforward approach, few relevant works have been reported. As described in the literature review, Chapter 2, Rhodes et al. (2003) proposed a technique that utilizes the opacity of a dotted texture to depict the uncertainty for MC-based isosurface rendering, and its results seem very effective. Motivated by this technique, we are determined to explore more texture-related uncertainty visualization techniques for MC-based isosurface rendering. Second, blurring is a natural cue to the eye to depict uncertainty. However, no work has been reported to use it to depict uncertainty for MC-based isosurface rendering. Thus, it may be a good idea to integrate blurred textures into the MC algorithm to convey uncertainty. Third, glyph is a very popular technique to depict different variables in the sub-field of multivariate data visualization (Wong and Bergeron, 1994)(Chan, 2006). Also, they have been successfully applied to the field of uncertainty visualization (Newman and Lee, 2004). However, no work has been reported to use it to depict uncertainty for MC-based isosurface rendering. Thus, it may be a good idea to integrate glyph textures into the MC algorithm to convey uncertainty. Therefore the goal of this chapter is to investigate texture-related uncertainty visualization techniques for MC-based isosurface rendering, which may be effective.

4.2 MR Modeling and Quantification for Univariate 3D Scalar Data

A two-step modeling approach has been exploited to generate the necessary error data (defined in Section 3.2.1) associated to a univariate 3D scalar data that can be used for the Texture uncertainty visualization research. First, a MR modeling technique is applied to an original 3D scalar data set to generate a set of its lower resolution 3D scalar data sets. Second, a quantification step is applied to compute the errors between the original 3D scalar data set and any one (determined by the user) of its lower resolution 3D scalar data sets. Section 4.2.2 and 4.2.3 will describe the two steps in great detail, and Section 4.2.1 will present the motivation to exploit such a modeling approach.

4.2.1 Motivation

There are three reasons that motivate us to exploit the two-step modeling approach.

First, this approach is a concrete instance of data transformation that occurs in the data acquisition stage. Therefore it enables us to conduct relevant research for visualization of uncertainty (refer to Section 1.1.1.1 for explanation).

Second, this approach can generate two important statistical variables: mean μ and standard deviation σ, which can present uncertainty in different ways (as described in Section 1.1.2) and thus enable us to explore different uncertainty visualization techniques. For example, for a singular scalar uncertainty at each voxel, it can be simply depicted by σ. For multivariate uncertainty, it can be depicted by σ^m for each variable m. For bounded uncertainty, it can be depicted by $\mu \pm \sigma$. For a PDF, it can be simplified as a Gaussian distribution and expressed as $N(\mu, \sigma)$.

Third, with the increasing computing and storage capacity of modern computers and instruments, data sets acquired from scientific simulations and real-world measurements are growing in an explosive manner (Ma, 2001)(Wijk, 2005)(Wijk, 2006)(Yang, 2006). Visualization and analysis of such large-scale data sets remain a challenge to the visualization community (Foulks and Bergeron, 2009)(Thompson et

al., 2011)(Brownlee et al., 2012)(Wong et al., 2012). Scientists have tried various techniques (Joy, 2009) to deal with these large data sets. As one of the popular solutions, MR modeling technique can solve the problem by reducing the size of original data and approximating it with a collection of lower resolution data. However, these lower resolution data may raise the data authenticity issue at their resolution level. To address this issue, the two-step approach is applied to provide the necessary data to explore possible visual solutions.

4.2.2 MR Modeling

MR modeling refers to a set of techniques (Cignoni, Rocchini and Scopigno, 1998) (Garland, 1999)(Mallat, 1989)(Daubechies, 1992)(Wong and Bergeron, 1995) (Westermann, 1994)(Guthe et al., 2002) that can reduce an original data set into varying lower resolution data sets (Cignoni et al., 1994). It can help improving the efficiency of data rendering and provide a valuable solution to large-scale data visualization (Joy, 2009). These lower resolution data together with the original data consist of a MR hierarchy in which each member corresponds to a different resolution of the original data (Borgo, Cignoni and Scopigno, 2004).

MR modeling techniques could be applied to reduce both spatial and temporal resolutions of spatial-temporal data (Foulks and Bergeron, 2009). However, as this thesis is mainly focused on exploration of uncertainty in 3D spatial domain rather than time domain, only the spatial component of MR modeling is exploited.

In particular, two specific MR modeling techniques, namely Haar Wavelet Transform and Decimation have been used in this thesis. The reason to choose the two techniques is because while simple, they can be effectively applied to 3D scalar data with regular girds that are used in this thesis, as described in Section 3.1.1, to generate a MR hierarchy with different resolutions.

4.2.2.1 Haar Wavelet Transform

The core of the Haar Wavelet Transform (Li and Drew, 2004) is to compute an average scalar value s of the scalar values $s_1, s_2, \cdots, s_n$ at n neighboring voxels (with each

voxel corresponding to a scalar value) of an original univariate 3D scalar data set, formulated mathematically as:

$$s = \frac{(s_1 + s_2 + \cdots + s_n)}{n} \quad (4.1).$$

It is applied to the original univariate 3D scalar data set piece by piece (piece = n) to generate a set of lower resolution data sets.

A simple example will illustrate this, as shown in Figure 4.1. Figure 4.1(a) shows an original univariate 3D scalar data set whose dimensions are $8 \times 8 \times 8$. For ease of reference, we call the original univariate 3D scalar data set R_0 (where R represents resolution), as its resolution has not been reduced yet. To obtain its next lower resolution univariate 3D scalar data set, the Haar Wavelet Transform has to be applied to it once. We call the next lower resolution univariate 3D scalar data set after one time Haar Wavelet Transform R_1. The process of how the Haar Wavelet Transform transforms R_0 to R_1 can be summarized as follows: (1) for a cell of R_0, say cell 1 as illustrated in Figure 4.1(a), the Haar Wavelet Transform is applied to it to compute a scalar average value s from the 8 scalar values at its 8 voxels. The resulting s forms the scalar value at a corresponding voxel of R_1, as illustrated in Figure 4.1(b). (2) Traverse to another cell of R_0 and repeat (1) until all cells of R_0 have been processed. As a result, R_1 is generated, whose dimensions are $4 \times 4 \times 4$ that is $\frac{1}{8}$ of R_0's size, with the scalar value at its each voxel coming from averaging 8 scalar values at 8 voxels of each corresponding cell of R_0.

In a similar fashion, it is possible to generate R_2 (the lower resolution univariate 3D scalar data set after two times Haar Wavelet Transform, and its dimension are $2 \times 2 \times 2$ that is $\frac{1}{64}$ of R_0's size) from R_0, as illustrated in Figure 4.1(c). However, for the average scalar value s at a voxel of R_2, instead of using 8 scalar values at 8 voxels of a corresponding cell of R_0 to compute it, now it is necessary to use 64 scalar values at 64 voxels of 8 corresponding cells of R_0, as the 8 cells highlighted in red in Figure 4.1(a), to compute it.

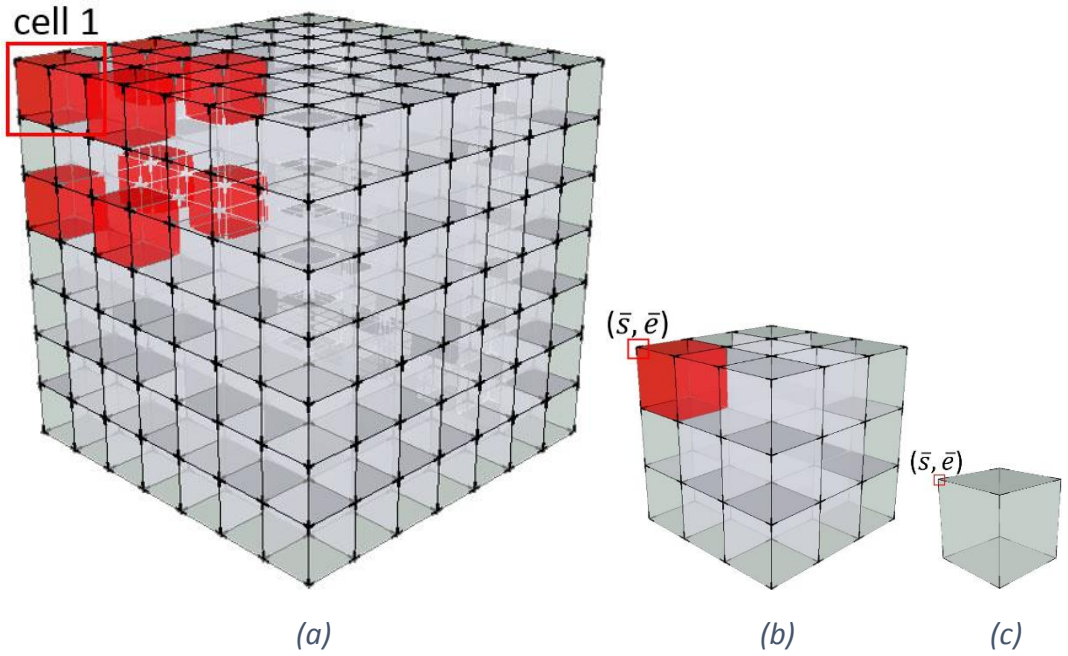


Figure 4.1: An example of a MR hierarchy.

4.2.2.2 Decimation

The core of the Decimation (Foulks and Bergeron, 2009) is to subsample a scalar value s from the scalar values $s_1, s_2, \cdots, s_n$ at n neighboring voxels (with each voxel corresponding to a scalar value) of R_0, and ignore the scalar values at the remaining $\mathrm{n} - 1$ neighboring voxels, formulated mathematically as:

$$\mathrm{s} = subsampling(s_1, s_2, \cdots, s_n) \quad (4.2).$$

It is also applied to R_0 piece by piece (piece = n) to generate its a set of lower resolution data sets, as illustrated in Figure 4.1. The process of how the Decimation transforms R_0 to R_1 can be summarized as follows: (1) for a cell of R_0, say cell 1 as illustrated in Figure 4.1(a), the Decimation is applied to it to subsample a scalar value s from the 8 scalar values at its 8 voxels, and ignore the scalar values at its other 7 voxels. The resulting s forms the scalar value at a corresponding voxel of R_1, as illustrated in Figure 4.1(b). (2) Traverse to another cell of R_0 and repeat (1) until all cells of R_0 have been processed. As a result, R_1 is generated, whose dimensions are $4 \times 4 \times 4$ that is $\frac{1}{8}$ of R_0's size, with the scalar value at its each voxel coming from subsampling the 8 scalar values at 8 voxels of each corresponding cell of R_0.

The similar method can be applied to generate R_2, whose dimension are $2 \times 2 \times 2$ that is $\frac{1}{64}$ of R_0's size, as illustrated in Figure 4.1(c), from R_0. However, for the scalar value s at a voxel of R_2, instead of subsampling 8 scalar values at 8 voxels of a corresponding cell of R_0 to compute it, now it is necessary to subsample 64 scalar values at 64 voxels of the 8 corresponding cells of R_0, as the 8 cells highlighted in red in Figure 4.1(a), to compute it.

4.2.3 Quantification

Given R_0 and any one of its lower resolution 3D scalar data sets, for example, R_1 or R_2, their errors can be quantified piece by piece (piece = n) at each voxel of R_1 or R_2 using the Standard Deviation, formulated mathematically as:

$$\mathrm{e} = \sqrt{\frac{1}{n}[(s_1 - s)^2 + (s_2 - s)^2 + \cdots + (s_n - s)^2]} \qquad (4.3),$$

where e represents a scalar error quantified at a voxel of R_1 or R_2.

For example, based on the previous description we have known the 8 scalar values $s_1, s_2, \cdots, s_8$ at the 8 voxels of cell 1 of R_0, as illustrated in Figure 4.1(a), and we also have known the scalar value s at the voxel that is indicated by the red rectangle of R_1. Therefore the scalar error value e at this voxel of R_1 can be computed by formula 4.3.

As a result of the two-step modeling approach, the lower resolution univariate 3D data sets are formally generated, such as R_1 and R_2 illustrated in Figures 4.1(b) and (c), with their each voxel incorporating both a scalar s and an error e associated to s. They can be now used as the input data for subsequent uncertainty visualization research.

4.3 Texture Uncertainty Visualization Technique

This section corresponds to (Ma et al., 2012a) and (Ma et al., 2012b), and will present the Texture uncertainty visualization technique that includes two types of textures to depict the degree of errors (for the Texture uncertainty visualization technique, we have normalized the errors incorporated within the lower resolution univariate

3D data sets to the range between 0 and 1, and we have divided these errors into five degrees evenly *i.e.* $[0.0,0.2]$ and $(0.2,0.4]$) in isosurface rendering. It includes three subsections. First, the extended MC algorithm, which is used to extract the small triangles that consist of the expected isosurface and map each small triangle with an appropriate texture according to its corresponding error value will be introduced. Second, the two types of textures, namely blurred textures and glyph textures with different number of edges will be presented. Third, a specific case study will be shown to illustrate their application.

4.3.1 Extended Marching Cubes Algorithm

The standard MC algorithm (Lorensen and Cline, 1987) does not take uncertainty into account. It forms the isosurface of a given scalar constant value by traversing every cell of the 3D scalar data, and constructing small triangles within it. To integrate the texture mapping into it for the purpose of uncertainty visualization, it has to be extended. Figure 4.2 illustrates a cell of the 3D scalar data, which is used to explain the extended algorithm.

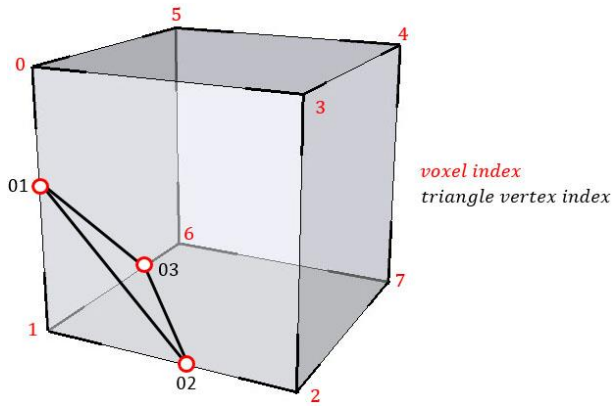


Figure 4.2: A cell of 3D scalar data with 8 voxels distributed at the 8 corners. The three red points represent the vertices of a triangle that is part of the isosurface to be extracted.

We can see from Figure 4.2 that this cell consists of 8 voxels with index $0,1,\cdots,7$ that are distributed in the 8 corners, and the three red points with index $01,02$ and 03 represent the vertices of a triangle that is part of this isosurface and intersects with the cell. Given the scalar value s_i and its associated error value e_i at each voxel i, and also given the isosurface's value $isovalue$, then the process of mapping a texture to

a triangle according to this triangle's average error value $e_{average}$ can be summarized as a three-step approach. First, the error values $e_{01}, e_{02,}\ e_{03}$ associated to the three vertices of the triangle can be computed by three 1D linear interpolations, as shown in formula 4.4:

$$\begin{cases} e_{01} = e_0 + \frac{(isovalue - s_0)(e_1 - e_0)}{(s_1 - s_0)} \\ e_{02} = e_1 + \frac{(isovalue - s_1)(e_2 - e_1)}{(s_2 - s_1)} \\ e_{03} = e_6 + \frac{(isovalue - s_6)(e_1 - e_6)}{(s_1 - s_6)} \end{cases} \quad (4.4).$$

Second, their average error value can be calculated as formula 4.5:

$$e_{average} = \frac{e_{01} + e_{02} + e_{03}}{3} \quad (4.5).$$

Third, according to this average error value $e_{average}$ (normalized between 0 and 1) of this triangle, an appropriate texture could be selected from the pre-processing textures and mapped to this triangle. Alternatively, $e_{average}$ can be used as an input parameter to process a texture in an online fashion, and the resulting texture is then mapped to this triangle. In this research, only the pre-processing fashion is exploited.

The above-mentioned three-step approach should be applied to every triangle of the isosurface. Finally, an isosurface with its errors being indicated by different textures can be acquired.

4.3.2 Two Types of Textures

Two types of textures, namely blurry textures and glyph textures with different number of edges have been designed to depict the degree of errors of an extracted isosurface.

4.3.2.1 Blurred Textures

This type of texture is intended to use the degree of blurring of the textures to indicate the error size, as illustrated in Figure 4.3. In particular, the leftmost image presents the original texture, and the remaining images are those textures after pre-processing of the original texture in Photoshop by using the Gaussian Blur (Patin,

2003)(PixelsTech.net, 2012) with different radius parameters. The larger errors correspond to more blurred textures adjusted by bigger radius parameters, while the fewer ones correspond to less blurred textures adjusted by smaller radius parameters.

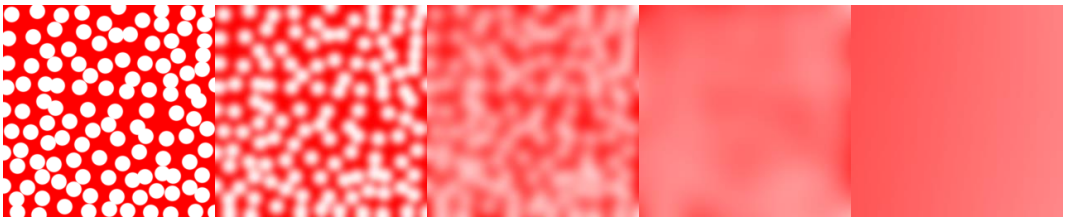


Figure 4.3: Blurred textures used to depict the degree of errors.

4.3.2.2 Glyph Textures with Different Number of Edges

This type of texture is intended to indicate the degree of errors by glyph textures with different number of edges, as illustrated in Figure 4.4. These textures are also generated from Photoshop. The larger errors correspond to the glyph textures with an increased number of edges, while the fewer errors correspond to the glyph textures with a reduced number of edges.

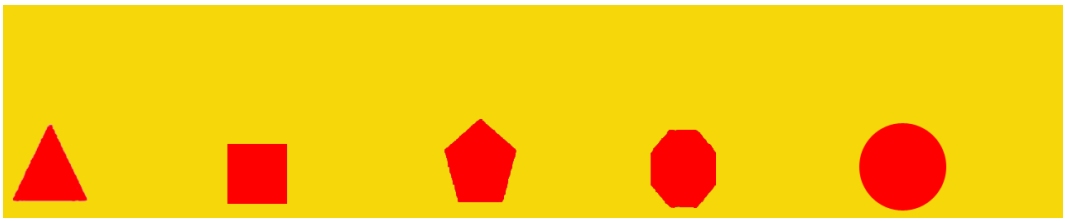
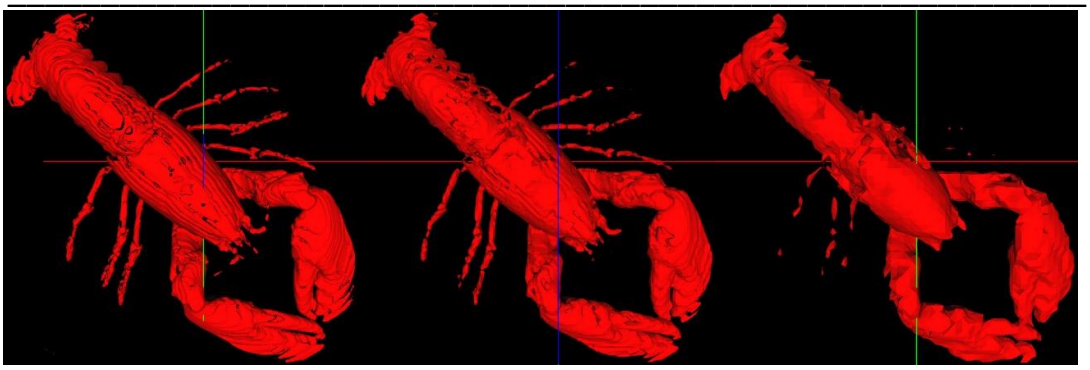


Figure 4.4: Glyph textures with different number of edges.

4.3.3 Case Study

In order to demonstrate the results of the two texture-related uncertainty visualization techniques, a case study is presented. Section 4.3.3.1 introduces the original data set and its pre-processing to generate the test MR data sets. Section 4.3.3.2 presents the corresponding test results.

4.3.3.1 Data Set and Pre-processing



(a) *(b)* *(c)*

Figure 4.5: The MR hierarchy of a CT scan's lobster contained in a block of resin, acquired from the Haar Wavelet Transform. The isosurface value exploited here is 55.

The data set used for this case study is a CT scan of a lobster contained in a block of resin, available at (Roettger, 2012). Its original dimension is $301 \times 324 \times 56$, with 8 bits per voxel. For the convenience of MR modeling, 23 slices are appended to the data set to form a $324 \times 324 \times 56$ derived data set. This derived data set is then processed by two successive MR modeling to generate the final MR data hierarchy with 3 different resolutions. Figures 4.5(a), (b) and (c) show an example of the MR hierarchy generated from the Haar Wavelet Transform. The isosurface value exploited in this case study is 55. We can see that the lobster shown in Figure 4.5(a) has the greatest detail, and it corresponds to the derived data set whose dimensions are $324 \times 324 \times 56$. Compared to the lobster shown in Figure 4.5(a), the one shown in Figure 4.5(b) is slightly coarser *i.e.* the lobster's body tends to be smoother, and its left four legs tend to be more discontinuous, which indicate some detail has been lost. It corresponds to the lower resolution data set after one time Haar Wavelet Transform and its dimensions are $162 \times 162 \times 28$. Figure 4.5(c) shows the coarsest version of the lobster and we can see from this figure that a substantial amount of detail has been lost *i.e.* the lobster tends to be much smoother and its 7 legs have almost disappeared. It corresponds to the lower resolution data set after two times Haar Wavelet Transform and its dimensions are $81 \times 81 \times 14$. For easy reference of these MR data sets in the subsequent paragraphs, we refer to the $324 \times 324 \times 56$ data set as R_0, the $162 \times 162 \times 28$ data set as R_1, and the $81 \times 81 \times 14$ data set as R_2.

4.3.3.2 Results and Discussion

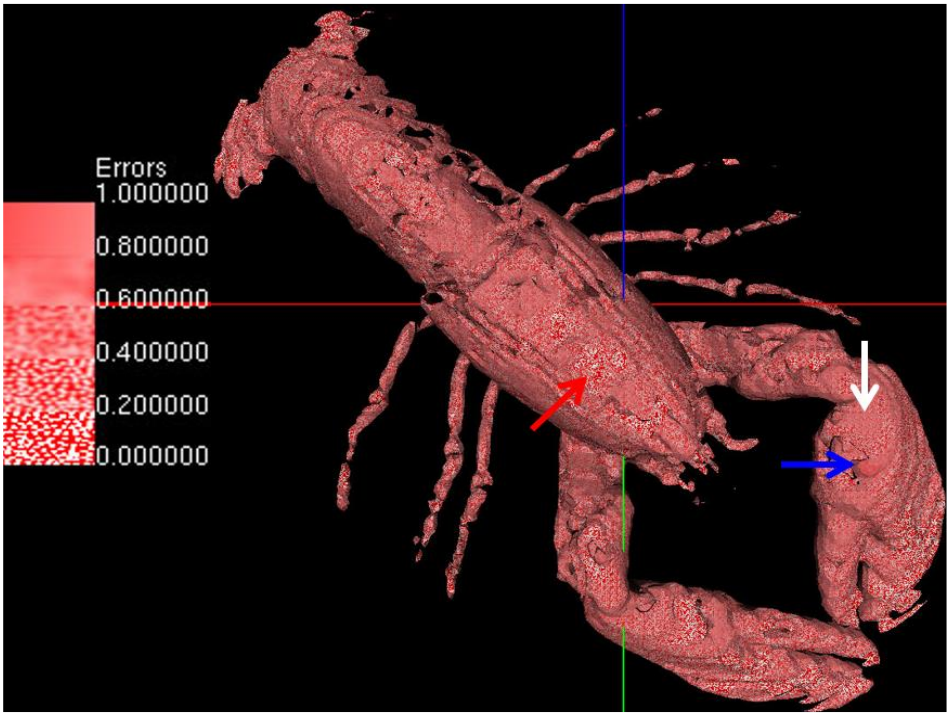
Now that the MR data sets are readily available, we can apply the blurred textures and the glyph textures with different number of edges, respectively to these data sets to inspect the corresponding results.

Figures 4.6(a) and (b) show the results with respect to the blurred textures. The data set used here is R_1. We can see from the scale bar in Figure 4.6(a) that the error data have been normalized between 0 and 1, and divided into five levels. The lowest level of error values are between 0 and 0.2, and corresponds to the texture without any blurring. The highest level of error values are between 0.8 and 1, and it corresponds to the most blurred texture. Other levels of error values are in between, and they correspond to the textures, which are more blurred than the one in the lowest level, and less blurred than the one in the highest level.

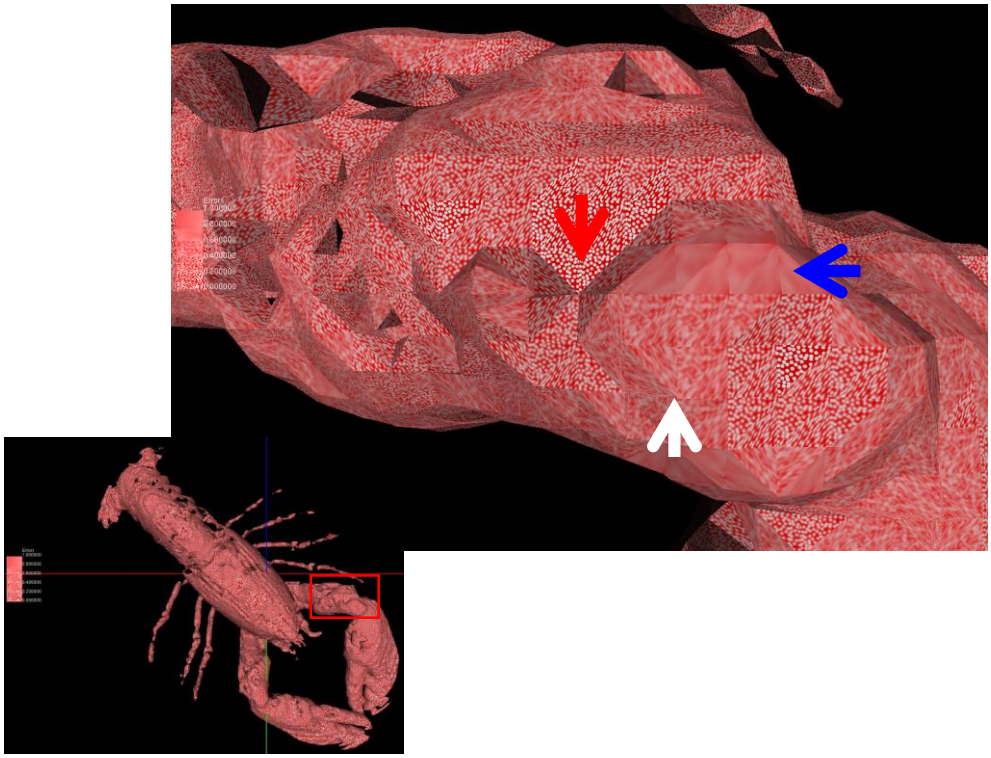
Figure 4.6(a) shows a global view of the lobster, and we can see by observing the blurred textures that different positions of the lobster have different levels of errors. For example, the textures indicated by the red arrow are very clear, and they thus represent the positions where the errors are relatively small. The textures indicated by the blue arrow are very blurred, and they thus represent the positions where the errors are relatively big. The textures indicated by the white arrow are not very clear, but not very blurred too, and they thus represent the positions where the errors are larger than the ones indicated by the red arrow, but fewer than the ones indicated by the blue arrow.

To provide an even clearer perception of the errors using the blurred textures, a close-up of the lobster's left arm region (indicated by the red rectangle in the bottom left image of Figure 4.6(b)) is presented, as illustrated in the top right image of Figure 4.6(b). Again, the textures denoted by the red arrow appear very clearly, and they thus represent the positions where the errors are relatively small. The textures denoted by the blue arrow appear very blurred, and they thus represent the positions where the errors are relatively big. The textures denoted by the white arrow appear slightly blurred, and they thus represent the positions where the errors are larger

than the ones indicated by the red arrow, but fewer than the ones indicated by the blue arrow.



(a)



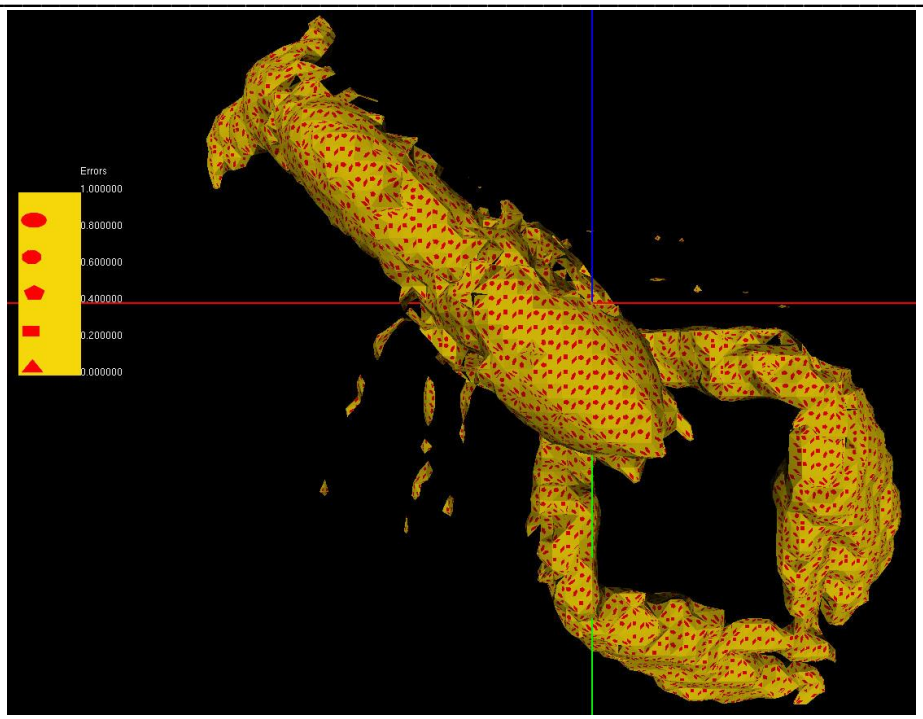
(b)

Figure 4.6: The results by using the blurred textures technique.

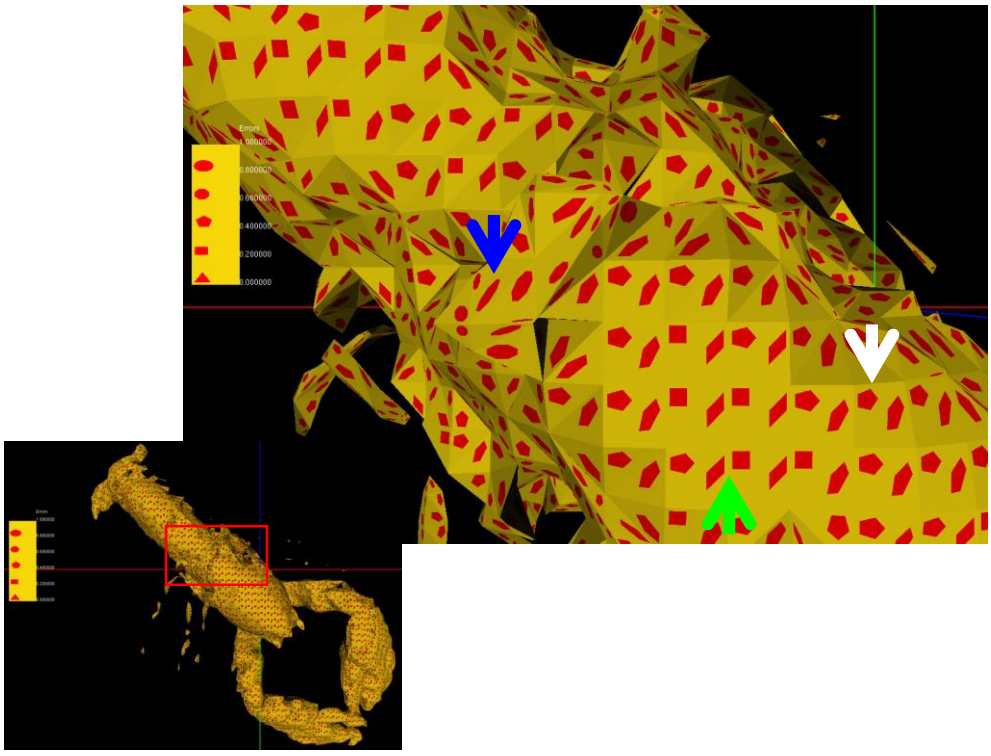
Figures 4.7(a) and (b) show the results with respect to the glyph textures with different number of edges, and the data set used here is R_2. It can be seen from the scale bars in both figures that the error data have been normalized between 0 and 1, and divided into five levels. The lowest level of error values are between 0 and 0.2, and it corresponds to the glyph texture with 3 edges. The highest level of error values are between 0.8 and 1, and it corresponds to glyph textures with infinite edges (a circle). Other levels of error values are in between, and they correspond to the textures whose number of edges are 4, 5 and 8, respectively.

Figure 4.7(a) shows a global view of the lobster by using the glyph textures technique with different number of edges. At a glance, it is not difficult to obtain the impression that there are barely textures that have 3 edges. This indicates that most positions incorporate errors that are greater than or equal to 0.2. However, it is very difficult to utilize this technique to quickly detect the local error pattern, in contrast to the blurred textures technique, as its local textures do not combine together to form a continuous clustering pattern for perception. In addition, the overall impression of the lobster rendered by this technique displays a little bit visual overload, and this case would be even worse if an isosurface with more triangles is extracted from a larger data set.

To obtain a better result, a close-up of the lobster's body region (as indicated by the bottom left image of Figure 4.7(b)) is presented, as illustrated by the top right image of Figure 4.7(b). From this figure we can see that the region indicated by the green arrow consists of textures with 4 edges, and it thus incorporates relatively small errors. The region indicated by the blue arrow consists of textures with 8 edges, and it thus incorporates relative big errors. The region indicated by the white arrow consists of textures with 5 edges, and it thus has errors that are larger than the ones indicated by the green arrow, but fewer than the ones indicated by the blue arrow. Although it is possible to use this technique to distinguish the degree of errors after zooming in to a ROI, it still suffers from these two major drawbacks: (1) it cannot form clustering pattern for easy perception of the degree of errors, (2) visual overload.



(a)



(b)

Figure 4.7: The results of using the glyph textures with different number of edges.

4.4 Evaluation of Perceptual Effectiveness

Many uncertainty visualization techniques have been proposed to visualize uncertainty in isosurface rendering, and some techniques seem to appear more effective than others (Sanyal et al., 2009). However, little research has been reported to evaluate their perceptual effectiveness. This has been identified as an often-overlooked aspect in the field of uncertainty visualization (Potter, Rosen and Johnson, 2012). Therefore we conducted an evaluation to compare the perceptual effectiveness of the Texture technique with 4 other commonly used isosurface rendering-based uncertainty visualization techniques. Such an evaluation corresponds to (Ma et al., 2013a) and has double significance. First, it could help us to validate the effectiveness of the proposed Texture uncertainty visualization technique. Second, it could discover advantages and disadvantages of those existing uncertainty visualization techniques. The resulting findings from the evaluation will provide useful guidance for future uncertainty visualization design.

As described in the literature review of Chapter 2, Newman and Lee (2004) presented research that evaluated the perceptual effectiveness of 8 isosurface rendering-based uncertainty visualization techniques. In particular, seven of these techniques can be classified as the overloading approach, and one can be classified as the seamless integration approach (Pang, Wittenbrink and Lodha, 1997). In contrast to their research, all the techniques described in this evaluation belong to the seamless integration approach.

4.4.1 Study Design

A user study has been designed to compare the perceptual effectiveness (Sanyal et al., 2009)(Alexandre and Tavares, 2010)(Newman and Lee, 2004) of the two types of Textures uncertainty visualization techniques with other 4 commonly used uncertainty visualization techniques. This includes the method of uncertainty data modeling, chosen uncertainty visualization techniques, participant pool and user study tasks, interface design, participant training, the trial run and finally the main study.

4.4.1.1 Uncertainty Data Modeling

The uncertainty data modeling method exploited in this user study strictly follows the two-step modeling approach, as described in Section 4.2. It is applied to both the lobster (used in the main study, introduced in Section 4.4.1.8) and the engine (used in the participant training, introduced in Section 4.4.1.6) univariate 3D scalar data sets (Roettger, 2012) to generate the input data for the 6 evaluated uncertainty visualization techniques, as will be described in the following section.

4.4.1.2 Six Uncertainty Visualization Techniques Chosen for Evaluation

The six uncertainty visualization techniques chosen for this evaluation are: hue, blurred textures, glyph textures with different number of edges, transparency, transparency with enhanced grid background, and texture opacity. They are all seamlessly integrated (Newman and Lee, 2004) into the extended MC algorithm, as described in Section 4.3.1.

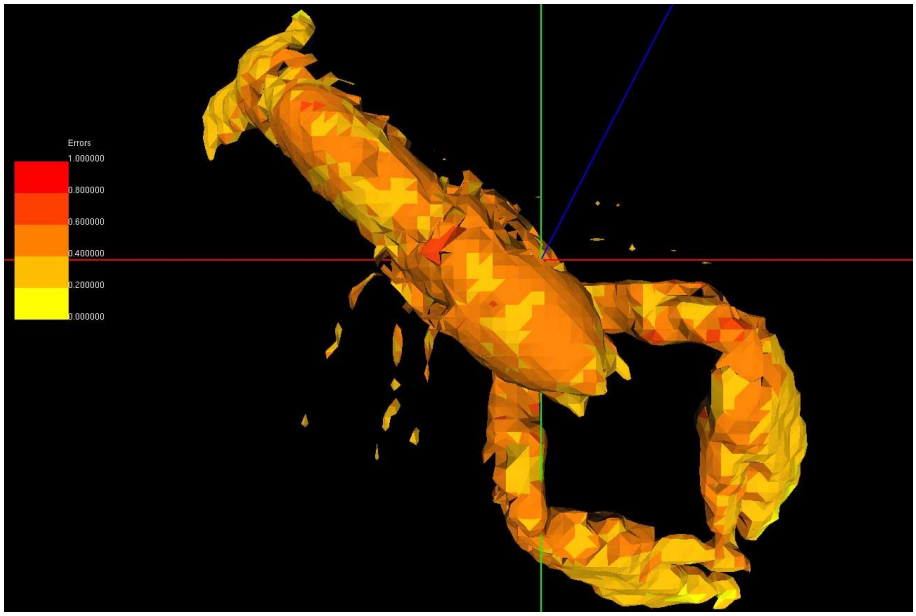


Figure 4.8: Errors are mapped to the hue.

Hue: this is a known technique proposed by Rhodes et al. (2003), and it uses the hue component of HSL to indicate the presence and degree of errors. Figure 4.8 illustrates a result of this technique after applying it to a 3D scalar data set of a lobster (Roettger, 2012), which has been used in our main study. We can see from this figure that the errors have been mapped to five discrete hue values. The fewer errors correspond to the hue values that appear more yellow, and the larger errors correspond to the

hue values that appear more red. Other errors correspond to the hue values with colors in-between.

Blurred textures: This is the proposed technique as described in Section 4.3.2.1. It indicates the presence and degree of errors by the blurred textures. Although using blur to depict uncertainty is not a new concept (Pang, Wittenbrink and Lodha, 1997)(Grigoryan and Rheignans, 2004), to our knowledge, integrating the blurred textures into the MC to depict the errors has not been explored. Figure 4.9 illustrates a result of this technique. The blurred effects used here are from Gaussian Blur. We can see from this figure that the errors have been mapped to five discrete blurred textures. The fewer errors correspond to less blurred textures, and the larger errors correspond to more blurred textures. Other errors correspond to the blurred textures in-between.

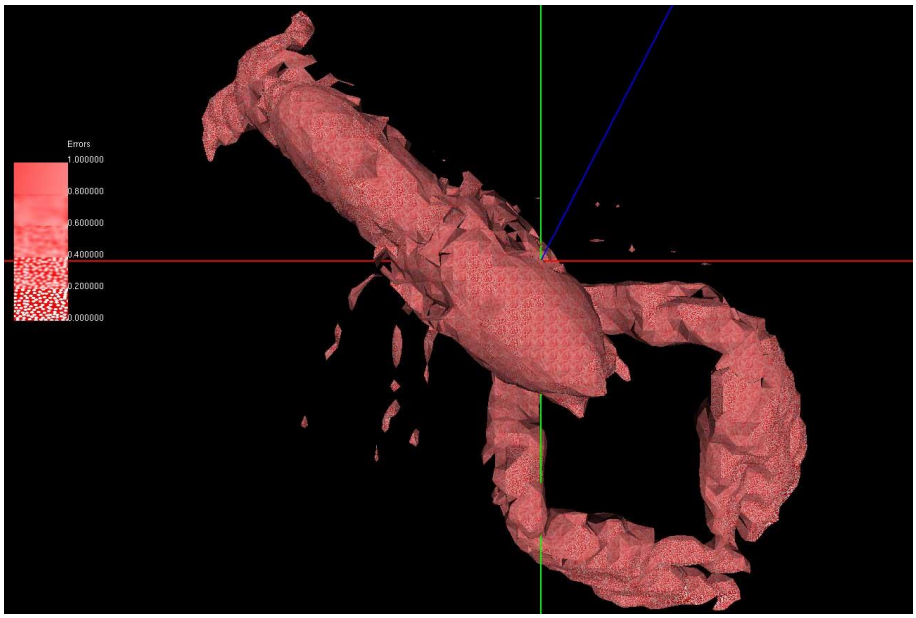


Figure 4.9: Errors are mapped to the blurred textures.

Glyph textures with different number of edges: This is the proposed technique as described in Section 4.3.2.2. It indicates the presence and degree of errors by different edge numbers of the glyph textures. Although using glyphs to depict uncertainty is not a new concept (Pang, Wittenbrink and Lodha, 1997)(Newman and Lee, 2004), to our knowledge, integrating the glyph textures with different edge numbers into the MC to depict the errors has not been explored. Figure 4.10

illustrates a result of this technique. We can see from this figure that the errors have been mapped to five discrete glyph textures, with each one being associated to a glyph with different number of edges. The fewer errors correspond to the glyph textures with fewer edges, while the larger errors correspond to the glyph textures with more edges.

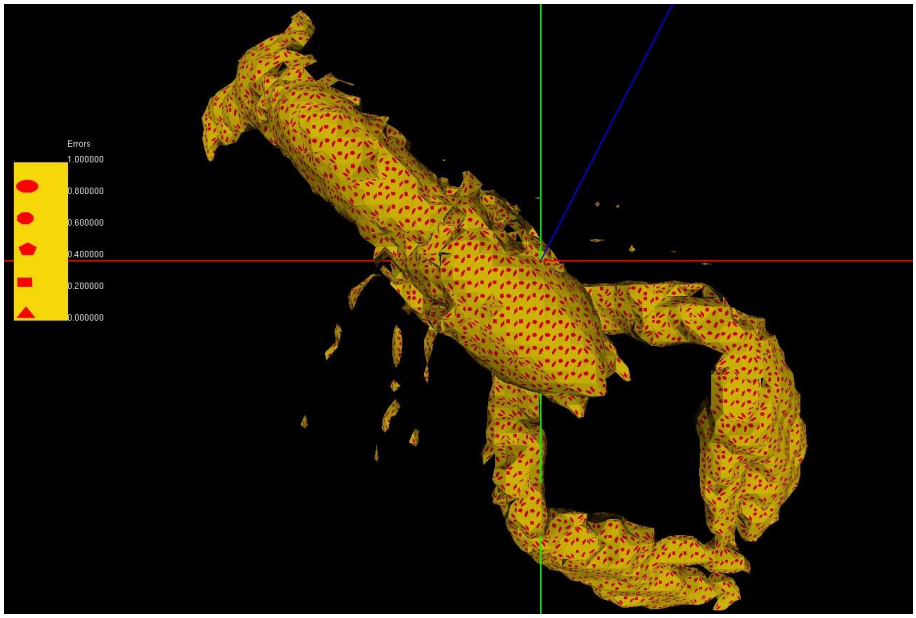


Figure 4.10: Errors are mapped to the glyph textures with different number of edges.

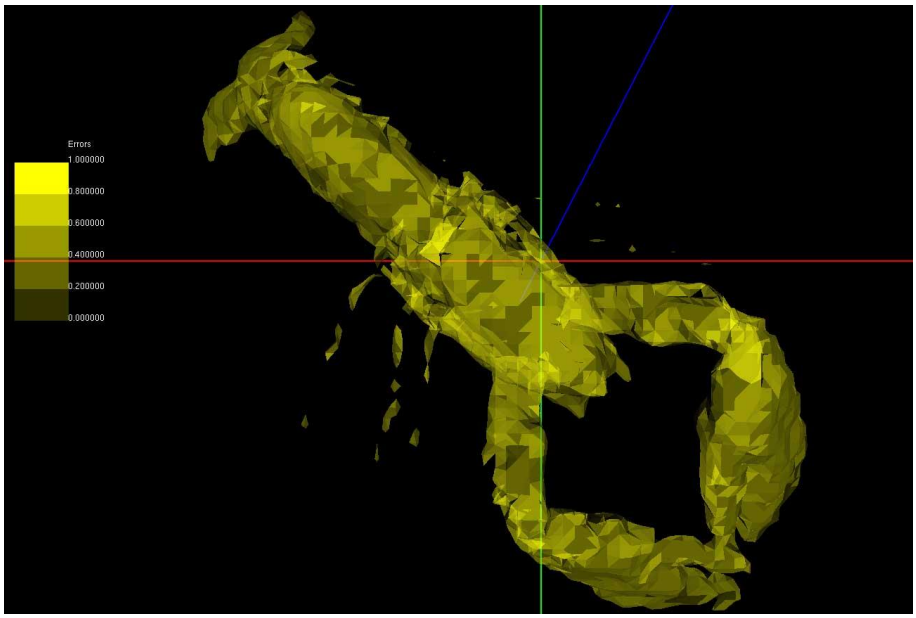


Figure 4.11: Errors are mapped to the transparency.

Transparency: this is a commonly used uncertainty visualization technique. It indicates the presence and degree of errors by transparency of the isosurface. Figure

4.11 illustrates a result of the technique. We can see from this figure that the errors have been mapped to five discrete scales of transparency. The fewer errors are mapped to more transparent, and the larger errors are mapped to less transparent. Other errors are mapped to transparency in-between.

Transparency with enhanced grid background: this technique is similar to the technique as described in the above except that extra grid lines are added as the background. We want to test whether such a cue is helpful for users to distinguish the errors depicted by the transparency, as suggested in (Djurcilov et al., 2001)(Djurcilov et al., 2002). Figure 4.12 illustrates a result of this technique.

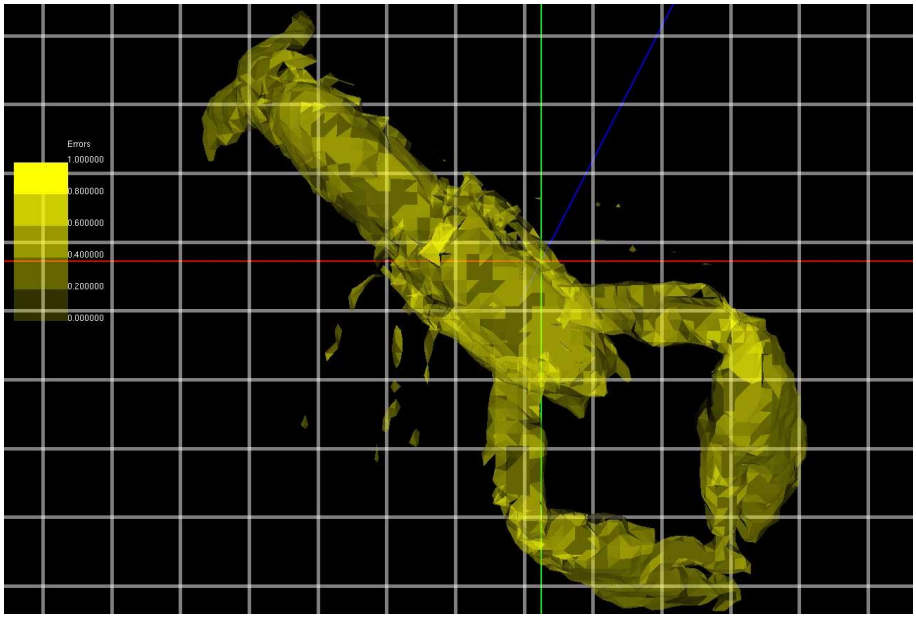


Figure 4.12: Errors are mapped to the transparency with enhanced grid background.

Texture opacity: this is a known technique proposed by Rhodes et al. (2003). It indicates the presence and degree of errors by the texture opacity. Figure 4.13 illustrates a result of this technique. We can see from this figure that the errors have been mapped to five discrete opacities of a texture. The fewer errors correspond to the less opaque textures, while the larger errors correspond to the more opaque textures.

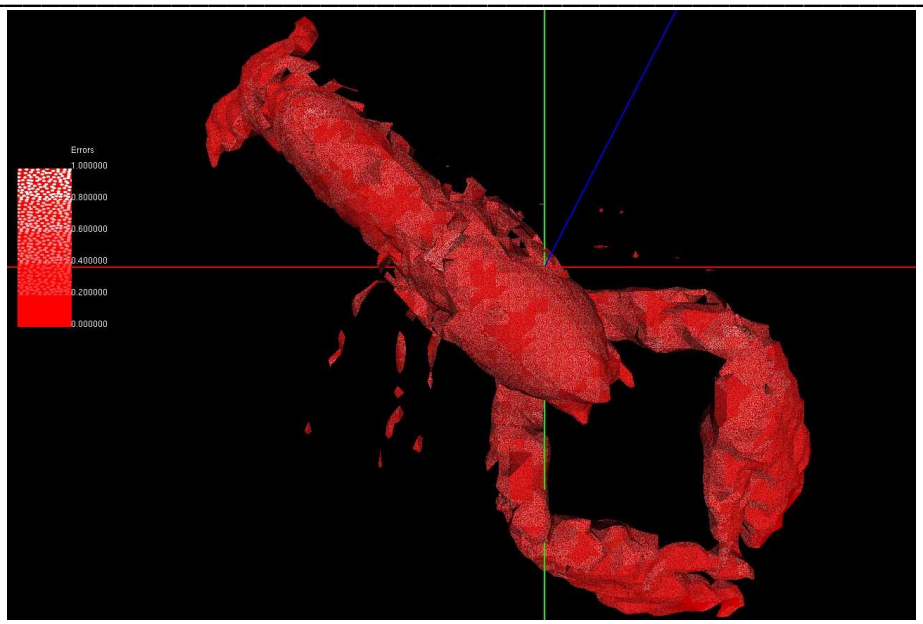


Figure 4.13: Errors are mapped to the texture opacity.

4.4.1.3 Participant Pool

In total we had 30 participants, of which 2 participated in a trial run, and the remaining 28 participated in the main user study. More specifically, there are 8 participants who are researchers in University College Cork (UCC) and there are 4 participants from our industrial partner. The remaining participants are postgraduate students in UCC. Among these participants, 10 are female and 20 are male. None of them are color-blind. All of them claimed that they use graphs and charts for day-to-day activities and they typically use a computer more than 21 hours per week.

4.4.1.4 User Study Tasks

Different methods have been proposed for usability and effectiveness studies. Some (Newman and Lee, 2004)(Rickenberg and Reeves, 2000) employed subjective user ratings of effectiveness for their evaluation. Some (Sanyal et al., 2009) used task completion time and accuracies as their evaluation. In this work, we employed the former and each participant is asked to rate four perceptual aspects for each uncertainty visualization technique (with 10-point scale): (1) how easy it is to identify the data? (2) How easy it is to identify the error? (3) Do you feel visual overload? (4) How do you feel about the brightness? No weighting of the relative importance of these aspects is considered here. In addition, each participant is asked to answer an

open question about his impression to the technique. Some basic personal information *i.e.* do you use charts/graphs for day-to-day activities is also collected from each participant.

4.4.1.5 Interface Design

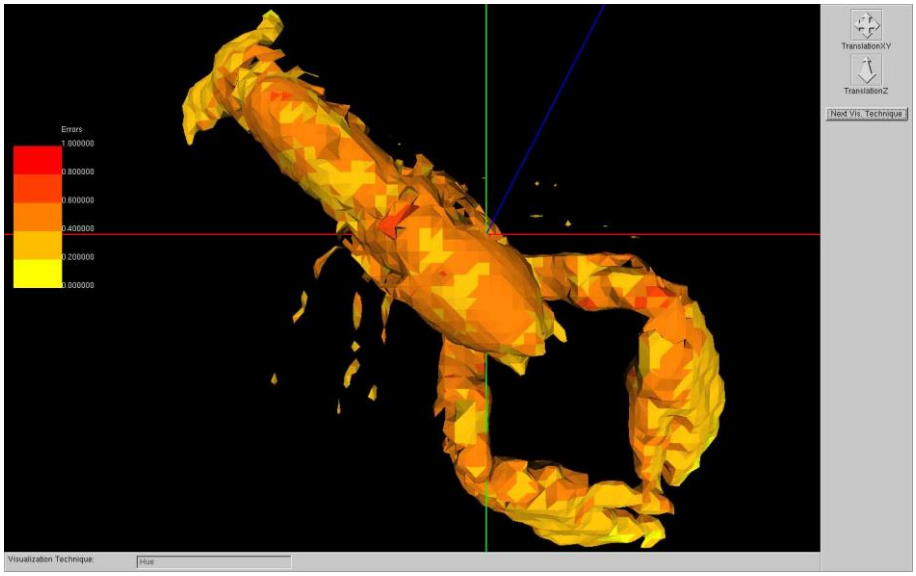


Figure 4.14: The user interface.

Keeping a real-world scenario in mind, scientists like to look at the entire data set first and then focus on a ROI (Sanyal et al., 2009), a simple interface is designed to facilitate the participants' evaluation. Figure 4.14 shows this interface, which consists of three parts. The first part is the 3D display area located in the middle. It is intended to display both the isosurface rendering and the scale bar that are associated to one uncertainty visualization technique. Please note that only one uncertainty visualization technique can be displayed in this area at a time, and it appears on the screen in a random order. The second part is the interaction area located on the right. It facilitates the participants' exploration and observation to the data by the two functionalities of translation and zooming in or out. Consequently the participants can easily navigate to their ROI. Once they completed the evaluation to an uncertainty visualization technique, they can go to the next technique by clicking the "Next Vis. Technique" button. The last part of this interface is a status bar located at the bottom, and it informs the name of currently displayed technique to the

participants. This interface is configured under the full screen mode to avoid unexpected user behaviour of untimely termination.

4.4.1.6 Participant Training

A training process is given to the participants before the main user study. This is to make sure that they are familiar with the evaluation and felt confident to take on the main user study. We typically spent about 5 to 10 minutes giving participants an overview of the user study. This involved getting them to understand the concept of uncertainty visualization, its significance, as well as the purpose of the user study. After this, each participant is assigned to a computer that ran a training module that is similar to the main study, but used a different data set, which is a CT scan of an engine (Roettger, 2012). The participant is asked to complete the corresponding tasks in the training module. Coupling with this process, we spent about 10 to 15 minutes explaining the participants the 6 different uncertainty visualization techniques, the user interface as well as the four user rating tasks expected from them. We believed that organizing the explanation along with the training module is a better idea than those methods that separate them, because this could better the participants' understanding with living examples. After the participants completed the training, we entered the "question and answer" stage where we ansared all questions that are unclear to them. Finally they went to the main study where they performed all tasks independently.

4.4.1.7 The Trial Run

A trial run has been conducted to identify the weakness of the user interface. Two participants have taken part in this procedure. One is a postgraduate student who has a good understanding of uncertainty visualization. Another one is a senior researcher in UCC who has rich experience in user study design. Based on the trial run we have identified a weakness of the user interface: initially, our interface included a rotation functionality that aimed to assist the participants in gaining a more comprehensive observation. While this idea is good for data and error identifications, it dramatically increases the time that the participants spent on the

evaluation. This may have resulted in the participants losing their motivation and giving unreliable rating results. Therefore we removed the rotation functionality from the final interface.

4.4.1.8 The Main Study

The data set we used in the main user study is a CT scan of a lobster (Roettger, 2012), as illustrated in Figure 4.14. We had a total of 28 participants who took part in the main user study and we only selected feedback from 25 of them for the analysis. We excluded the feedback from the other 3 participants for three reasons, either they seemed unmotivated, completed the tasks in a rush, or their rating answers are inconsistent with their answers for the open question. In order to avoid interplay between participants, we ran the user study one participant at a time. We kept all participants in a similar environment to eliminate impact from the environment.

4.4.2 Initial Findings

The initial findings of the average scores of the 4 aspects of perceptual effectiveness with respect to the 6 uncertainty visualization techniques are reported in this section.

4.4.2.1 Data Identification

Figure 4.15 presents the average scores of the data identification rated by the participants for the six uncertainty visualization techniques. We can see that both existing hue and texture opacity techniques have the highest average scores (above 9 points), which indicates they are the best practice among the six techniques to identify the data. On the contrary, the two transparency-related techniques have the lowest average scores (below 7 points), which indicates they are difficult for the participants to identify the data. However, both their average scores are more than 5 points, which means it is still possible to use them to identify the data. As for the other two techniques, they had medium average scores (between 8 and 9 points), which indicated ease in identifying the data. We attribute the reasons for the low scores of these two transparency-related techniques to the partial missing of the data caused by transparency or semi-transparent rendering.

4.4.2.2 Error Identification

Figure 4.16 illustrates the average scores of the error identification for these techniques. It is obvious that the hue technique has the highest score (nearly 9 points), which indicates it is the best method to identify errors. On the contrary, the glyph textures with different number of edges technique has the lowest average score (below 2 points), which indicates its uselessness to identify errors. As for the remaining four techniques, whilst it is possible to employ the texture opacity and both transparency-related techniques to identify errors, it is relatively difficult to use blurred textures to identify errors. In addition, one surprising finding from the study is that adding auxiliary grid lines, as background, did not guarantee any enhancement to the participants' perception of the errors depicted by the transparency. Conversely, it appeared to lead to visual overload that increased the difficulty to recognize data, as described in the following section.

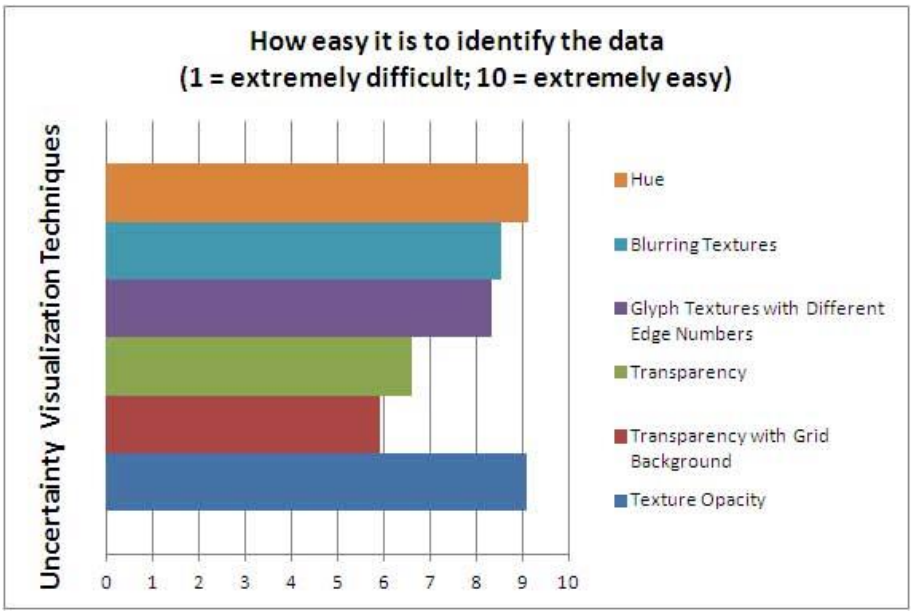


Figure 4.15: Average scores of the 6 uncertainty visualization techniques on the data identification aspect.

We can see from Figure 4.16 that the mean of the glyph textures with different number of edges technique is dramatically lower than any other techniques. We believe the reason is because our visual system is more sensitive to clustering of patterns (Tao et al., 2004) rather than non-clustering ones, and this technique failed

to provide us with the clustering patterns. In addition, distortion occurred during the process of texturing, which may cause perceptual difficulties to identify the number of edges of the textures. Also, we can see from this figure that the mean of the blurred textures technique is relative low (below 5.0 points). This indicates that blur is not a good metaphor to depict uncertainty.

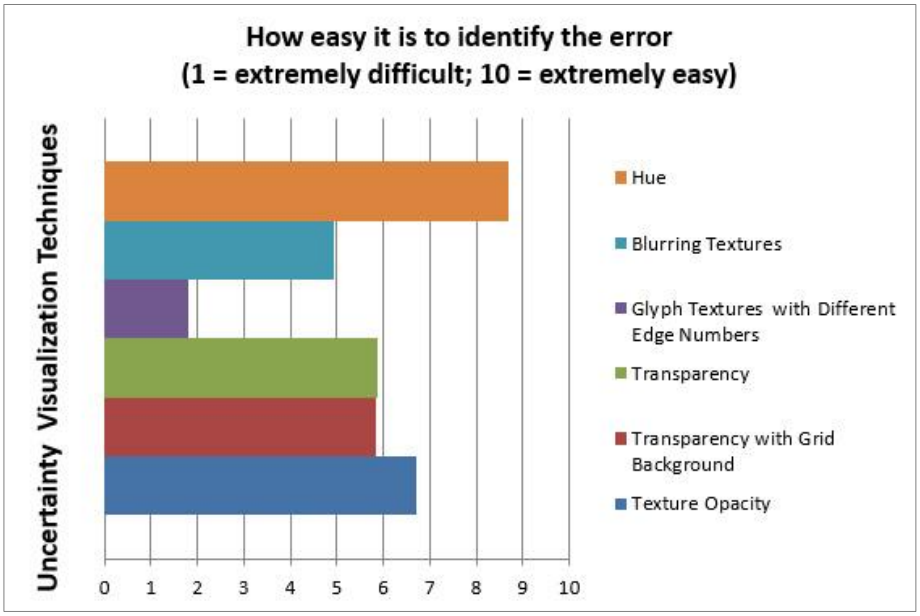


Figure 4.16: Average scores of the 6 uncertainty visualization techniques on the error identification aspect.

4.4.2.3 Visual Overload

Figure 4.17 illustrates the average scores of the visual overload for these techniques. We can see that both of the glyph textures with different number of edges and the transparency with enhanced grid background techniques have the highest average scores (slightly above 6 points), which indicates their difficulties for the participants to make a clear observation. The hue technique has the lowest average score (below 2 points), which indicates its ease for the participants to make an observation. The remaining three techniques have average scores more than 2.5 and less than 4.5 points, which reveals their usefulness for a clear observation.

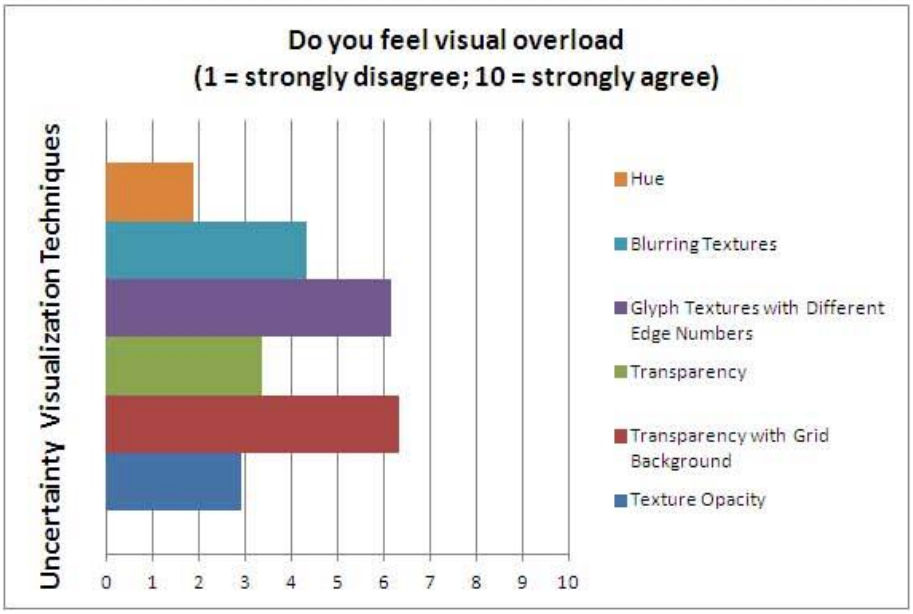


Figure 4.17: Average scores of the 6 uncertainty visualization techniques on the visual overload aspect.

4.4.2.4 Brightness

Figure 4.18 illustrates the average scores of the brightness for these techniques. We can see that the overall average scores of these techniques are quite similar (close to 5 points), which indicates they all have appropriate brightness for the participants' observation.

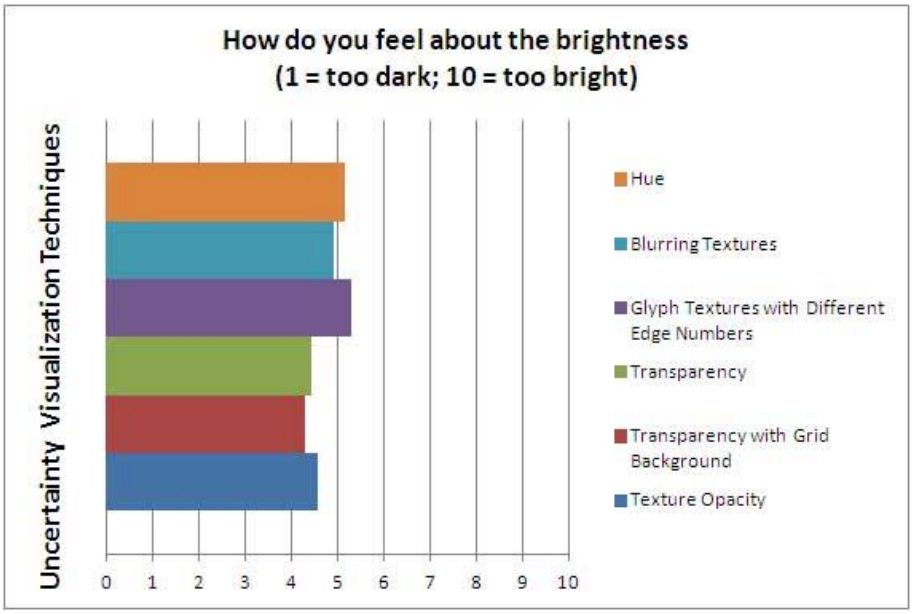


Figure 4.18: Average score of the 6 uncertainty visualization techniques on the brightness aspect.

4.4.3 Method of Analysis

Our analysis followed a common, classic statistical approach. First, a one-way (since we have only one independent variable, which is our uncertainty visualization techniques) multivariate analysis of variance (MANOVA) (Laerd Statistics, 2013a) is conducted to test the hypothesis that all 6 uncertainty visualization techniques have the same mean on the 4 aspects of perceptual effectiveness. If the result is true, we will accept the hypothesis and halt our analysis. If not, we will continue our analysis with 4 (since we have 4 aspects) univariate analysis of variance (ANOVA). In particular, considering that each participant is exposed to the 4 effectiveness aspects and having to response to each of them, we employed the univariate ANOVA's Repeated Measurements (RM) form (Laerd Statistics, 2013b). For each univariate RM ANOVA, it is associated to an individual effectiveness aspect and used to test the hypothesis that all 6 techniques have the same mean on that aspect. If the hypothesis is true, we will accept the hypothesis and draw our conclusions. If not, standard post hoc tests will be performed to test the significant difference between any one pair of these techniques in regard to that effectiveness aspect. In particular, we employed the Bonferroni post hoc test in this analysis, and since we have 6 different techniques, in total there are 15 pairwise comparisons.

4.4.4 Results and Discussion

The Statistical Product and Service Solutions (SPSS) is used for the statistical analysis. It is indicated by the Wilks' Lambda row in Table 4.1 that there is a statistically significant difference between the 6 uncertainty visualization techniques on all 4 aspects of effectiveness ($F(20,468.594)$=15.323,$p<0.0005$). Therefore 4 univariate RM ANOVAs are conducted to analyze the significant difference between the 6 techniques on each of the 4 effectiveness aspects.

4.4.4.1 Data Identification

Table 4.1: Multivariate tests[d] table.

Effect		Value	F	Hypothesis df	Error df	Sig.	Partial Eta Squared	Noncent. Parameter	Observed Power[b]
Intercept	Pillai's Trace	.978	1.535E3[a]	4.000	141.000	.000	.978	6139.405	1.000
	Wilks' Lambda	.022	1.535E3[a]	4.000	141.000	.000	.978	6139.405	1.000
	Hotelling's Trace	43.542	1.535E3[a]	4.000	141.000	.000	.978	6139.405	1.000
	Roy's Largest Root	43.542	1.535E3[a]	4.000	141.000	.000	.978	6139.405	1.000
techniques	Pillai's Trace	1.171	11.920	20.000	576.000	.000	.293	238.402	1.000
	Wilks' Lambda	.188	15.323	20.000	468.594	.000	.341	242.613	1.000
	Hotelling's Trace	2.573	17.950	20.000	558.000	.000	.391	358.997	1.000
	Roy's Largest Root	1.722	49.596[c]	5.000	144.000	.000	.633	247.981	1.000

a. Exact statistic

b. Computed using alpha = .05

c. The statistic is an upper bound on F that yields a lower bound on the significance level.

d. Design: Intercept + techniques

By analyzing both Mauchly's Test of Sphericity and Table 4.2 from the univariate RM ANOVA we know that there is a strongly significant difference between the 6 techniques on the data identification aspect ($\mathrm{F}(2.634,63.208) = 22.973, \mathrm{p} < 0.0005$, as shown in Table 4.2). Therefore we further performed a Bonferroni post hoc test to find out the significant difference between any two of these techniques. Table 4.3 illustrates the results of the test. From this table we can see that there is a strong significance between the hue technique and the transparency technique. Also, the mean of the hue technique is higher than the mean of the transparency

technique. Therefore, we can draw the conclusion that it is significantly much easier to use the hue technique than the transparency technique to identify the data. We applied the same method in the remainder of this section to summarize the results acquired from Bonferroni test.

Table 4.2: Tests of within-subjects effects on data identification.

Source		Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
techniques	Sphericity Assumed	226.193	5	45.239	22.973	.000	.489
	Greenhouse-Geisser	226.193	2.634	85.885	22.973	.000	.489
	Huynh-Feldt	226.193	2.988	75.701	22.973	.000	.489
	Lower-bound	226.193	1.000	226.193	22.973	.000	.489
Error(techniques)	Sphericity Assumed	236.307	120	1.969			
	Greenhouse-Geisser	236.307	63.208	3.739			
	Huynh-Feldt	236.307	71.711	3.295			
	Lower-bound	236.307	24.000	9.846			

From Table 4.3 we can also see that there is a strong significance between the hue technique and the transparency with enhanced grid background technique. This indicates that the hue technique is significantly much superior to the latter in the data identification aspect. As for the texture opacity and blurred textures techniques, they are all significantly much better than the two transparency-related techniques to identify the data. In terms of the glyph textures with different number of edges technique, whilst it is significantly much easier than the transparency with enhanced gird background technique to identify the data, it is slightly easier than the transparency technique.

Table 4.3: Statistically significant difference between the 6 techniques on the aspect of data identification.

		Hue	B.T.[a]	G.T.[b]	Trans.[c]	Trans.G.B.[d]	T.O.[e]
Mean(bigger)	Hue				**	**	
	T.O.[e]				**	**	
	B.T.[a]				**	**	
↓	G.T.[b]				*	**	
	Trans.[c]						
Mean(smaller)	Trans.G.B.[d]						

* indicates significance at the 0.05 level.

** indicates strong significance at the 0.025 level.

[a] B.T. is short for Blurring Textures technique.

[b] G.T. is short for Glyph Textures with Different Edge Numbers technique.

[c] Trans. is short for Transparency technique.

[d] Trans.G.B. is short for Transparency with Grid Background technique.

[e] T.O. is short for Texture Opacity technique.

Table 4.4: Tests of within-subject effects on error identification.

Source		Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
techniques	Sphericity Assumed	644.240	5	128.848	47.415	.000	.664
	Greenhouse-Geisser	644.240	3.578	180.052	47.415	.000	.664
	Huynh-Feldt	644.240	4.282	150.444	47.415	.000	.664
	Lower-bound	644.240	1.000	644.240	47.415	.000	.664
Error(techniques)	Sphericity Assumed	326.093	120	2.717			
	Greenhouse-Geisser	326.093	85.874	3.797			
	Huynh-Feldt	326.093	102.774	3.173			
	Lower-bound	326.093	24.000	13.587			

4.4.4.2 Error Identification

Based on both Mauchly's Test of Sphericity and Table 4.4 from the univariate RM ANOVA we know that there is a strongly significant difference between the 6 uncertainty visualization techniques on the error identification aspect ($F(3.578,85.874) = 47.415, p < 0.0005$, as illustrated in Table 4.4). Thus we performed the Bonferroni test to compare the significant difference between any two of these techniques. Table 4.5 presents the results of the test.

Table 4.5: Statistically significant difference between the 6 techniques on the aspect of error identification.

		Hue	B.T.[a]	G.T.[b]	Trans.[c]	Trans.G.B.[d]	T.O.[e]
Mean(bigger)	Hue		**	**	**	**	**
	T.O.[e]		*	**			
	Trans.[c]			**			
↓	Trans.G.B.[d]			**			
	B.T.[a]			**			
Mean(smaller)	G.T.[b]						

* indicates significance at the 0.05 level.

** indicates strong significance at the 0.025 level.

[a] B.T. is short for Blurring Textures technique.

[b] G.T. is short for Glyph Textures with Different Edge Numbers technique.

[c] Trans. is short for Transparency technique.

[d] Trans.G.B. is short for Transparency with Grid Background technique.

[e] T.O. is short for Texture Opacity technique.

From Table 4.5 we can see that there is a strong significance between the hue technique and the remaining techniques. This indicates that hue is significantly the best technique to identify errors. As for the texture opacity technique, whilst it is significantly a little bit easier than the blurred textures technique to identify errors, it is significantly much easier than the glyph textures with different number of edges technique to identify errors. In terms of the remaining three techniques, they are all

significantly much better than the glyph textures with different number of edges technique to identify errors.

Table 4.6: Tests of within-subjects effects on visual overload.

Source		Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
techniques	Sphericity Assumed	401.680	5	80.336	19.555	.000	.449
	Greenhouse-Geisser	401.680	3.546	113.276	19.555	.000	.449
	Huynh-Feldt	401.680	4.236	94.817	19.555	.000	.449
	Lower-bound	401.680	1.000	401.680	19.555	.000	.449
Error(techniques)	Sphericity Assumed	492.987	120	4.108			
	Greenhouse-Geisser	492.987	85.105	5.793			
	Huynh-Feldt	492.987	101.673	4.849			
	Lower-bound	492.987	24.000	20.541			

4.4.4.3 Visual Overload

By analyzing both Mauchly's Test of Sphericity and Table 4.6 from the univariate RM ANOVA we know that there is a strongly significant difference between the 6 uncertainty visualization techniques on the visual overload aspect ($\mathrm{F}(3.546,85.105) = 19.555, \mathrm{p} < 0.0005$, as illustrated in Table 4.6). Therefore we performed the Bonferroni test to test the significant difference between any two of these techniques. Table 4.7 illustrates the results of the test.

From Table 4.7 we can see that there is a strong significance between the hue technique and the rest of these techniques but not the texture opacity technique. This indicates that the hue technique generates significantly less visual overload than the other 4 techniques (blurred textures, glyph textures with different number of edges and the two transparency-related techniques). As for both the texture opacity and the transparency techniques, they are strongly significant than the glyph textures with different number of edges technique and the transparency with enhanced grid

background technique. Thus we can summarize that they are significantly less visual overload than the latter two techniques. Also, we can see from Table 4.7 that there is a strong significance between the blurred textures technique and the glyph textures with different number of edges technique. This indicates that the former display significantly less visual overload than the latter.

Table 4.7: Statistically significant difference between the 6 techniques on the aspect of visual overload.

		Hue	B.T.[a]	G.T.[b]	Trans.[c]	Trans.G.B.[d]	T.O.[e]
Mean(bigger)	Hue		**	**	**	**	
	T.O.[e]			**		**	
	Trans.[c]			**		**	
↓	B.T.[a]			**			
	G.T.[b]						
Mean(smaller)	Trans.G.B.[d]						

* indicates significance at the 0.05 level.

** indicates strong significance at the 0.025 level.

[a] B.T. is short for Blurring Textures technique.

[b] G.T. is short for Glyph Textures with Different Edge Numbers technique.

[c] Trans. is short for Transparency technique.

[d] Trans.G.B. is short for Transparency with Grid Background technique.

[e] T.O. is short for Texture Opacity technique.

4.4.4.4 Brightness

By analyzing both Mauchly's Test of Sphericity and Table 4.8 from the univariate RM ANOVA we know that there is a slightly significant difference between these 6 uncertainty visualization techniques on the brightness aspect ($F(3.004,72.097) = 3.191, p = 0.029$, as illustrated in Table 4.8). Thus we continued the analysis with the Bonferroni test to find out the statistically significant difference between any two of these techniques. It turns out from the Bonferroni analysis that there is no statistically significant difference between any of these 6 uncertainty visualization

techniques on the brightness aspect. However, this is inconsistent with the result from our univariate RM ANOVA. We attribute this to the conservatism of Bonferroni adjustment (Gordon et al., 2007).

Table 4.8: Tests of within-subjects effects on brightness.

Source		Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
techniques	Sphericity Assumed	20.693	5	4.139	3.191	.010	.117
	Greenhouse-Geisser	20.693	3.004	6.889	3.191	.029	.117
	Huynh-Feldt	20.693	3.482	5.944	3.191	.022	.117
	Lower-bound	20.693	1.000	20.693	3.191	.087	.117
Error(techniques)	Sphericity Assumed	155.640	120	1.297			
	Greenhouse-Geisser	155.640	72.097	2.159			
	Huynh-Feldt	155.640	83.560	1.863			
	Lower-bound	155.640	24.000	6.485			

4.5 Evaluation of Execution Time

Apart from the perceptual effectiveness, we also measure execution time for each of the 6 evaluated uncertainty visualization techniques, which are run on the lobster data set (Roettger, 2012), as illustrated in Table 4.9.

Table 4.9: Execution time of the 6 evaluated uncertainty visualization techniques.

Uncertainty visualization techniques	**Execution time (in seconds)**
Hue	2.875
Blurred textures	2.907
Glyph textures with different number of edges	2.907
Transparency	2.875
Transparency with enhanced grid background	2.906
Texture opacity	2.907

It is clear from this table that all the 6 evaluated uncertainty visualization techniques have a very similar execution time, although the execution time of the three texture-based uncertainty visualization techniques and the transparency with enhanced grid background uncertainty visualization technique are slightly longer than the execution time of the hue and transparency uncertainty visualization techniques. We attribute the relatively long execution time of the three texture-based uncertainty visualization techniques to their time spent on loading textures. We attribute the relative long execution time of the transparency with enhanced grid background uncertainty visualization technique to its time spent on drawing the grid background.

4.6 Summary

This chapter presented an investigation of the Texture uncertainty visualization technique to depict the errors associated to isosurface rendering. This includes three main steps. First, a two-step modeling approach used to generate the lower resolution univariate 3D data sets (whose each voxel incorporates both a scalar value s and an error value e associated to s), which serve as input data for subsequent Texture uncertainty visualization technique is introduced. Second, the actual Texture uncertainty visualization technique that includes two types of textures is presented. They are applied to the input data generated from the first main step to illustrate their results. Third, an evaluation work that compared the perceptual effectiveness of the Texture technique with other 4 existing uncertainty visualization techniques is shown. Also, the corresponding analysis method and analysis results are reported. The analysis results suggest that while both types of textures have some utility in certain perceptual aspects, they are less effective than the two existing hue and texture opacity techniques. Additionally, a surprising finding is that adding auxiliary grid lines, as background, does not guarantee to enhance the participants' perception to the errors conveyed by transparency. Conversely, it may result in visual overload that increases difficulty to recognize the scalar data. These findings can provide useful guidance for future uncertainty visualization design.

The next chapter will describe the work relevant to the LVIS uncertainty visualization technique, which is designed to visualize the uncertainty in DVR, as opposed to the work presented in this chapter.

Chapter 5 Uncertainty Visualization in DVR Using LVIS

5.1 Introduction

This chapter will present two elements of work in related to the LVIS uncertainty visualization technique (hypothesis 2) and ELVIS uncertainty visualization technique (extended work for hypothesis 2), as illustrated in Figure 3.13.

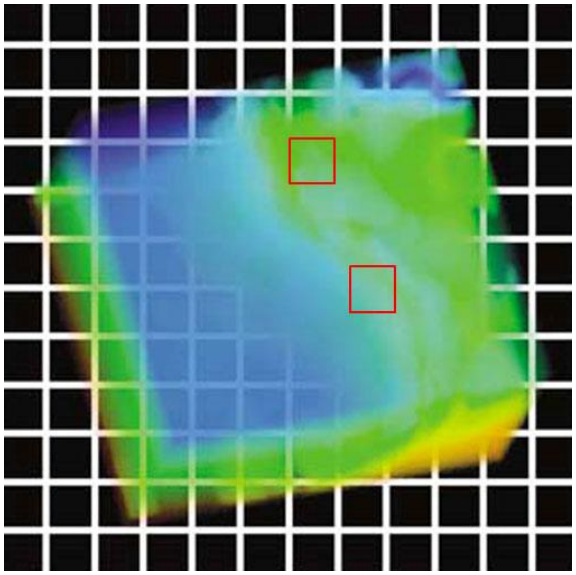


Figure 5.1: An example of the qualitative uncertainty visualization, where uncertainty is depicted by the degree of opacities (Djurcilov et al., 2002).

Many techniques of uncertainty visualization have been proposed to visualize uncertainty in DVR, since its significance has been emphasized by several leading researchers (Johnson, 2004)(Chen, 2005)(Johnson et al., 2006)(Laramee and Kosara, 2007)(Johnson and Sanderson, 2003). However, most of these techniques tend to only qualitatively depict uncertainty by displaying its location and relative size through certain visual metaphors, rather than quantitatively depict it (Potter, Rosen and Johnson, 2012). While they may be useful for those cases where only a coarse distinction of the uncertainty is sufficient, they are less useful for those cases where

a more accurate distinction of the uncertainty is required. In addition, these techniques cannot guarantee the user could draw the same conclusions from the final results because the uncertainty observation may vary from people to people. Figure 5.1 illustrates an example of the qualitative uncertainty visualization, where the high opacity indicates the high uncertainty, and the low opacity indicates the low uncertainty. While it may be very easy from this figure to distinguish the uncertainties with very different values *i.e.* the low uncertainty in the left columns and the high uncertainty in the right columns, it is very difficult to distinguish the uncertainties with similar values *i.e.* the uncertainties specified within the two red regions.

Therefore, the goal of this chapter is to explore new uncertainty visualization techniques that enable users to more accurately visualize uncertainty in DVR. To this end, a LVIS uncertainty visualization technique is proposed, which combines multiple linked views and a depth sample extraction interaction technique together to visualize uncertainty in univariate 3D scalar data. The evaluation results suggest that it indeed enables users to more accurately search both extreme (max. and min.) scalar and error values in univariate 3D scalar data than other 4 commonly used uncertainty visualization techniques. Thus, we further extended it to become ELVIS, which aims to visualize uncertainty in multivariate 3D scalar data.

5.2 MR Modeling and Quantification for Multivariate 3D Scalar Data

Section 4.2 presented a two-step modeling approach (MR modeling and quantification) that is applied to an original univariate 3D scalar data set to generate its lower resolution univariate 3D data sets (with their each voxel including both a scalar value s and an error value e associated to s), which can be used as the input data for the LVIS uncertainty visualization technique that will be introduced in the next section.

In addition to the univariate 3D scalar data, the two-step modeling approach can also be applied to an original multivariate 3D scalar data set to generate its lower

resolution multivariate 3D data sets (with their each voxel including scalar values $s^1, s^2, \cdots, s^k$ that correspond to k variables, as well as k error values $e^1, e^2, \cdots, e^k$ associated to the k scalar values), which can be used as the input data for the ELVIS uncertainty visualization technique (extended LVIS technique) that will be introduced in Section 5.6. This section will describe the latter.

5.2.1 MR Modeling

Again, both the Haar Wavelet Transform (Li and Drew, 2004) and the Decimation (Foulks and Bergeron, 2009) techniques have been exploited in this thesis to generate a MR hierarchy.

5.2.1.1 Haar Wavelet Transform

The case of applying the Haar Wavelet Transform to a multivariate 3D scalar data set is relatively complicated. Thus, an example is taken to illustrate it. Here the notation s_n^m is used to represent a scalar value, where the subscript represents its associated voxel index (beginning with 0), while the superscript represents its corresponding variable. For example, s_3^2 represents a scalar value that is associated to variable 2 and distributed at voxel 3.

Figure 5.2(a) shows a $8 \times 8 \times 8$ original multivariate 3D scalar data set R_0 (R represents resolution), with its each voxel includes scalar values $s_*^1, s_*^2, \cdots, s_*^k$ (where $*$ refers to arbitrary voxel index) that correspond to k variables. Figure 5.2(b) shows its $4 \times 4 \times 4$ lower resolution multivariate 3D scalar data set R_1, whose size is $\frac{1}{8}$ of R_0, with each voxel also includes scalar values $s^1, s^2, \cdots, s^k$ that correspond to the k variables. The process of how the Haar Wavelet Transform transforms R_0 to R_1 can be summarized as follows: (1) for a cell of R_0 *i.e.* the cell 1 as indicated by the red rectangle in Figure 5.2(a), apply formula 5.1:

$$\begin{cases} s^1 = \frac{s_0^1 + s_1^1 + \cdots + s_{n-1}^1}{n} \\ s^2 = \frac{s_0^2 + s_1^2 + \cdots + s_{n-1}^2}{n} \\ \vdots \\ s^k = \frac{s_0^k + s_1^k + \cdots s_{n-1}^k}{n} \end{cases} \quad (5.1),$$

to it to compute a set of average scalar values $s^1, s^2, \cdots, s^k$, which are considered as the scalar values of R_1's voxel, as indicated by the red rectangle in Figure 5.2(b). (2) traverse to another cell of R_0 and repeat (1) until all cells of R_0 have been processed. As a result, R_1 is generated.

Figure 5.2(c) shows R_0's a further $2 \times 2 \times 2$ lower resolution multivariate 3D scalar data set R_2, whose size is $\frac{1}{64}$ of R_0. The process of how the Haar Wavelet Transform transforms R_0 to R_2 can be summarized as follows: (1) for 8 cells of R_0, say the 8 cells that are highlighted in red in Figure 5.2(a), apply formula 5.1, where $\mathrm{n} = 64$ (as every 64 scalar values that correspond to an identical variable and are at the 64 voxels of 8 cells are used to compute an average scalar value corresponding to that variable) to them to compute a set of average scalar values $s^1, s^2, \cdots, s^k$ for the R_2's voxel that is indicated by the red rectangle in Figure 5.2(c). (2) traverse to another 8 cells of R_0 and repeat (1) until all cells of R_0 have been processed. As a result, R_2 is generated, with its each voxel including $s^1, s^2, \cdots, s^k$ that correspond to the k variables.

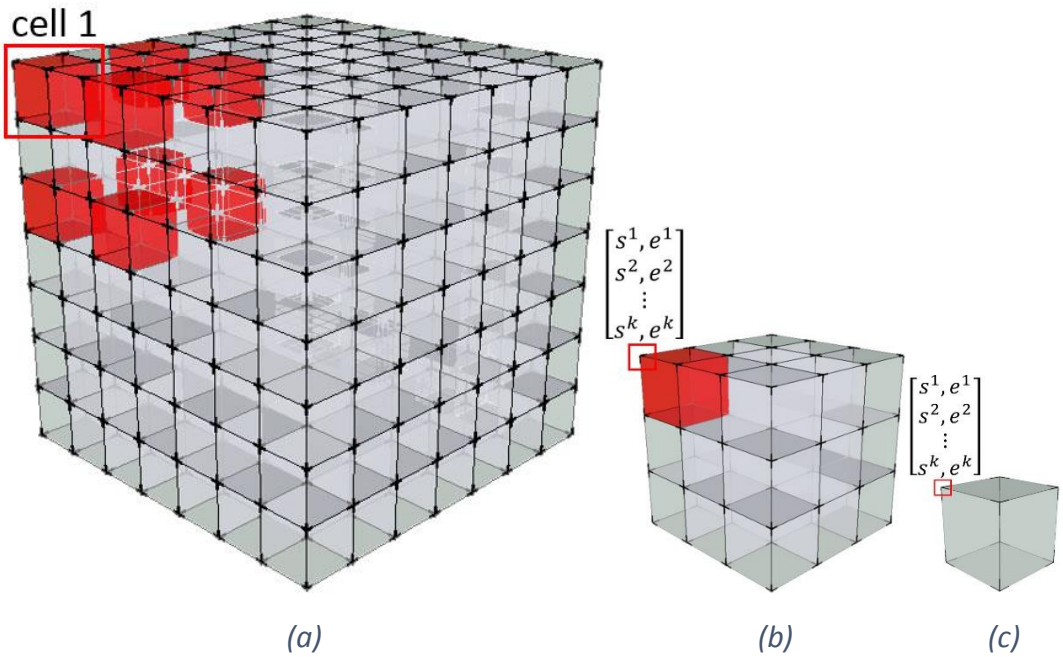


Figure 5.2: An example of a MR hierarchy.

5.2.1.2 Decimation

Continuing with the example above, the process of how the Decimation transforms R_0 to R_1 can be summarized as follows: (1) for a cell of R_0, say the cell 1 that is indicated by the red rectangle in Figure 5.2(a), apply formula 5.2:

$$\begin{cases} s^1 = subsampling(s_0^1, s_1^1, \cdots, s_{n-1}^1) \\ s^2 = subsampling(s_0^2, s_1^2, \cdots, s_{n-1}^2) \\ \vdots \\ s^k = subsampling(s_0^k, s_1^k, \cdots, s_{n-1}^k) \end{cases} \quad (5.2),$$

where $\mathrm{n} = 8$ (as every 8 scalar values that correspond to an identical variable and are at the 8 voxels of a cell are used to subsample a scalar value corresponding to that variable) to it to compute a set of scalar values $s^1, s^2, \cdots, s^k$ for the R_1's voxel that is indicated by the red rectangle in Figure 5.2(b). (2) traverse to another cell of R_0 and repeat (1) until all cells of R_0 have been processed. As a result, R_1 is generated, with its each voxel including scalar values $s^1, s^2, \cdots, s^k$ that correspond to the k variables.

The same method can be used to summarize the process of how the Decimation transforms R_0 to R_2: (1) for 8 cells of R_0, say the 8 cells that are highlighted in red in Figure 5.2(a), apply formula 5.2, where $\mathrm{n} = 64$ (as every 64 scalar values that correspond to an identical variable and are at the 64 voxels of 8 cells are used to subsample a scalar value corresponding to that variable) to them to compute a set of scalar values $s^1, s^2, \cdots, s^k$ for the R_2's voxel that is indicated by the red rectangle in Figure 5.2(c). (2) traverse to another 8 cells of R_0 and repeat (1) until all cells of R_0 have been processed. As a result, R_2 is generated, with its each voxel including $s^1, s^2, \cdots, s^k$ that correspond to the k variables.

5.2.2 Quantification

Following the example described above, the process of quantifying the errors between R_0 and R_1 can be summarized as follows: (1) for a R_0's cell, say the cell 1 that is indicated by the red rectangle in Figure 5.2(a) and its derived R_1's voxel that is indicated by the red rectangle in Figure 5.2(b), apply formula 5.3:

$$\begin{cases} e^1 = \sqrt{\frac{1}{n}[(s_0^1 - s^1)^2 + (s_1^1 - s^1)^2 + \cdots + (s_{n-1}^1 - s^1)^2]} \\ e^2 = \sqrt{\frac{1}{n}[(s_0^2 - s^2)^2 + (s_1^2 - s^2)^2 + \cdots + (s_{n-1}^2 - s^2)^2]} \\ \vdots \\ e^k = \sqrt{\frac{1}{n}[\left(s_0^k - s^k\right)^2 + \left(s_1^k - s^k\right)^2 + \cdots + (s_{n-1}^k - s^k)^2]} \end{cases} \quad (5.3),$$

where $\mathrm{n} = 8$ to them to compute error values $e^1, e^2, \cdots, e^k$ that are associated to the derived R_1' voxel's scalar values $s^1, s^2, \cdots, s^k$. (2) traverse to another R_0's cell and its corresponding derived R_1's voxel, repeat (1) until all R_0's cells (or all R_1's voxels) have been processed. As a result, the error values associated to R_1's each voxel's scalar values can be obtained.

In the same way, the error values associated to R_2's each voxel's scalar values can be computed. But instead of using R_0's each cell in formula 5.3, it is necessary to use its every 8 neighboring cells in formula 5.3, that is $\mathrm{n} = 64$, to obtain the corresponding results.

Now that the lower resolution multivariate 3D data sets, such as R_1 and R_2, whose each voxel includes both scalar values $s^1, s^2, \cdots, s^k$ that correspond to k variables and their associated error values $e^1, e^2, \cdots, e^k$ are available, they can be used as the input data for subsequent ELVIS uncertainty visualization technique.

5.3 LVIS Uncertainty Visualization Technique for Univariate 3D Scalar Data

This section corresponds to the content in (Ma et al., 2012c), where a LVIS technique is proposed, which is intended to enable users to more accurately visualize the uncertainty in univariate 3D scalar data.

The concept of linked views is not new in the visualization community, and many researchers have used it in their visualization study to explore data in different application domains (Akiba and Ma, 2007)(Akiba et al., 2007)(Gasser, 2004) (Piringer, Kosara and Hauser, 2004)(Jones et al., 2008)(Linsen, Long and Rosenthal, 2009). Compared to those single view-based visualization techniques, the linked views-based visualization has three main advantages. First, it could combine various

visualization techniques into a more powerful visualization tool to give deeper insight. Second, it separates large amount of diverse information into many individual views and thus yields a clearer data presentation. Third, it facilitates a greater level of visual data analysis. Because of these advantages, in recent years some works (Potter et al., 2009)(Sanyal et al., 2010) have been proposed to apply the linked views to uncertainty visualization studies. However, these works are mainly based on 2D data, and cannot directly apply to the 3D data. This is because in contrast to 2D data, the 3D data always contain an inherent "depth" dimension. Thus, the traditional interaction techniques that are used to extract the relevant information of a specified ROI from the 2D data are not suitable for the 3D case. New interaction techniques have to be developed to overcome this issue. To this end, a depth sample extraction technique has been developed, which can be used to extract the 3D scalar and its associated error sample data from a specified 2D ROI.

5.3.1 Workflow

Figure 5.3 shows the proposed visualization's workflow, which is simplified from Figure 3.15. It is designed by keeping a real-world scenario in mind: scientists like to explore data by first looking at the entire data, and then focusing on a ROI (Sanyal et al., 2009). It begins with a process of loading data, where the user is asked to load both the scalar and its associated error data interested to explore. Once the data are loaded successfully, three separate views will be displayed on the screen.

First, we use volume ray-casting algorithm to render the lower resolution univariate 3D scalar and error data set's scalar values to generate the scalar data object-space view, for users' inspection. Second, we use volume ray-casting algorithm to render its error values to generate the error data object-space view, for users' inspection. Third, we draw both its scalar and error values to generate the scatter plot feature-space view to show its distribution feature. Users are able to specify a ROI in either the scalar or error data object-space view. Once a ROI is specified, two things will occur. First, a marked area will synchronously appear in the same position of both views to denote the specified ROI. Second, both the scalar and its associated error data inside the ROI will be extracted and plotted on a scatter plot for quantitative

analysis. Here it is worth mentioning that the data extracted from the ROI are not simply the 2D pixel values. They in fact contains all 3D scalar and its error data samples that are acquired from the volume ray-casting rendering and associated with all the pixels inside the ROI. As a result, although the user specified a 2D ROI, he can obtain the quantitative "3D insight" of both the scalar and its associated error data inside it based on the scatter plot. He could also repeatedly refine the ROI based on the feedback of the scatter plot until get satisfied results *i.e.* he wants to search a ROI inside which the maximum error is smaller than a specified threshold.

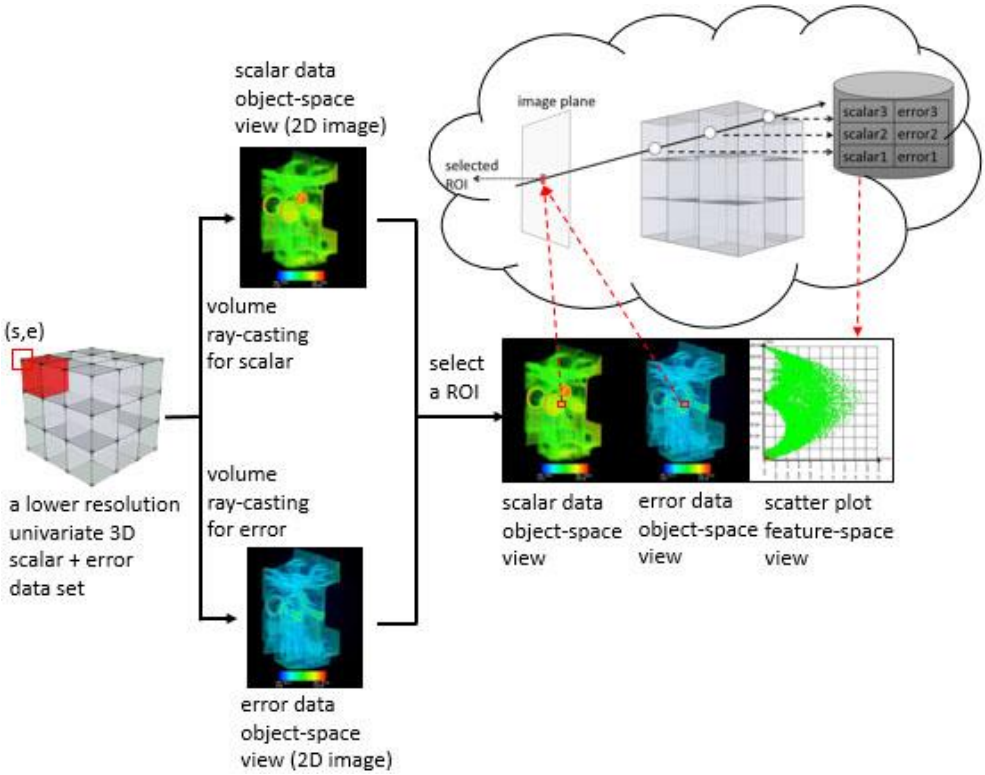


Figure 5.3: Overview of the LVIS workflow.

5.3.2 Visualization Components

This section presents the three components that consist of the proposed visualization.

5.3.2.1 Scalar/Error Data Object-space View

These two views have exactly the same functionalities but with an exception: the scalar data object-space view is only used to render the scalar data, while the error data object-space view is only used to render the associated error data. The reason

they are kept separately from each other is because in this way it can yield a clearer data presentation. For both views, they are rendered by the classic volume ray-casting algorithm. In addition, they allow the user to specify a ROI on them by dragging the mouse. Once a ROI has been specified, there will be a marked area appeared synchronously in the same position of both views to denote the ROI. They are arranged in a side by side manner for an easy comparison and observation.

Here the object-space views refer to those views which are rendered by the volume visualization techniques and thus are capable to present the corresponding 3D spatial characteristics of the data. They are relative to the feature-space views, which are often rendered by the information visualization techniques and only present the data values' distribution characteristics. These two terms are often appeared in a sub-field of visualization research where both scientific visualization techniques and information visualization techniques are combined. See (Linsen, Long and Rosenthal, 2009)(Linsen et al., 2008) for examples.

5.3.2.2 Scatter Plot Feature-space View

This component is a straightforward scatter plot which is used to display the features of the scalar and its associated error data inside the ROI specified by the user. The horizontal axis of the scatter plot represents the error values, while its vertical axis represents the scalar values. It will be automatically plotted and updated once a ROI has been specified. Consequently the user can obtain the quantitative and immediate feedback from it.

5.3.2.3 Linking between Views and Depth Sample Extraction

The three views mentioned above are linked through the user's interaction for a ROI specification. A key question here is how to extract the 3D scalar and its associated error sample data inside the 2D ROI. A depth sample extraction interaction technique has been proposed for this purpose. It can be considered as a two-step approach, as illustrated by the content inside the cloud of Figure 5.3. First, during the process of volume ray-casting rendering to generate the scalar and error data object-space view, we pre-stored the sampled scalar and its associated error data that correspond to every pixel of these two views into the memory. In this way for any one of pixels on

these two views, we can extract and acquire its corresponding scalar and error sample data. Second, when a user specified a 2D ROI in either the scalar or error data object-space view, we firstly find out all pixels that are inside this ROI, and then extract their corresponding scalar and error sample data from the memory, and finally plot these data onto the scatter plot. As a result, we reached this target that enables the user to obtain quantitative 3D scalar and its associated error sample data within the 2D ROI.

5.3.3 Case Study

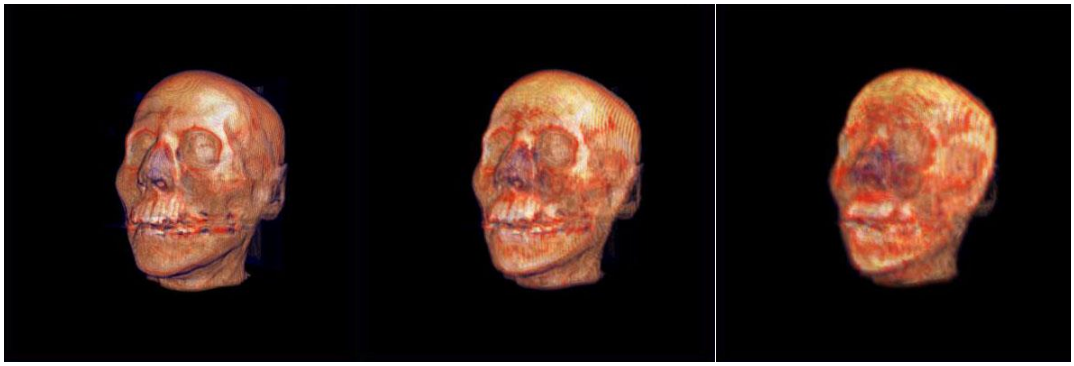
A case study is shown here to illustrate the LVIS uncertainty visualization technique's application. Section 5.3.3.1 introduces the original data set and its pre-processing to generate the lower resolution univaraite 3D data sets that are used as input by the LVIS technique. Section 5.3.3.2 presents the corresponding results.

5.3.3.1 Original Data Set and Pre-processing

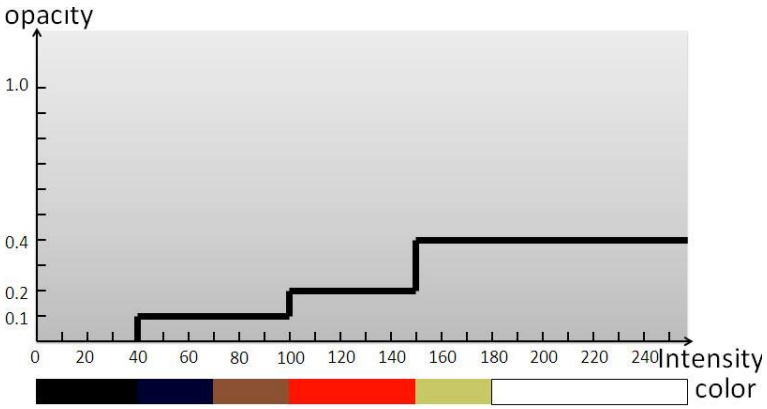
The original data set exploited to generate the lower resolution univariate 3D data sets is a female cadaver head from CT scan, available at (Levoy, 2008). The dimensions of the original data set are $256 \times 256 \times 113$, with 8 bits per voxel. For convenience of the 3D Haar Wavelet Transform or Decimation, we generate 143 extra slices whose scalar values are all 0 and append them at the end of the original data set. As a result, a $256 \times 256 \times 256$ derived data set is formed. This derived data set is then processed by the two-step modeling approach (as described in Section 4.2) to generate a MR hierarchy (including associated errors for the lower resolution data sets) with 3 different levels of resolution.

Figures 5.4(a), (b) and (c) illustrate an example of the MR hierarchy generated from the Haar Wavelet Transform. We can see that the head shown in Figure 5.4(a) has the greatest detail, and it corresponds to the derived data set whose dimensions are $256 \times 256 \times 256$. Compared to the head shown in Figure 5.4(a), the one shown in Figure 5.4(b) is slightly coarser *i.e.* the image in the regions of nose and mouth becomes unclear, which indicates some details have been lost. It corresponds to the MR data set after one time Haar Wavelet Transform and its dimensions are $128 \times 128 \times 128$. Figure 5.4(c) shows the coarsest version of the head and we can see

from this figure that a substantial amount of details have been lost. It corresponds to the MR data set after two times Haar Wavelet Transform and its dimensions are $64 \times 64 \times 64$. Figure 5.4(d) shows the TF used to generate these images. For easy reference of these MR data sets in the subsequent paragraphs, we refer to the $256 \times 256 \times 256$ data set as R_0, the $128 \times 128 \times 128$ data set as R_1, and the $64 \times 64 \times 64$ data set as R_2.



(a) *(b)* *(c)*



(d)

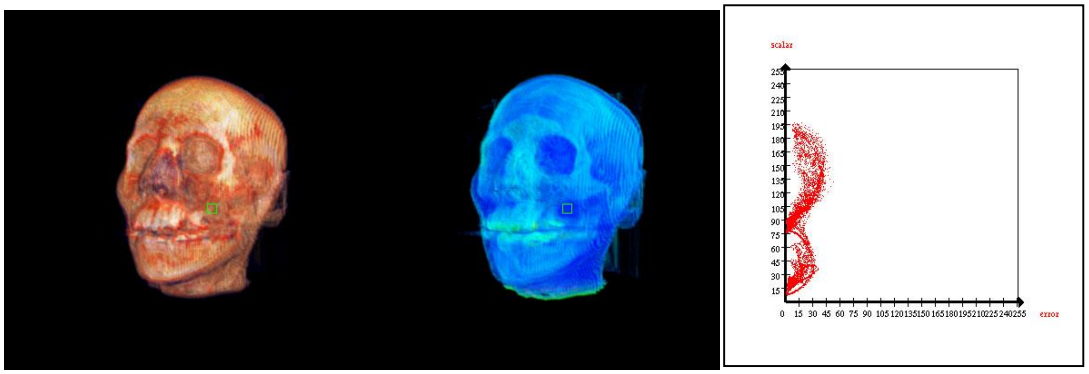
Figure 5.4: An example of a MR hierarchy generated from the Haar Wavelet Transform. (a) R_0 data set without the Haar Wavelet Transform. (b) R_1 data set after one time Haar Wavelet Transform. (c) R_2 data set after two times Haar Wavelet Transform. (d) the corresponding TF used to render these data sets.

5.3.3.2 Results and Discussion

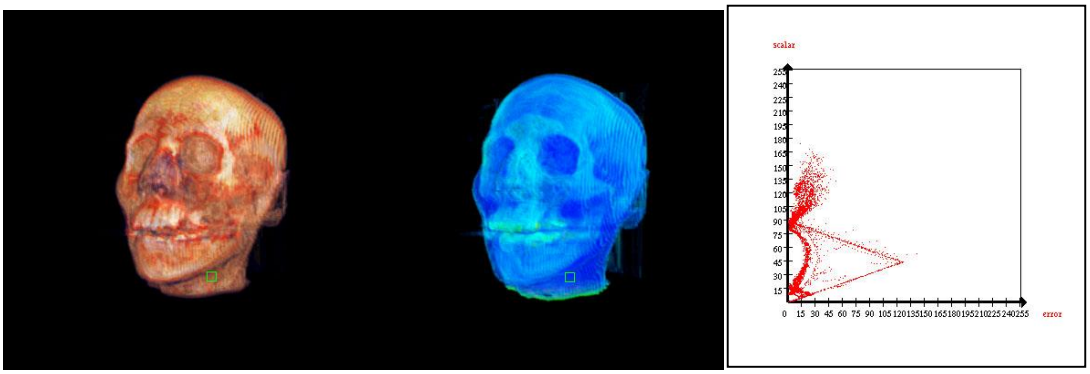
With the MR data sets readily available, we can now quantitatively explore them by using the proposed uncertainty visualization technique. Here we present three specific examples that we feel interested in exploration. For the first example, we want to test whether the proposed visualization can reveal the delicate differences

of errors between two ROIs with very similar error values, which are often difficult to distinguish by most qualitative uncertainty visualization techniques (which only rely on visual metaphors such as color to distinguish uncertainty). For the second example, we want to quantitatively compare the differences of data features between two ROIs at different positions of a MR data set. For the third example, we want to quantitatively find out how the data features change inside a ROI when it is applied to two successive MR data sets.

Figures 5.5(a) and (b) illustrate the results of the first example. The data set that is utilized here is R_1, generated from the Haar Wavelet Transform. The left column images of both Figures 5.5(a) and (b) represent the scalar data object-space view of the proposed visualization, while the middle column images represent the error data object-space view. The right column images represent the scatter plot component. We can see from the two error data object-space views (in these views the error values are mapped to the rainbow colors. Blue colors indicate fewer errors, while red colors indicate larger errors) that the two ROIs (10 pixels width and 10 pixels height) indicated by the two green marked areas have very close colors, and it is thus very difficult to distinguish the error size between them by simply observing their colors. To effectively discriminate their errors, a quantitative feedback of the errors is necessary. This requirement can be easily achieved by the scatter plot component of the proposed visualization. From the scatter plots in Figures 5.5(a) and (b) we can see that the maximum error of the ROI specified above the mouth is close to 60, while the maximum error of the ROI specified on the neck is close to 150, although most errors in these two ROIs are between 0 and 45. Thus, the maximum error of the latter is more than two times than the one of the former. Through the proposed visualization, a more accurate error comparison between two similar ROIs is achieved.



(a)

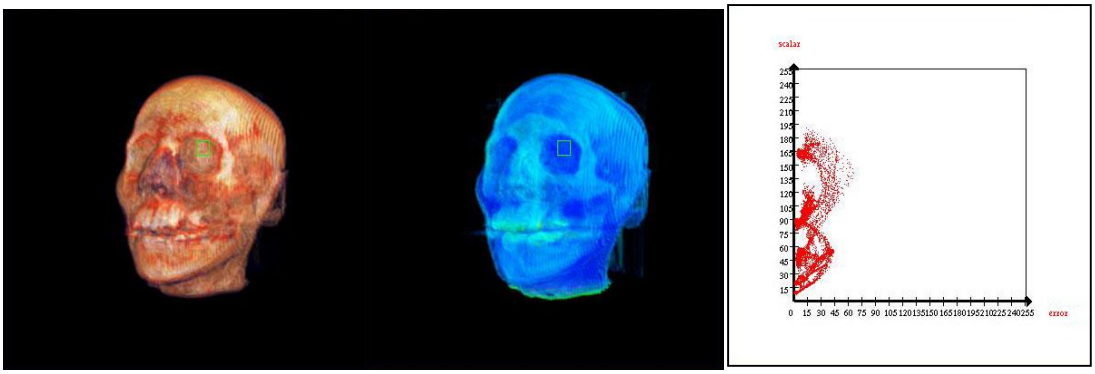


(b)

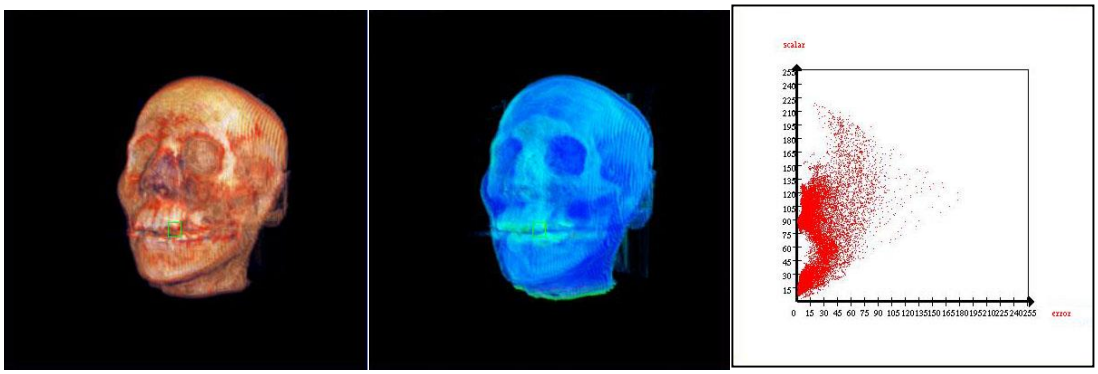
Figure 5.5: The LVIS uncertainty visualization is applied to R_1 to reveal the delicate error differences between the two ROIs that are indicated in (a) and (b), which have very similar error values.

Figures 5.6(a) and (b) show the results with respect to the second example. The data set used for this example is R_1, generated from the Haar Wavelet Transform. From these two figures we can see that two ROIs have been specified for the comparison. One is inside the left eye, and another one is around the mouth. Both ROIs have the same size, with 14 pixels width, and 16 pixels height. By observing the scatter plot in Figure 5.6(a) we can see that the error range of the left eye ROI is $[0, 75)$, with most of its errors are concentrated between 0 and 45, and only a few are beyond 45. Its scalar range is $(0, 195)$, with only a few scalar data are concentrated between 120 and 150. By observing the scatter plot in Figure 5.6(b) we can see that the error range of the mouth ROI is $[0, 180)$, with most of its errors are concentrated between 0 and 60, and only a few errors are beyond 60. As for its scalar data, they are ranged within $(0, 225)$, with most scalar data are concentrated between 0 and 150, and only a few are beyond 150. By comparing both scatter plots together, we can see that the error

range in the mouth ROI is more than twice greater than the one in the left eye ROI, while its scalar range is nearly 30 intensity units greater than the one in left eye ROI. As a result, an accurate comparison of the differences of the data features between two ROIs at different positions of the same MR data set is achieved.



(a)

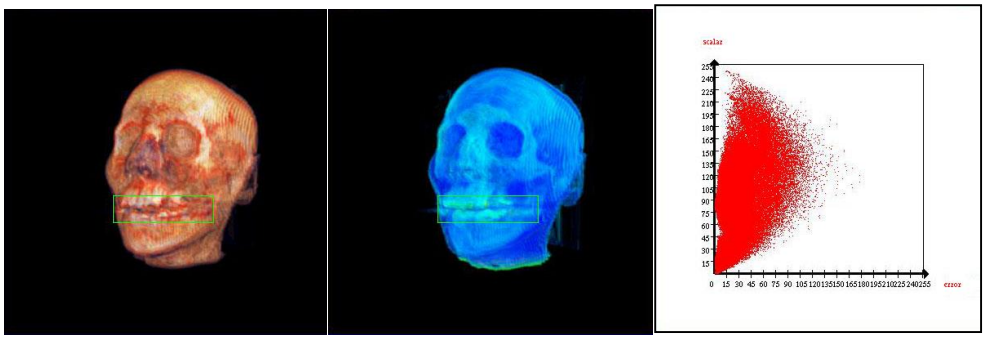


(b)

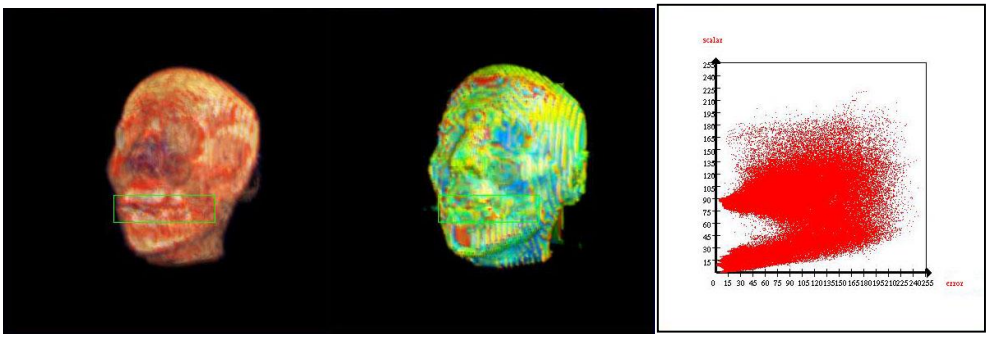
Figure 5.6: The LVIS uncertainty visualization is applied to R_1 to quantitatively compare the differences of data features between two ROIs at different positions. (a) Results regarding the left eye ROI. (b) Results regarding the mouth ROI.

Figures 5.7(a) and (b) show the results with respect to the third example. Figure 5.7(a) corresponds to the data set R_1, while Figure 5.7(b) corresponds to the data set R_2. From both figures we can see that a mouth ROI that is indicated by the green marked area has been specified for both MR data sets. Its size is 123 pixels width, and 33 pixels height. From the scatter plot in Figure 5.7(a) we can see that the mouth ROI's error range for the data set R_1 is $[0, 180)$, with most of its errors are concentrated between 0 and 105, and only a few errors are beyond 105. As for its scalar data, they are ranged within $[0, 255)$ in almost a uniform manner. From the scatter plot in Figure 5.7(b) we can see that the mouth ROI's error range for the data set R_2 is

$[0, 255)$, with most of the errors are concentrated between 0 and 195, and only some of them are beyond 195. As for its scalar data, they have a range of $[0,\ 225)$, with most scalar data are concentrated between 0 and 180, and only few are beyond 180. Compared to the two scatter plots together, we can see that the error range shown in the scatter plot in Figure 5.7(b) is about 75 intensity units greater than the one shown in the scatter plot in Figure 5.7(a), while the scalar range shown in the scatter plot in Figure 5.7(b) is about 30 intensity units smaller than the one shown in the scatter plot in Figure 5.7(a). Based on the quantitative analysis of the data features change within the mouth ROI between R_1 and R_2, an inference about how the Haar Wavelet Transform works can be made even one has no knowledge of it: it actually reduces the scalar data values it is processing while introduces bigger errors. By exploring other ROIs *i.e.* the nose (as shown in Figures 5.8(a) and (b)) or ear (as shown in Figures 5.9(a) and (b)) under the same scenario using the proposed visualization, this inference could be validated.

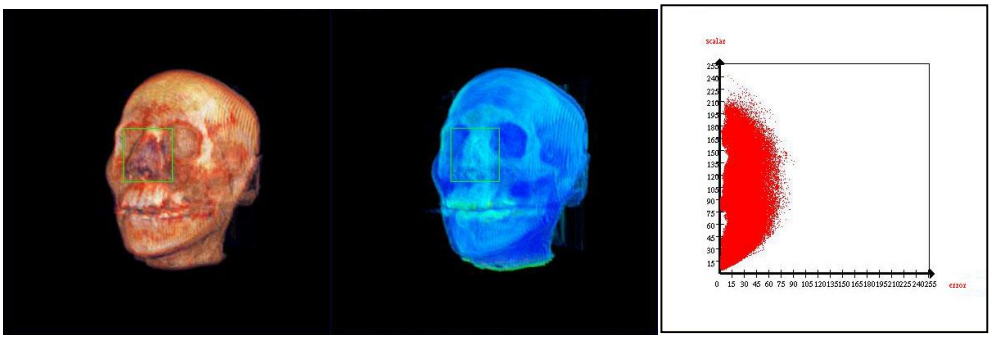


(a)

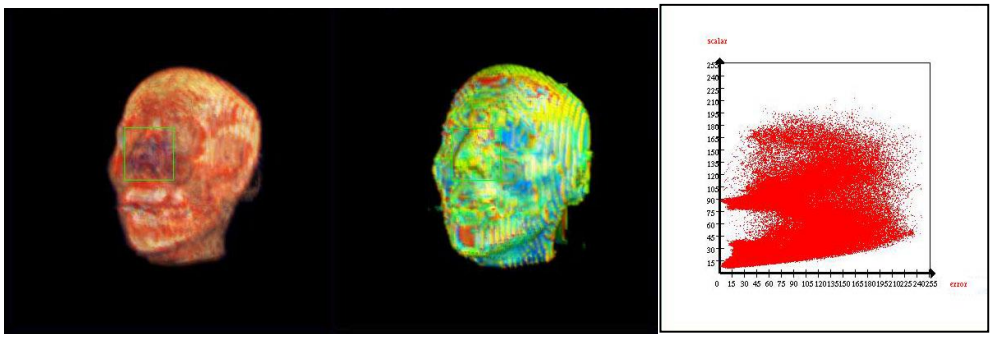


(b)

Figure 5.7: The LVIS uncertainty visualization is applied to (a) R_1 and (b) R_2 to reveal the data feature change inside the same mouth ROI.

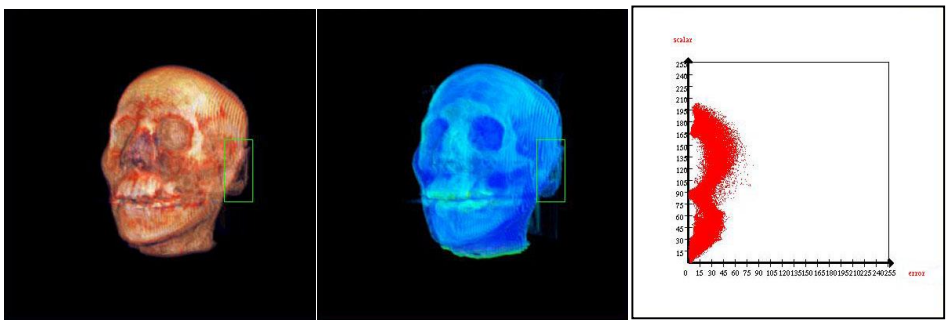


(a)

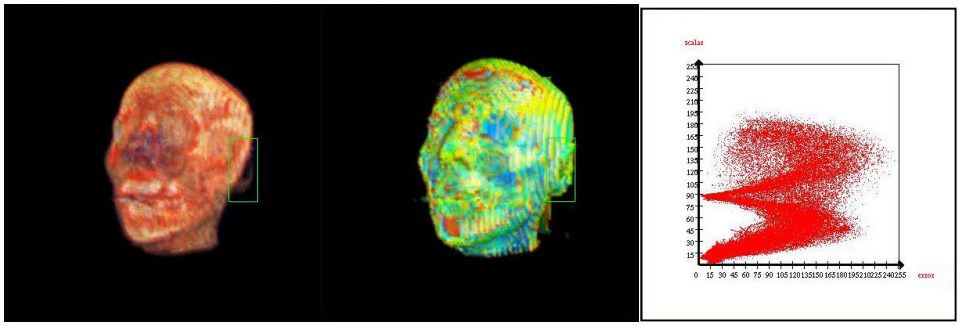


(b)

Figure 5.8: The LVIS uncertainty visualization is applied to (a) R_1 and (b) R_2 to reveal the data feature change inside the same nose ROI.



(a)



(b)

Figure 5.9: The LVIS uncertainty visualization is applied to (a) R_1 and (b) R_2 to reveal the data feature change inside the same ear ROI.

5.4 Evaluation of Perceptual Effectiveness

Like the previously proposed LVIS uncertainty visualization technique, many uncertainty visualization techniques have been proposed to visualize uncertainty in DVR, and some seem to appear more effective than others (Sanyal et al., 2009). However, to our knowledge, no research has been conducted to evaluate their perceptual effectiveness. Therefore we conducted an evaluation work to compare the perceptual effectiveness of the proposed LVIS techniques with other 4 commonly used DVR-based uncertainty visualization techniques. Such an evaluation has double significance. First, it could help us to validate the effectiveness of the proposed LVIS uncertainty visualization technique. Second, it could find out advantages and disadvantages of those existing uncertainty visualization techniques. The evaluation results will provide useful guidance for future uncertainty visualization design. This evaluation corresponds to the content in (Ma et al., 2013b).

5.4.1 Study Design

A user study has been designed to compare the perceptual effectiveness of the LVIS technique with other 4 existing uncertainty visualization techniques. This includes the method of uncertainty data modeling, the chosen uncertainty visualization techniques, the participant pool and their study tasks, the interface design, the participant training, and finally the main user study.

Usability and effectiveness studies have been used by researchers to evaluate new systems and new techniques. Some studies (Sanyal et al., 2009)(McCartney, 1997)(Hinchley et al., 1997) utilized task completion accuracies and time as foundation for their evaluation. Some (Newman and Lee, 2004)(Cassell and Vihjalmsson, 1999)(Rickenberg and Reeves, 2000) utilized user ratings of effectiveness as their evaluation basis. In this work, we utilized the first approach for our research.

5.4.1.1 Uncertainty Data Modeling

The uncertainty data modeling method exploited in this user study strictly follows the two-step modeling approach, as described in Section 4.2. It is applied to both the

engine (Roettger, 2012) and the female cadaver head (Levoy, 2008) univariate 3D scalar data sets to generate their corresponding lower resolution univariate 3D data sets (with each voxel corresponding to a scalar value s and an error value e associated to s), which are used as the input data for the 5 evaluated uncertainty visualization techniques, as will be described in the following section.

5.4.1.2 Five Uncertainty Visualization Techniques Chosen for Evaluation

Four existing uncertainty visualization techniques (Transparency, Discontinuities, Overlays and Hue) and the LVIS uncertainty visualization technique are chosen for this evaluation. They are all based on DVR, in particular, the volume ray-casting rendering (see Section 3.1.2.2 for detail). The four existing techniques will be introduced first, followed by the LVIS technique.

Transparency: recall the standard volume ray-casting algorithm that is mentioned in Section 3.1.2.2. Traditionally, it only uses a 3D scalar data set's scalar values to determine voxels' both RGB and opacity value α (through TFs), and based on this to generate the final 2D image. However, since now the data we are interested consist of both scalar values and their associated errors, we have opportunity to use the scalar values to determine the voxels' RGB, and use the error values to determine their opacity value α. This forms the basic idea of the transparency technique.

Figure 5.10 illustrates an example of this transparency technique. The scalar data are mapped to the rainbow colors. The blue colors indicate relatively small scalar data, while the red colors indicate relatively big scalar data. Other colors indicate in-between scalar data. In addition, the error data are mapped to the transparency. The big transparencies indicate relatively small errors, while the small transparencies indicate relatively big errors. Other transparencies indicate in-between errors.

The implementation of this technique is relatively simple and its resulting image can be directly generated from the volume ray-casting rendering.

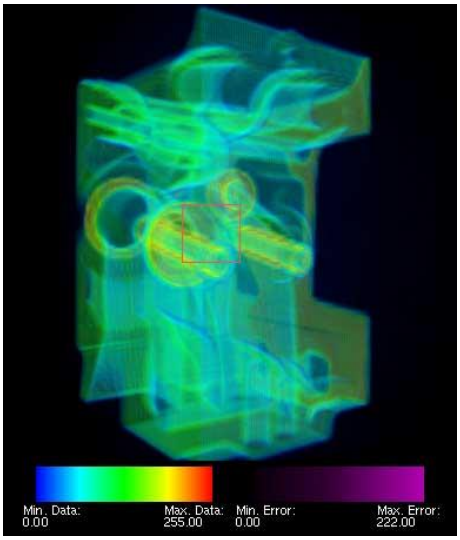


Figure 5.10: The transparency technique.

Discontinuities: in this technique, the scalar data are again mapped to the rainbow colors, the same as the transparency technique. However, their associated error data are mapped to the sizes of holes. The small-sized holes indicate relatively small error data, while the big-sized holes indicate relatively big error data. Other sized holes indicate in-between error data.

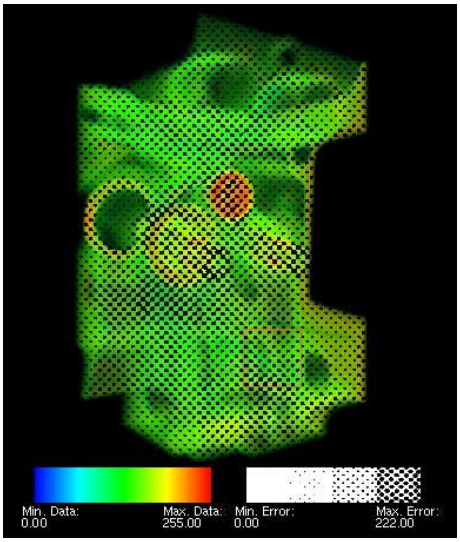


Figure 5.11: The discontinuities technique.

The implementation of this technique is more complex, in contrast to the transparency technique. Four steps have to be involved to obtain the resulting image: (1) produce a standard volume rendering image of the scalar data. (2) produce a grey-scale volume rendering image of the error data from the same viewpoint. (3) dither (Bayer, 1973) the grey-scale image into a black and white bitmap with inverted

values.(4) generate a composite image by laying the bitmap on top of the standard volume rendering image pixel by pixel if its pixel value is black. If not, then display the pixel value of the standard volume rendering image. Figure 5.11 illustrates an example of the discontinuities technique.

Overlays: similar to the previous two techniques, in this technique, the scalar data are mapped to the rainbow colors. However, their associated error data are mapped to greyscales. The dark greys indicate relatively small error data, while the light greys indicate relatively big error data. Other greys indicate in-between errors.

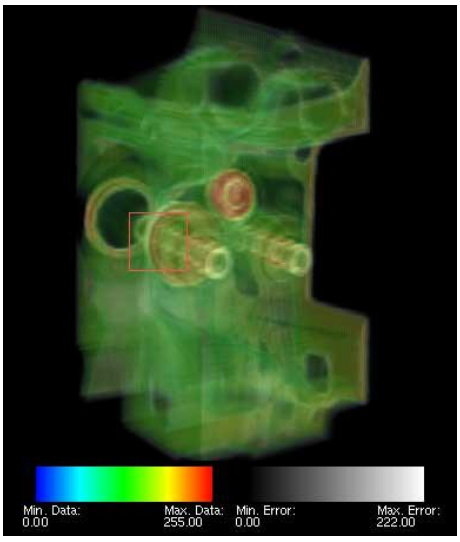


Figure 5.12: The overlays technique.

The implementation of this technique is also complex and three steps are involved to obtain a resulting image: (1) produce a standard volume rendering image of the scalar data. (2) produce a greyscale volume rendering image of the error data from the same viewpoint. (3) generate a composite image by overlaying these two images together, with the opacity of the standard volume rendering image being 0.35, and the opacity of the greyscale image being 0.65. Figure 5.12 illustrates an example of the overlays technique.

Hue: this technique is an inverse version of the transparency technique. This time, the scalar data are mapped to the transparency. Big transparencies indicate relatively small scalar data, while small transparencies indicate relatively big scalar data. Other transparencies indicate in-between scalar data. In addition, the error data are mapped to the rainbow colors. Blue colors indicate relatively small error data, while

red colors indicate relatively big error data. Other colors indicate in-between colors. The implementation of this technique is simple and the resulting image can be directly acquired from the volume ray-casting rendering. Figure 5.13 illustrates an example of the hue technique.

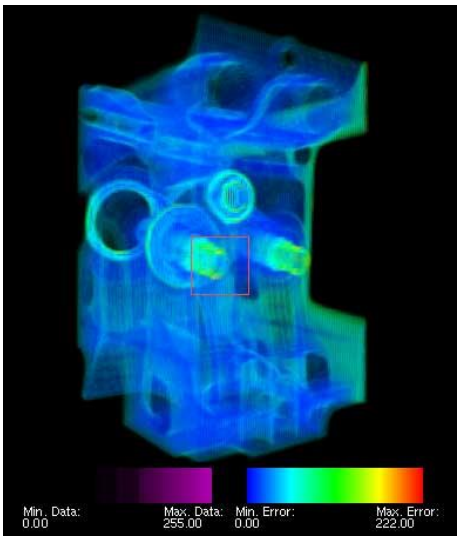


Figure 5.13: The hue technique.

LVIS: this technique corresponds to the proposed uncertainty visualization technique, as described in Section 5.3. Unlike the previous four techniques which only enable the user to qualitatively distinguish both scalar data and their associated error data by certain visual metaphors, this technique is intended to enable the user to quantitatively visualize these data by offering effective feedback in relation to the user's selection.

Figure 5.14 illustrates an example of this LVIS technique, which is composed of four subwindows, with every one being designed for a specific purpose. The upper left subwindow presents the volume rendering of the scalar data, which are mapped to the rainbow colors to indicate their magnitude. Similarly, the upper right subwindow presents the volume rendering of the error data, which are also mapped to the rainbow color to indicate their magnitude. Both subwindows are enabled with the depth sample extraction interaction, as described in Section 5.3.2.3, which has capability of extracting depth sample data from a 2D ROI that is automatically generated by the program. The separation of these two renderings is extremely

meaningful as it could yield a clearer data presentation. Both subwindows together consist of the object space of the 3D data.

As for the bottom left subwindow, it is intended to offer the user a clear feedback about his selection. It plots all scalar data against their associated error data inside the ROI in green. In addition, it draws the user specified scalar data against their associated error data that are inside the ROI and selected based on either of the two upper subwindows in red. In this way, it allows the user to clearly judge whether his selection has achieved a sufficient accuracy for a specific user study task, and thus refine his selection if necessary. However, one drawback of this subwindow is that the scale of its plot is too big to accurately reflect the minimum scalar and error value *i.e.* it is very difficult to identify the minimum scalar and error value in the bottom left region as the data displayed there are quite clustered. Therefore, the bottom right subwindow is included in this technique, which is intended to present a zoom in version of the region specified in the bottom left subwindow to help the user to accurately identify the minimum scalar and error value. The bottom two subwindows together consist of the feature space of the 3D data.

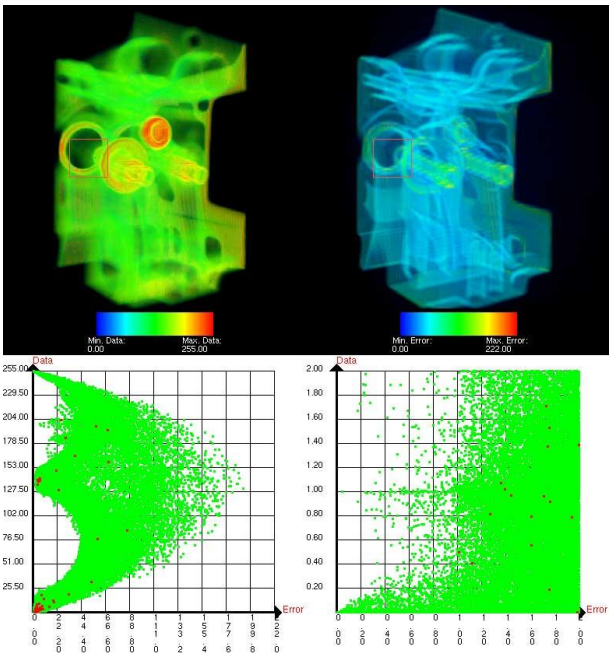


Figure 5.14: The LVIS technique.

5.4.1.3 Participant Pool

In total there are 30 participants in this study, of which 19 are males, and 11 are females. More specifically, there are 6 participants who are researchers in UCC, and there are 3 participants who are staff from our industrial partner. In addition, there are 3 participants who are graduate, design engineer and administrator, respectively in UCC. For the remainder, they are all postgraduate students in UCC. Among all participants, none of them is reported to be color-blind. 28 out of 30 participants claimed that they use graphs and charts for day-to-day activities although this skill is not a prerequisite to participate in this study. Most participants typically use a computer for more than 15 hours per week.

5.4.1.4 User Study Tasks

By consulting Dr. Jamie Dyer, a meteorologist, Sanyal et al. (2009) determined a real-world scenario where uncertainty could be part of his decision making process. They found that he is interested in looking at regions of extreme (big or small) uncertainty while discerning the features in the data. Moreover, they found that he liked to explore the data by looking at the entire data first, and then focusing on a ROI. By keeping this in mind, four types of searching tasks that can simulate the exploratory navigation of the data are designed. These four types of searching tasks are:

• identify the spot where may include the maximum error value in the marked area (as illustrated in Figure 5.15).

• identify the spot where may include the minimum error value (non-zero) in the marked area.

• identify the spot where may include the maximum scalar value in the marked area.

• identify the spot where may include the minimum scalar value (non-zero) in the marked area.

These four types of searching tasks are assigned to every chosen uncertainty visualization technique for the user's performance. Their orders are randomly appeared in the screen to avoid the bias that the user tends to complete the first task more accurately than other tasks. The position of the marked area (as illustrated in Figure 5.15) is also generated randomly for every technique. This is to avoid the case

where the user is able to make an identical selection based on his previous experience by using other uncertainty visualization techniques. Additionally, the user is allowed to select any spot within the marked area and refine his selection subsequently.

In this user study, the accuracies of these techniques with respect to the four types of searching tasks are evaluated by comparing the maximum or minimum scalar or error value of the spot that is specified by the user with the maximum or minimum scalar or error value inside the ROI/marked area. Equation 5.4 to 5.7 list the score criteria for the tasks of identifying the maximum error value, identifying the minimum error value, identifying the maximum scalar value and identifying the minimum scalar value, respectively:

$$emax_{spot} \geq emax_{ROI} - \alpha(emax_{ROI} - emin_{ROI}) \quad (5.4),$$

$$emin_{spot} \leq emin_{ROI} + \beta(emax_{ROI} - emin_{ROI}) \quad (5.5),$$

$$smax_{spot} \geq smax_{ROI} - \gamma(smax_{ROI} - smin_{ROI}) \quad (5.6),$$

$$smin_{spot} \leq smin_{ROI} + \delta(smax_{ROI} - smin_{ROI}) \quad (5.7),$$

where $emax_{spot}$ or $smax_{spot}$ represents the maximum error or scalar value of the spot selected by the user; $emin_{spot}$ or $smin_{spot}$ represents the minimum error or scalar value of the spot selected by the user; $emax_{ROI}$ or $smax_{ROI}$ represents the maximum error or scalar value inside the ROI; $emin_{ROI}$ or $smin_{ROI}$ represents the minimum error or scalar value inside the ROI; α, β, γ and δ are constant coefficients, determined according to data distribution used for evaluation. For the CT data of engine block used in the main study, we chose $20\%, 0.6\%, 12\%$ and 0.05% for α, β, γ and δ, respectively. If the user's selection meets the corresponding criteria, a score of 1 is given, otherwise a score of 0 is given.

One thing worth mentioning is that the maximum and minimum error or scalar values of a spot selected by the user are not simply its pixel value. They are actually obtained by comparing all error or scalar sample values along the ray casted from the spot (a spot is equivalent to a pixel in the final 2D image). This is possible because of the depth sample extraction interaction technique, as described in Section 5.3.2.3.

Similarly, the maximum and minimum error or scalar values of a ROI (which is composed of many pixels) are not simply obtained by comparing its pixel values. They are actually obtained by comparing all error or scalar sample values along the rays casted from the pixels that consist of the ROI.

Another thing worth mentioning is that the data exploited in the user study are from real-world medical CT scan. Typically, this type of data contain a large amount of 0 values which are the minimum value of the data and represent no material or air inside or outside the scanned object. In this case, it is very likely that the minimum scalar or error value of a spot specified by the user is the 0 value. This will result in invalidity of the evaluation for tasks of identifying the minimum scalar or error value, as the user may always obtain a correct answer by selecting any one spot within a ROI. To avoid the invalidity and make sure that the answer is not always correct when the user performs tasks of identifying the minimum scalar or error value, the minimum boundary is adjusted to be non-zero.

5.4.1.5 Interface Design

A simple and user friendly interface is designed to enable the user to perform the 4 searching tasks with respect to the 5 chosen uncertainty visualization techniques, as illustrated in Figure 5.15. It consists of 3 parts which are the main display (upper left), the interaction (right) and the status bar (bottom left), respectively. In the main display, the 5 uncertainty visualization techniques are presented one by one randomly. This is to avoid the bias caused by the display order. In addition, a red rectangle is displayed on top of every uncertainty visualization technique. It indicates the ROI where the user is asked to make selections. In the interaction part, a big text box is arranged at the top. It is used to inform the user the current searching task. In addition, two buttons are arranged at the bottom, and both are set to disabled as default. This is to make sure that the user could not skip any one of these tasks. When the user selected a spot within the ROI, the “Accept” button will be enabled. It is used by the user to confirm the selection. Once it is pressed, two things happen. First, according to the specific task undertaken by the user, for example, the user is asked to select a spot within a ROI for the task of identifying maximum error value, then

the maximum error value of the spot as well as the maximum and minimum error values within the ROI will be recorded into a file for subsequent score measurement (as mentioned in Section 5.4.1.4), which will be used for final statistical analysis (will introduce in Section 5.4.2). As a result, every task of an uncertainty visualization technique corresponds to an individual file. The second thing occurred after the "Accept" button being pressed is that the "Next" button will be enabled. At this point, the user could still revise the answer by reselecting a spot within the ROI and confirming it. However, once the "Next" button is hit, the selection will be done and the next random searching task or uncertainty visualization technique will be presented on the screen. In this status bar, clear prompt is given to guide the user's operation.

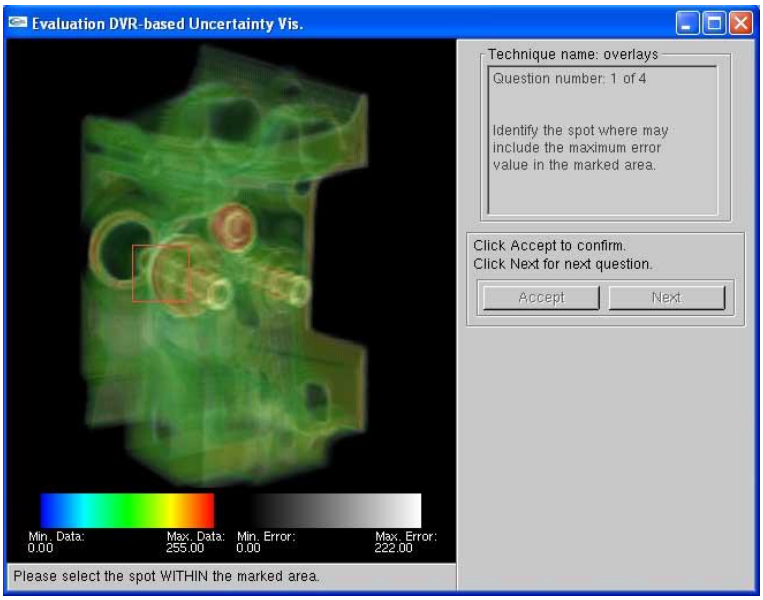


Figure 5.15: The user interface.

5.4.1.6 Participant Training

A training process is given to the user before the main user study. This is to make sure that the user is familiar with the evaluation and feels confident to take on the main user study independently. We typically spent about 5 to 10 minutes to give the user an overview of the user study. This involves getting them understanding the concept of uncertainty visualization, its significance, as well as the purpose of the user study.

After that, every user is assigned to a computer which runs a training module that is similar to the main study, but uses a different data set, which is CT scan of a female cadaver head, with 8 bits for each voxel (Levoy, 2008). Every user is asked to complete the corresponding tasks in the training module. Coupling with this process, we spent about 10 to 15 minutes explaining the user the 5 different uncertainty visualization techniques, the user interface, as well as the 4 types of searching tasks expected from them. We believe that organizing the explanation along with the training module is a better idea than those methods to separate them, because this could provide the user with more living examples that promote the understanding. After the user completed the training, a "question and answer" stage is followed where we will answer the questions that are unclear to the user. Finally, every user will take on the main study independently.

5.4.1.7 The Main Study

The data set used in the main user study is CT scan of an engine block, available at (Roettger, 2012), as illustrated in Figure 5.15. There are in total of 30 participants taking part in the main user study and only 28 participants' feedback is selected for statistical analysis. This is because the other 2 participants seemed unmotivated and completed their tasks in a rush.

To avoid the interplay between participants, the main user study is run on one participant at a time. Every participant is kept in a similar environment to eliminate the impact from the environment. Apart from some personal information, every participant needs to answer a total of 20 questions for the main user study. This formed a complete 5×4 design for the 5 chosen uncertainty visualization techniques with respect to the 4 searching tasks. The response time is recorded in seconds for every question.

5.4.2 Method of Analysis

Both task completion accuracies and task completion time with respect to the 5 evaluated uncertainty visualization techniques are statistically analyzed to find out the significant difference. The classic approach (Bentley, 2009)(Laerd Statistics, 2013c) begins with a two-way full factorial ANOVA, followed by either the individual

one-way ANOVA if there is no significant interaction, or the one-way ANOVA which can capture the Simple Main Effects if there is significant interaction. Standard post-hoc test found by ANOVA is also performed. The statistical analysis is conducted in the SPSS.

Table 5.1: Tests of between-subjects effects on accuracies.

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	33.757[a]	19	1.777	9.334	.000
Intercept	187.457	1	187.457	984.834	.000
techniques	12.239	4	3.060	16.075	.000
tasks	19.486	3	6.495	34.124	.000
techniques * tasks	2.032	12	.169	.890	.557
Error	102.786	540	.190		
Total	324.000	560			
Corrected Total	136.543	559			

a. R Squared = .247 (Adjusted R Squared = .221)

For analysis of the accuracies, the two-way full factorial 5×4 ANOVA is used to test whether there is a significant interaction between the 5 uncertainty visualization techniques and the 4 types of searching tasks. The Table 5.1 generated from SPSS indicates that there is no significant interaction between the 5 uncertainty visualization techniques and the 4 types of searching tasks ($\mathrm{F}(12, 540) = 0.890, \mathrm{p} = 0.557 > 0.05$). This implied that the type of searching tasks did not have an impact on whether there is a significant difference between the 5 techniques or not. Thus, an individual one-way ANOVA is computed to assess whether there is a significant difference between the 5 uncertainty visualization techniques. According to the Table 5.2 from SPSS, it implied that there is a significant difference between the 5 uncertainty visualization techniques$(\mathrm{F}(4, 135) = 14.341, \mathrm{p} < 0.0005)$. Therefore, the analysis is continued with the Bonferroni test to compare all pairwise significant difference (10 pairwise comparisons) of the 5 techniques. The Table 5.6 lists the specific findings which will be discussed in the next section.

Table 5.2: Results for one-way ANOVA on accuracies.

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	48.957	4	12.239	14.341	.000
Within Groups	115.214	135	.853		
Total	164.171	139			

For analysis of the time, a two-way full factorial 5 x 4 ANOVA is computed to assess whether there is a significant interaction between the 5 uncertainty visualization techniques and the 4 types of searching tasks. The Table 5.3 generated from SPSS indicated that there is a significant interaction between the techniques and the user tasks ($\mathrm{F}(12,539) = 4.370, \mathrm{p} < 0.0005$). This implied that whether there is a significant difference between techniques or not, depended on the type of tasks assigned to the participants. Therefore 9 one-way ANOVAs are computed to capture the Simple Main Effects. The first 4 one-way ANOVAs are computed to see whether there is a significant difference between the 5 techniques with respect to each searching task. The Table 5.4 from SPSS indicated that there is a significant difference between the 5 techniques with respect to the completion time of the task of identifying the maximum error ($\mathrm{F}(4,539) = 32.531, \mathrm{p} < 0.0005$). In addition, it indicated that there is a significant difference between the 5 techniques with respect to the completion time of the task of identifying the minimum error ($\mathrm{F}(4,539) = 15.032, \mathrm{p} < 0.0005$). Thus the Bonferroni test is computed (10 pairwise comparisons) to see which of the specific techniques differed. Table 5.7 and Table 5.8 list the specific findings and we discuss the statistical significant results in the next section. The second 5 one-way ANOVAs are computed to check whether there is a significant difference between the 4 types of searching tasks with respect to each uncertainty visualization technique. The Table 5.5 from SPSS indicated that there is a significant difference between the 4 types of searching tasks with respect to the LVIS technique ($\mathrm{F}(3,539) = 20.197, \mathrm{p} < 0.0005$). Thus the Bonferroni test is computed (6 pariwise comparisons) to see which of the specific tasks differed. Table 5.9 lists the specific findings and we discuss it in the next section.

Table 5.3: Tests of between-subjects effects on time.

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	32573.824[a]	19	1714.412	11.536	.000
Intercept	183329.974	1	183329.974	1.234E3	.000
techniques	22876.667	4	5719.167	38.482	.000
tasks	1713.038	3	571.013	3.842	.010
techniques * tasks	7794.084	12	649.507	4.370	.000
Error	80105.364	539	148.618		
Total	295853.000	559			
Corrected Total	112679.188	558			

a. R Squared = .289 (Adjusted R Squared = .264)

Table 5.4: Univariate tests on time from the first 4 one-way ANOVAs.

tasks		Sum of Squares	df	Mean Square	F	Sig.
identify max error	Contrast	19338.757	4	4834.689	32.531	.000
	Error	80105.364	539	148.618		
identify min error	Contrast	8935.929	4	2233.982	15.032	.000
	Error	80105.364	539	148.618		
identify max data	Contrast	1255.757	4	313.939	2.112	.078
	Error	80105.364	539	148.618		
identify min data	Contrast	1311.240	4	327.810	2.206	.067
	Error	80105.364	539	148.618		

Each F tests the simple effects of techniques within each level combination of the other effects shown. These tests are based on the linearly independent pairwise comparisons among the estimated marginal means.

5.4.3 Results and Discussion

Table 5.6 illustrates the statistical significance between the 5 uncertainty visualization techniques in terms of the accuracies. We can see from this table that our new LVIS technique is the most accurate one among all of these evaluated techniques. It is strongly better than the Transparency technique, the Discontinuities technique and the Hue technique. The second most accurate technique is the Overlays technique. It is strongly better than the Transparency technique, and is

better than the Hue technique. For the remainder three techniques, there is no statistical significance between them. But from the table we can see that the Discontinuities technique has the highest average score among the three techniques, while the Transparency technique has the lowest one.

Table 5.5: Univariate tests on time from the second 5 one-way ANOVAs.

techniques		Sum of Squares	df	Mean Square	F	Sig.
Transparency	Contrast	27.500	3	9.167	.062	.980
	Error	80105.364	539	148.618		
Discontinuities	Contrast	291.884	3	97.295	.655	.580
	Error	80105.364	539	148.618		
Overlays	Contrast	93.670	3	31.223	.210	.889
	Error	80105.364	539	148.618		
Hue	Contrast	69.143	3	23.048	.155	.926
	Error	80105.364	539	148.618		
LVIS	Contrast	9005.085	3	3001.695	20.197	.000
	Error	80105.364	539	148.618		

Each F tests the simple effects of tasks within each level combination of the other effects shown. These tests are based on the linearly independent pairwise comparisons among the estimated marginal means.

We believe that the LVIS technique is proved to be the most accurate due to its mechanism of quantitative feedback that makes the user clearly aware the features of inspected data. As for the Overlays technique, it is proved relatively accurate due to its easy visibility for both scalar data and error data in visualization. Likewise, we believe that the remainder three techniques are evaluated as being relatively inferior due to the fact that their visibility for either scalar data or error data, or a combination of them could be less perceived to the user.

Table 5.7 illustrates the statistical significance between the 5 uncertainty visualization techniques in terms of the completion time for the task of identifying the maximum error. We can see from this table that our new LVIS technique took strongly longer than the others to complete this task. For the remainder four techniques, there is no statistical significance between them.

Table 5.6: Statistical significance of accuracies between the 5 uncertainty visualization techniques.

		Transparency	Discontinuities	Overlays	Hue	LVIS
bigger mean(more accurate)	LVIS	**	**		**	
	Overlays	**			*	
↓	Discontinuities					
smaller mean(less accurate)	Hue					
	Transparency					

A * indicates significance at the 0.05 level.
A ** indicates strong significance at the 0.025 level.

We believe that the LVIS technique is proved to be more time-consuming due to its capability of exploratory analysis and refining. Likewise, the remainder four techniques are proved to be less time-consuming due to the fact that they fail to provide the user with such a capability, but only the visual metaphors.

Table 5.7: Statistical significance of the completion time between the 5 uncertainty visualization techniques for the task of identifying the maximum error.

		Transparency	Discontinuities	Overlays	Hue	LVIS
bigger mean(more time)	LVIS	**	**	**	**	
	Discontinuities					
↓	Hue					
smaller mean(less time)	Transparency					
	Overlays					

A * indicates significance at the 0.05 level.
A ** indicates strong significance at the 0.025 level.

Table 5.8 illustrates the statistical significance results between the 5 uncertainty visualization techniques on the completion time for the task of identifying the minimum error. The same conclusion and explanation for Table 5.7 can be given here. However, the difference between Table 5.7 and Table 5.8 is the listed order of the Overlays technique, the Transparency technique and the Hue technique. In Table 5.7, we can see that the Hue technique took the longest time among the three techniques,

while the Overlays technique took the shortest. But in Table 5.8, we can see that the consumed time between the Hue technique and the Overlays technique has been reversed, while the Transparency technique remains the same.

Table 5.8: Statistical significance of the completion time between the 5 uncertainty visualization techniques for the task of identifying the minimum error.

		Transparency	Discontinuities	Overlays	Hue	LVIS
bigger mean(more time)	LVIS	**	**	**	**	
↓	Discontinuities					
	Overlays					
smaller mean(less time)	Transparency					
	Hue					

A * indicates significance at the 0.05 level.
A ** indicates strong significance at the 0.025 level.

Table 5.9: Statistical significance of the completion time between the 4 types of searching tasks with respect to the LVIS technique.

		Identify the max. error data	Identify the min. error data	Identify the max. scalar data	Identify the min. scalar data
bigger mean(more time)	Identify the max. error data		*	**	**
↓	Identify the min. error data			**	**
	Identify the min. scalar data				
smaller mean(less time)	Identify the max. scalar data				

A * indicates significance at the 0.05 level.
A ** indicates strong significance at the 0.025 level.

Table 5.9 illustrates the statistical significance results between the 4 types of searching tasks in terms of the completion time based on the LVIS technique. We can

see from this table that the task of identifying the maximum error data consumed the longest time. It consumed strongly longer than the tasks of identifying either the maximum scalar data or the minimum scalar data. In addition, it consumed longer than the task of identifying the minimum error data. The task that took the second longest time is the one of identifying the minimum error data. It took strongly longer than both the task of identifying the maximum scalar data and the task of identifying the minimum scalar data. As for the remainder two tasks, there is no statistical significance between them.

5.5 Evaluation of Execution Time

Apart from the perceptual effectiveness, we also measure execution time for each of the 5 evaluated uncertainty visualization techniques, which are run on the engine data set (Roettger, 2012), as illustrated in Table 5.10.

Table 5.10: Execution time of the 5 evaluated uncertainty visualization techniques.

Uncertainty visualization techniques	**Execution time (in seconds)**
Transparency	39.14
Discontinuities	38.953
Overlays	38.843
Hue	38.922
LVIS	39.89

It is clear from this table that all the 5 evaluated uncertainty visualization techniques have a very similar execution time, although the LVIS uncertainty visualization technique takes slightly longer than others to execute. This is probably because it takes more time to draw its data.

5.6 ELVIS Uncertainty Visualization Technique for Multivariate 3D Scalar Data

In addition to univariate 3D scalar data, the multivariate 3D scalar data that are often generated from instrumental measurements and many scientific simulations may also incorporate uncertainty. However, very little research has been reported to visualize uncertainty in multivariate 3D scalar data. This is because of the big

challenge caused by it: simply visualize multivariate 3D scalar data (also known as multifield visualization), in itself, a recognized challenge to the visualization community (Johnson, 2004). Taking their uncertainties into account will make this visualization problem even harder, as there may not be enough visualization channels available to depict so many variables (both multivariate and their uncertainties). Also, the resulting visualization often suffers from classic visualization problems such as visual clutter and occlusion.

From the above-mentioned evaluation work it is clear that the proposed LVIS is a very effective uncertainty visualization technique, as it enables users to more accurately search both extreme (max. and min.) scalar and error data than other 4 commonly used techniques. Therefore, we determined to further extend it to address the need of visualizing uncertainty in multivariate 3D scalar data, and we call this extended LVIS work as ELVIS, which corresponds to the content in (Ma et al., 2012d).

5.6.1 Overview of Functionalities

Based on the important industrial experience as described in (Doleisch et al., 2005), the proposed uncertainty visualization consists of three main functionalities to enable the user to interactively and quantitatively explore and analyze multivariate scalar data and their errors. The first functionality is denoted by the red dotted lines in Figure 3.16, and it allows the user to inspect the statistical information and correlations of the multivariate scalar data and their errors with respect to certain ROI of the spatial domain. The second functionality is denoted by the yellow dotted lines in Figure 3.16, and it allows the user to investigate the features of multivariate scalar data and their errors by specifying a feature of a scalar data and their errors in one scatter plot, and concurrently analyzing the features with respect to the remaining scalar data and their errors in other scatter plots. The third functionality is knows as feature localization, denoted by the blue dotted lines in Figure 3.16. It allows the user to search for places in the 3D spatial domain where certain features that have been specified are presented. In the next three sections, the components of every functionality will be presented in great details.

5.6.2 Functionality 1: Visualization of Statistical Information and Correlations

This section introduces all the components that consist of the first functionality, which allow the user to inspect the statistical information and correlations of multivariate scalar data and their errors with respect to certain ROI of the spatial domain.

5.6.2.1 Scalar/Error Data Views

This type of views is intended to give the user quick understanding and insight about the scalar or error data being inspecting. They are all rendered by the classic volume ray casting algorithm, which is capable of revealing both external and internal structures of data. Due to the complexity of multivariate scalar data and their errors, presenting them into a single view often causes typical visualization problems such as visual clutter or occlusion. Thus, to obtain a much clearer data presentation, they are displayed into multiple views, with every view corresponding to a singular scalar or its error data. In addition, to facilitate a visual comparison, every error data view is arranged side by side to its corresponding scalar data view. Moreover, the user is allowed to interact with any one of these view for a ROI specification. Once a ROI has been specified in one view, it will be indicated by a white marked area that is synchronously appeared in the same position of all the views.

5.6.2.2 Scalar/Error Histogram Views

To give the user quantitative insight about the statistical information of the 3D sample data inside a specified ROI, the scalar/error histogram views are developed in the proposed uncertainty visualization. Every histogram view only presents the statistical results for a singular scalar or its error data for a clear data presentation. Its horizontal axis depicts the interval information of a specific scalar or its error data, while its vertical axis counts the number of sample data inside a ROI falling into every particular interval.

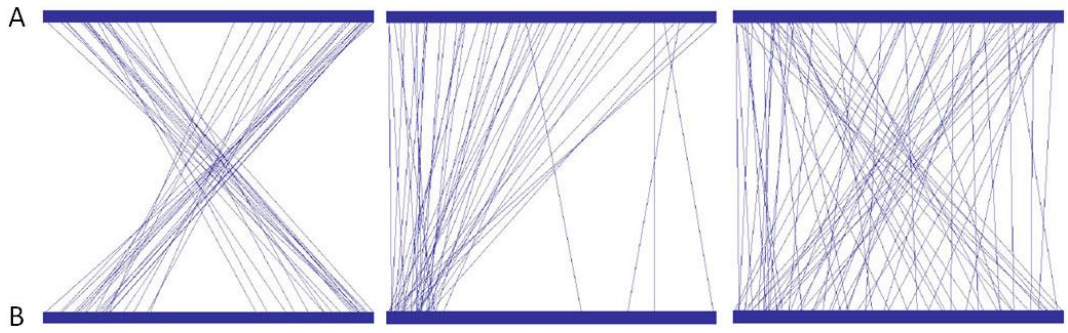


Figure 5.16: Typical correlations that could be shown in a PC. (a) Shows a negative correlations between A and B. (b) Shows a strong correlation between A and the low-value range of B. (c) Shows no obvious correlation between A and B (Akiba and Ma, 2007).

5.6.2.3 Parallel Coordinates View

PC (Inselberg, 1985)(Inselberg and Dimsdale, 1990)(Inselberg, 2009) is a popular technique created to visualize multivariate data. Its basic form is a 2D display plotting data using parallel axes, one for each variable. It generally can provide a good overview of the correlation among multivariate. Thus, it is included in the proposed uncertainty visualization to offer the user the correlation of multivariate scalar data and their errors that are specified inside a ROI. In particular, to facilitate the user to inspect the correlation between a scalar and its associated error data, every scalar parallel axis is arranged adjacent to its associated error parallel axis. By observing the patterns between two neighboring axes of a PC, the correlation between the corresponding variables can be detected. Figure 5.16 illustrates some typical correlations of a PC.

5.6.2.4 Linking between Views and Depth Sample Extraction

While every component mentioned above has certain utility, linking them together could form a more powerful visualization system that allows quantitative exploration and analysis of the multivariate scalar data and their error data inside certain ROI specified in the spatial domain. A typical workflow of functionality 1 can be summarized as follows: The user starts the exploration by inspecting the scalar/error data views in the spatial domain and dragging a rectangle through a mouse interaction on any of these views to specify a ROI. He can repeatedly modify this ROI

until he feels satisfied. Once a ROI has been specified, two things occur. First, a white marked area will synchronously appear in the same position of all scalar/error data views to denote the ROI. Second, the “relevant” multivariate scalar data and their errors inside the ROI will be extracted from the memory and passed to the scalar/error histogram views and PC view to generate the corresponding graphs for subsequent analysis. A key here is what “relevant” data we should extract and use so that they can represent the entire characteristics of the ROI. Again, like described in Section 5.3.2.3, the 3D samples of multivariate scalar data and their errors are extracted from the memory through the depth feature extraction technique and used in the scalar/error histogram views and PC view, as they are the data utilized for the ROI rendering and thus can represent its entire characteristics. As a result, we enable users to analyze the 3D samples’ scalar and error values within the specified 2D ROI in the spatial domain.

5.6.3 Functionality 2: Visualization of Data Features

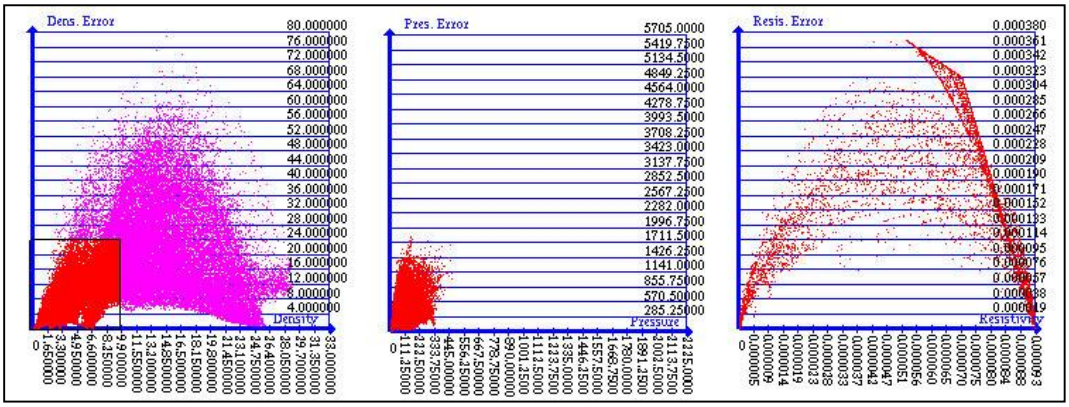
The second functionality consists of multiple linked scatter plot views, which allow the user to investigate the features of multivariate scalar data and their errors by specifying a feature of a univariate scalar data and their errors in one scatter plot, and concurrently analyzing the features with respect to the remaining scalar data and their errors in other scatter plots.

5.6.3.1 Scatter Plot Views

For interactive features or ROIs specification, multiple scatter plots (Wolfram Research Inc., 2014) are exploited in the proposed uncertainty visualization. Every scatter plot view plots a univariate scalar variable against its error variable for a clear data presentation. It serves as both feature display and ROI specification tool, where the user can observe the features and specify a ROI by dragging a rectangle on it through a mouse interaction. In addition, every scatter plot view is assigned to a unique color that is used to indicate its ROI as well as those features that match its ROI shown in other scatter plot views. In this way, the user can identify which features match certain ROI by simply observing whether they have the same color.

5.6.3.2 Linking between Views

These scatter plot views are tightly linked by user interaction for the ROI specification. A typical workflow of them can be described as follows: the user starts his exploration by inspecting the features of multivariate scalar data and their errors based on the scatter plot views and specifying a ROI on any one of these views. He could repeatedly modify his ROI until he feels satisfied. Two things occur once a ROI has been specified. One thing is that the view used for the ROI specification will be updated automatically to denote the ROI with the unique color assigned to this view. Another thing is that the remaining views that have not been used for a ROI specification will be automatically updated as well, to show the corresponding features that match the specified ROI with the same color. Here an example is taken to explain how to acquire the corresponding features of a ROI, as illustrated in Figures 5.17(a), (b) and (c). We can see from these figures that the multivariate data consist of 3 scalar data (density, pressure and resistivity) and their associated errors. Thus there are in total of 3 scatter plot views, with each corresponding to a univaraite scalar and its errors. From Figure 5.17(a) we can see that a red ROI is specified for the density scalar and its errors, and Figures 5.17(b) and (c) show its corresponding features. The method to obtain the two features shown in Figures 5.17(b) and (c) can be explained as follows: first, the entire multivariate data is searched and those voxels whose both density values (here the density values and the following mentioned pressure and resistivity values are from the simulation data as described in Section 5.6.5.1) and their associated error values matched the specified ROI are identified. Second, the pressure values and their associated error values at the identified voxels are extracted and plotted in Figure 5.17(b). Also, the resistivity values and their associated error values at the identified voxels are extracted and plotted in Figure 5.17(c). Consequently the corresponding features that matched the specified ROI are acquired.



(a) *(b)* *(c)*

Figure 5.17: An example about how to acquire the corresponding features of a ROI for multivariate data.

The user can continue his exploration by repeating the above operation in those views which have not been used for a ROI specification to formulate more complicated and compound ROI specification. A specific example of this case is shown in Figure 5.18. As a result, the user is enabled to investigate the features of multivariate scalar data and their errors in an interactive and quantitative way.

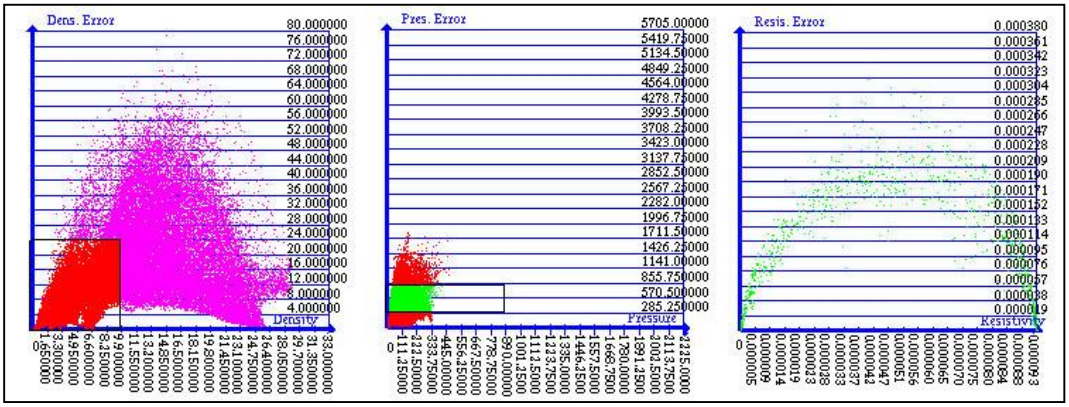


Figure 5.18: An example of a compound ROI specification.

5.6.4 Functionality 3: Feature Localization

The third functionality is composed of two types of views, which are tightly linked together to allow searching for places in the 3D spatial domain where certain specified features are presented.

5.6.4.1 Logical Scatter Plot Views

To allow the user to specify features whose 3D spatial positions are of interest in exploration, multiple scatter plots are again utilized. Similar to the scatter plot views mentioned in the second functionality, every scatter plot in this functionality plots a singular scalar variable against its error variable for a clear data presentation. Also, it could be used by the user for a ROI specification. However, they have some differences from the previously mentioned ones. First, a ROI specification performed in any one of these scatter plot views will not update and affect any other scatter plot views. Only the view used for this ROI specification will be updated to denote the ROI. Every scatter plot always presents a complete feature with respect to a singular scalar and its error data for the user's inspection. Second, all scatter plot views share the same color for the ROI specification. This color is also used for the features rendering in the feature localization view (will be introduced in Section 5.6.4.2) so that the user can determine which ROIs correspond to the localized features. Third, these linked scatter plot views in fact serve as the TFs for the feature localization view. In other words, the features specified in them will control appearance of the final data displayed in feature localization view. Last, they offer two logical operations "AND" and "OR" for feature extraction. Thus, we refer to them as logical scatter plot views, to make a clear distinction with those scatter plot views previously mentioned in Functionality 2.

AND Operation: the idea to enable the logical scatter plot views with logical operations is from the work of Gasser (2004), who introduced the logical operations into his visualization for complex selection. Two simple logical operations AND and OR have been implemented in this research, which serve as different purpose. The AND operation is defined as an operation that can extract those voxels which concurrently match multiple criteria specified by multiple ROIs in the logical scatter plot views. It is very useful for many cases. For example, assuming our multivariate data are about the prediction of temperature and pressure of a cold room, and we are interested in those positions which meet both conditions as specified in formulas 5.8 and 5.9:

$$\begin{cases} -18°C \leq temperature \leq -10°C \\ 0°C \leq temperature.error \leq 1°C \end{cases} \quad (5.8),$$

$$\begin{cases} 0.05MPa \leq pressure \leq 0.15MPa \\ 0.0001MPa \leq pressure.error \leq 0.0005MPa \end{cases} \quad (5.9),$$

where temperature.error refers to the error data associated to the temperature data; pressure.error refers to the error data associated to the pressure data; MPa refers to megapascal. This can be easily achieved by specifying two ROIs corresponding to the two conditions in two appropriate logical scatter plot views and enabling the AND operation. Consequently the corresponding positions in the spatial domain will be revealed in the feature localization view.

OR Operation: the OR operation is defined as an operation that can extract those voxels which match any one of the criteria specified by ROIs in the logical scatter plot views. It is very useful for those cases where it is interested to inspect all positions where as long as a feature appears. Continuing with the cold room example described above, we are interested in those positions which meet any one condition among 4.4 and 4.5. This can be also easily achieved by specifying the same two ROIs as those mentioned in the AND operation. But this time, we need to enable the OR operation instead. Consequently the corresponding positions with respect to the OR operation will be appeared in the feature localization view.

5.6.4.2 Feature Localization View

The feature localization view is used to give the user immediate feedback about the spatial positions of those features specified in the logical scatter plot views. It is rendered by the volume ray-casting algorithm. The color used to render the features in this view is kept as the same as the color used for ROI specification in the logical scatter plot views. In this way, the user can identify which ROIs have been used to localize the features.

5.6.4.3 Linking between Views

The logical scatter plot views and the feature localization view are tightly linked by the user interaction for ROI specification. A typical workflow can be described as follows: the user starts his exploration of feature localization by inspecting the data features in the logical scatter plot views, and specifying a ROI on any one of those views. He could modify the ROI specification repeatedly until he feels satisfied. Once

a ROI is specified, two things happen. One thing is that the view used for the ROI specification will be updated to denote it. Another thing is that the feature localization view will be immediately updated to show the spatial positions of the features match the ROI. The user can repeat the above operation in other logical scatter plot views to formulate a more complicated and compound ROI specification, and based on the AND or OR operation selected, the feature localization view will present the corresponding feature positions for his inspection.

5.6.5 Case Study

To demonstrate the application of the proposed uncertainty visualization, a case study is conducted. Section 5.6.5.1 introduces the original data set and the pre-processing applied to the original data sets to generate the test data sets. Section 5.6.5.2 presents the corresponding results after applying the proposed uncertainty visualization to the test data sets.

5.6.5.1 Original Data Set and Pre-processing

Three data sets are exploited in this case study and they are generously offered by Mr. Andrew Foulks from University of New Hampshire. They are a sample (at time step 900 seconds from the start of the simulation) of the entire data source which is generated from the Open Geospace General Circulation Model (OpenGGCM) simulation performed at Space Science Research Center in University of New Hampshire, and used to study the phenomenon of solar wind and its interaction with the earth's magnetosphere. Three scalar variables including density, pressure and resistivity (Community Coordinated Modeling Center, 2007) can be available from this simulation, with every scalar being sampled on the Cartesian grid (Raeder, 1995) and corresponding to an individual data set. The dimensions of every data set are $1024 \times 512 \times 512$, and its size is 1.05GB. In addition, its format is 32-bit IEEE floating point values with little endian (Lin, 2003).

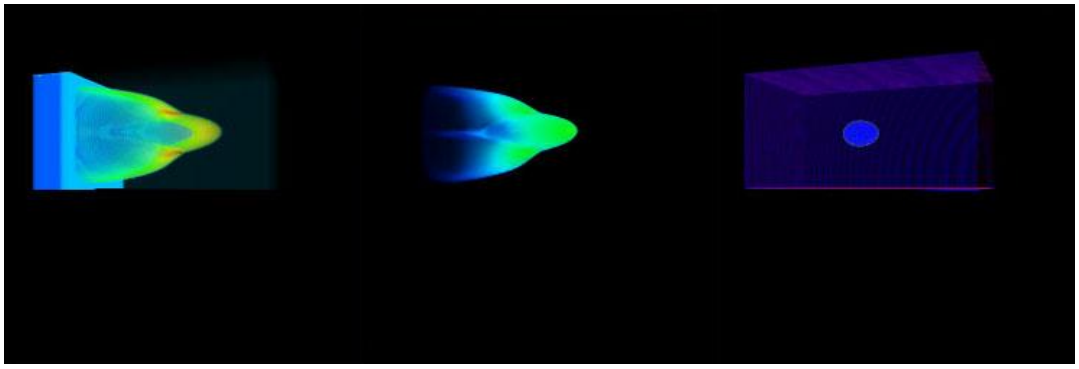
As we focus on quantitative uncertainty visualization of multivariate data rather than visualization of large-scalar data, the three original data sets are reduced by the MR modeling techniques (as introduced in Section 4.2) to three smaller derived data sets whose dimensions are $256 \times 128 \times 128$ that can be handleable by the memory.

These derived data sets are then processed by two successive MR modeling to obtain its two lower resolution data sets that are used to test the proposed visualization.

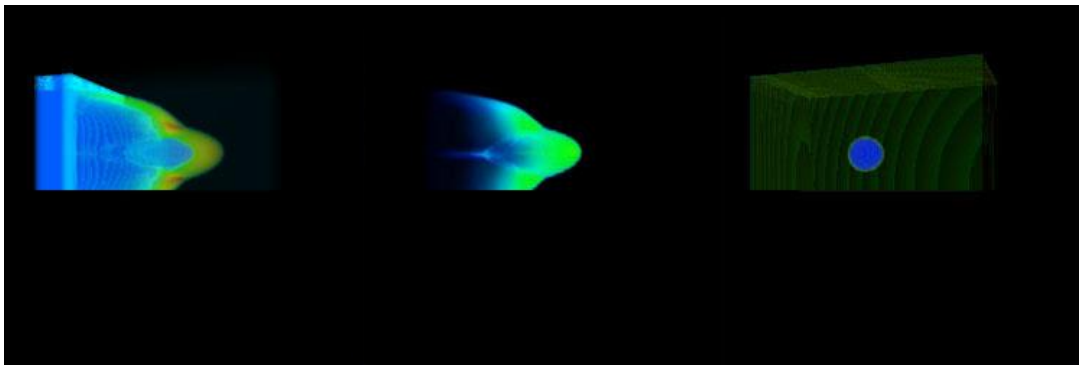
Figures 5.19(a), (b) and (c) illustrates the corresponding MR hierarchy acquired from the Haar Wavelet Transform. The left column three images correspond to the density data. The middle column three images correspond to the pressure data, and the right column three images correspond to the resistivity data. We can see that the three scalar variables shown in Figure 5.19(a) have the greatest detail, and they correspond to the three derived data sets whose dimensions are $256 \times 128 \times 128$. Compared to Figure 5.19(a), the three scalar variables shown in Figure 5.19(b) are slightly coarser *i.e.* the edges of the density and pressure data appear a little bit coarse, which indicates some details have been lost. They correspond to the MR data after one time Haar Wavelet Transform and their dimensions are $128 \times 64 \times 64$. Figure 5.19(c) shows the coarsest version of the three scalar variables and we can see from this figure that a substantial amount of details have been lost. They correspond to the MR data after two times Haar Wavelet Transform and their dimensions are $64 \times 32 \times 32$. Figures 5.19(d), (e) and (f) show the three TFs utilized to render the density data, pressure data and resistivity data, respectively. For easy reference of these MR data sets in the subsequent paragraphs, we refer to the $256 \times 128 \times 128$ data set as R_0, the $128 \times 64 \times 64$ data set as R_1, and the $64 \times 32 \times 32$ data set as R_2.

5.6.5.2 Results and Discussion

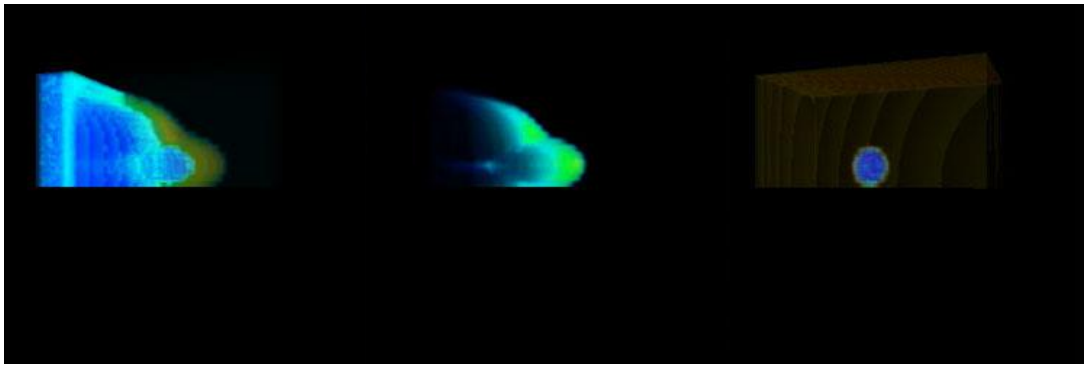
Three questions are expected to be ansared by the proposed uncertainty visualization after applying it to the MR data sets. The first question is: “what are the statistical characteristics and correlations among multivariate scalar variables and their error variables within a subset of the spatial domain?” The second question is: “what is the feature of the multivariate scalars and their associated error data set after its certain scalars and their associated errors’ features have been specified?” The third question is: “where certain features specified in the feature space will appear in the spatial domain?”



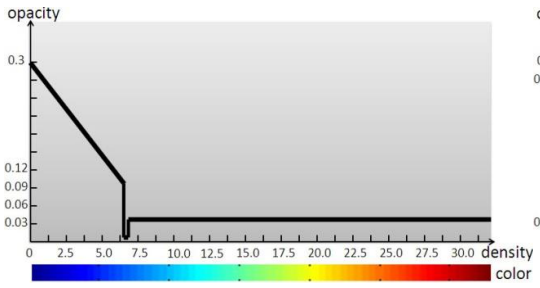
(a)



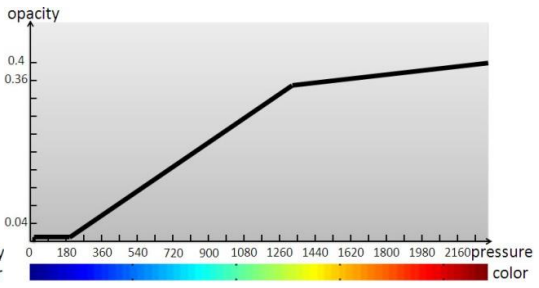
(b)



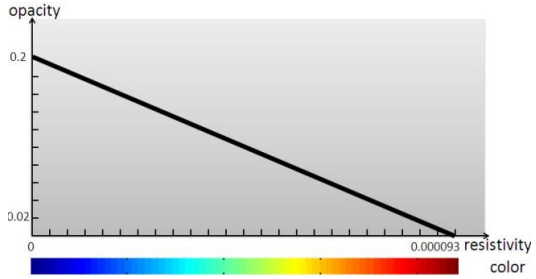
(c)


opacity
0.3
0.12
0.09
0.06
0.03
0 2.5 5.0 7.5 10.0 12.5 15.0 17.5 20.0 22.5 25.0 27.5 30.0 density
color

(d)


opacity
0.4
0.36
0.04
0 180 360 540 720 900 1080 1260 1440 1620 1800 1980 2160 pressure
color

(e)



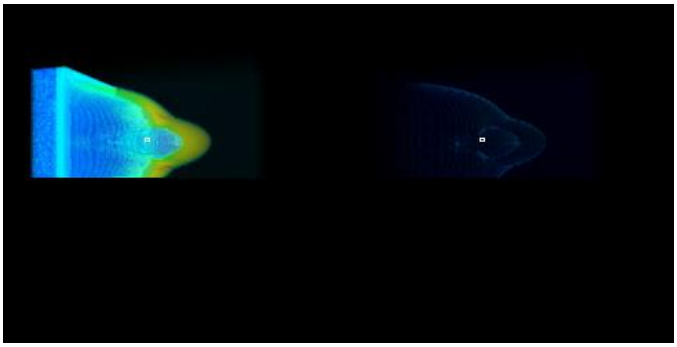
(f)

Figure 5.19: A MR hierarchy for the density variable (left column), the pressure variable (middle column) and the resistivity variable (right column) of the OpenGGCM simulation, generated from the Haar Wavelet Transform.

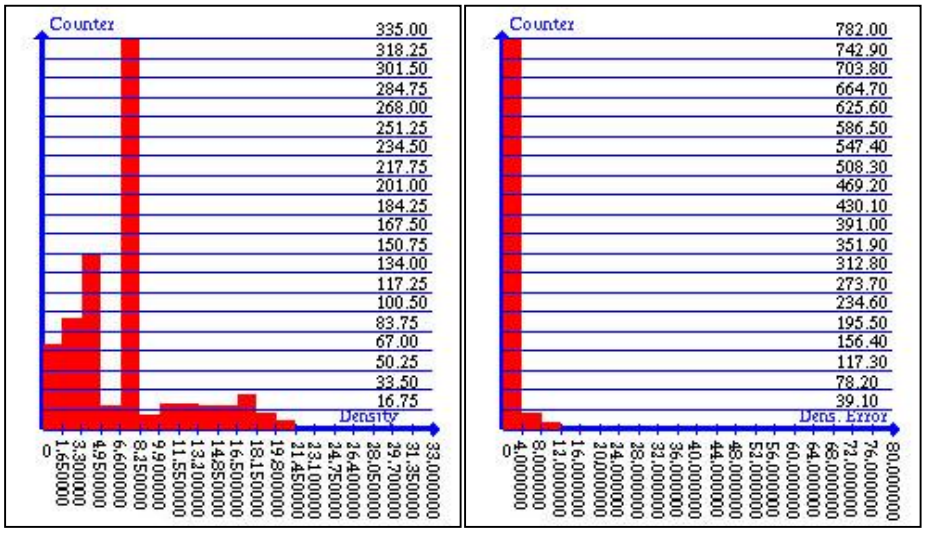
Figure 5.20 shows the results corresponding to the first question and the MR data set used here is R_1. The three images in the leftmost column correspond to the scalar data views which rendered the density (Figure 5.20(a1)), pressure (Figure 5.20(b1)) and resistivity (Figure 5.20(c1)) data, respectively. The ones in the left middle column correspond to the error data views, which rendered the error data associated to the density (Figure 5.20(a2)), pressure (Figure 5.20(b2)) and resistivity data (Figure 5.20(c2)), respectively. The ones in the right middle column correspond to the scalar histogram views, which present the statistical information (the number of scalar data samples falling into every interval) of density (Figure 5.20(a3)), pressure (Figure 5.20(b3)) and resistivity data (Figure 5.20(c3)), respectively that is inside a ROI specified by the user. The ones in the rightmost column correspond to the error histogram views, which present the statistical information (the number of error data samples falling into every interval) of error data that are associated to the density (Figure 5.20(a4)), pressure (Figure 5.20(b4)) and resistivity data (Figure 5.20(c4)), respectively inside the same ROI as the one for the scalar histogram views. Finally, Figure 5.20(d) corresponds to the PC view of the proposed uncertainty visualization.

We can see from the scalar/error data views that a small white ROI has been specified in the spatial domain. From these scalar/error histogram views it is very easy to draw the following statistical conclusions regarding the three scalar variables and their associated errors inside the ROI: the values of the density data inside the ROI are continuously distributed between 0 and 21.45, which occupy more than 50% of the entire data space, while the values of their error data are continuously distributed

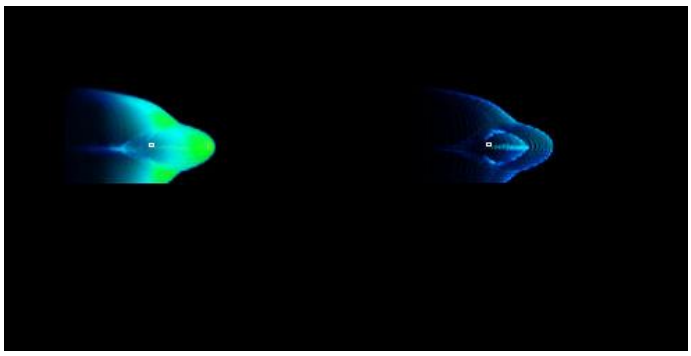
between 0 and 12, which only occupy a very small percentage of the entire data space. In addition, most of the density data fall into the interval $(6.6, 8.25]$, and least of them fall into the interval $(19.8, 21.45]$. While most of their error data are concentrated in the interval $(0, 4]$, and least of their error data fall into the interval $(8, 12]$. As for the pressure data inside the ROI, their values are continuously distributed between 0 and 667.5, which occupy less than 50% of the entire data space, while the values of their error are continuously distributed between 0 and 570.5, which only occupy a very small percentage of the entire data space. Additionally, most of the pressure data fall into the interval $(0, 111.25]$, and least of them fall into the interval $(556.25, 667.5]$. While most of their error data are concentrated in the interval $(0, 285.25]$, and least of their error fall into the interval $(285.25,\ 570.5]$. In terms of the resistivity data inside the ROI, their values are continuously distributed between 0 and 0.000093, which occupy the entire data space, while the values of their error data are discretely distributed between 0 and 0.000133 (no values between $(0.000095, 0.000114]$), which occupy nearly 50% of the entire data space. Moreover, most of the resistivity data fall into the interval $(0.000088, 0.000093]$, and least of them fall into three intervals concurrently that are $(0.000009, 0.000014]$, $(0.000037, 0.000042]$ and $(0.000051, 0.000056]$, respectively. While most of their error data are concentrated in the interval $(0, 0.000019]$, and least of them fall into the interval $(0.000076, 0.000095]$. The statistical information offers valuable analysis about the characteristics of the three scalar variables and their error data inside the ROI. Furthermore, their correlations can be quickly revealed by the PC. We can see from Figure 5.20(d) that there is a strong correlation between the low-middle-value range of density and the low-value range of its errors; there is a strong correlation between the low-value range of density's errors and the low-value range of pressure; there is a strong correlation between the low-value range of pressure and the low-value range of its errors; there is strong correlation between the low-value range of pressure's errors and resistivity; there is a strong correlation between resistivity and the low-value range of its errors.



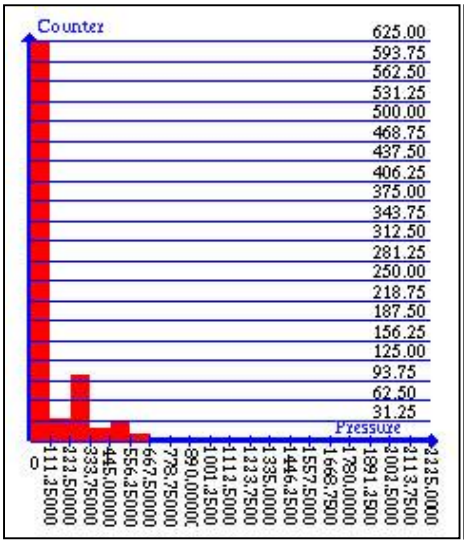
(a1) *(a2)*

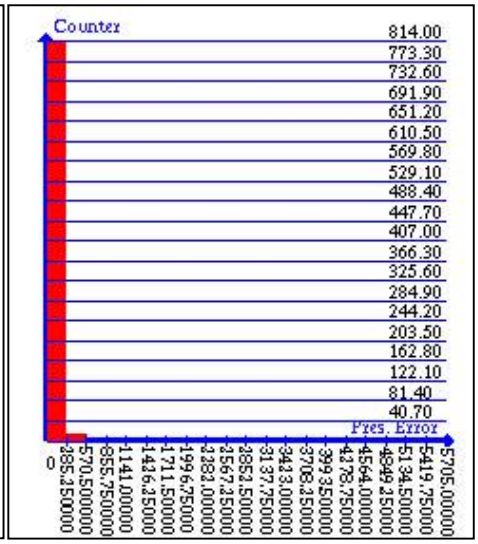


(a3) *(a4)*

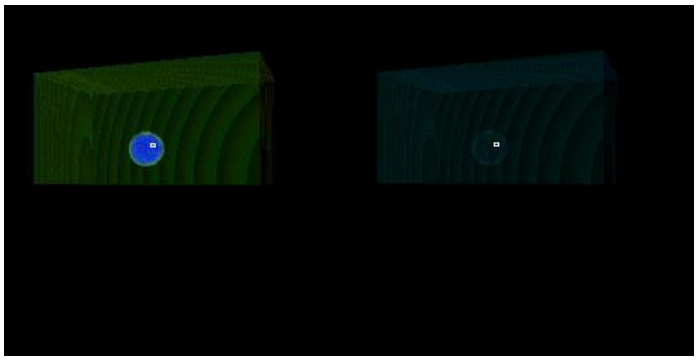


(b1) *(b2)*

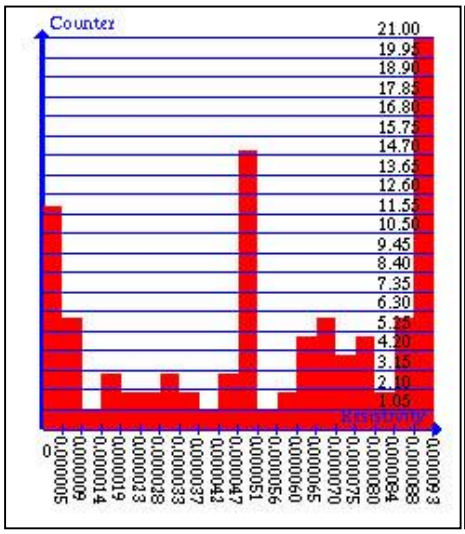


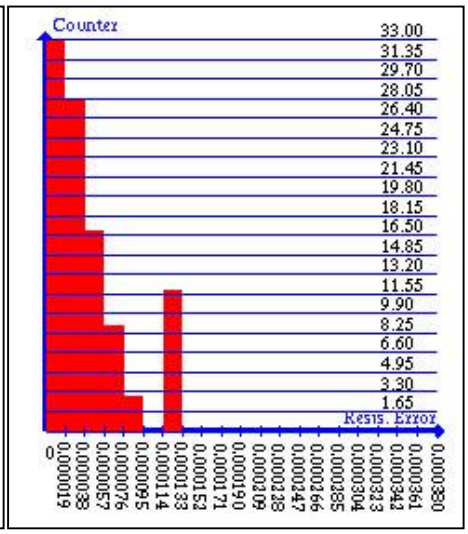


(b3) *(b4)*

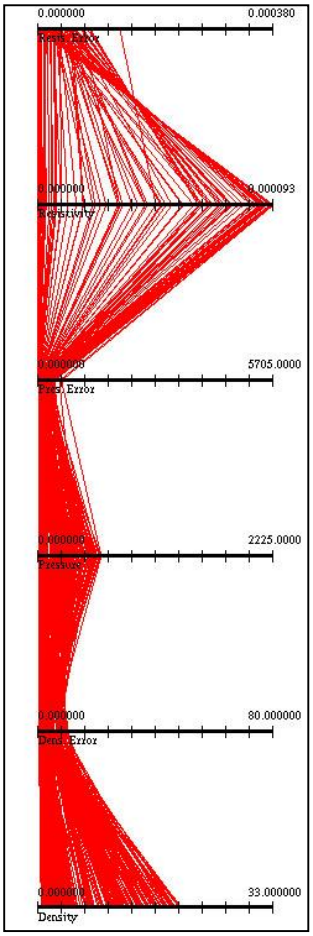


(c1) *(c2)*





(c3) *(c4)*



(d)

Figure 5.20: The results after a ROI of the spatial domain is specified.

Figure 5.21 shows the results with respect to the second question and the data set used here is R_2. Figures 5.21(a1), (a2) and (a3) show the initial features with respect to the three scalar variables and their associated errors of the multivariate data. In particular, Figure 5.21(a1) corresponds to the initial feature with respect to the density variable and its associated errors. Figure 5.21(a2) corresponds to the initial feature with respect to the pressure variable and its associated errors. Figure 5.21(a3) corresponds to the initial feature with respect to the resistivity variable and its associated errors. Figures 5.21(b2) and (b3) show the features with respect to the pressure variable and its associated errors as well as the resistivity variable and its associated errors, after the feature of the density variable and its associated errors ($\begin{cases} 0 \leq density \leq 9.9 \\ 0 \leq density.error \leq 24 \end{cases}$) has been specified, as highlighted in red in Figure 5.21(b1). In contrast to the features shown in Figures 5.21(a2) and (a3), we can see

from these two figures that the feature with respect to the pressure variable and its associated errors has been reduced dramatically, while the feature with respect to the resistivity variable and its associated errors remains a very similar pattern.

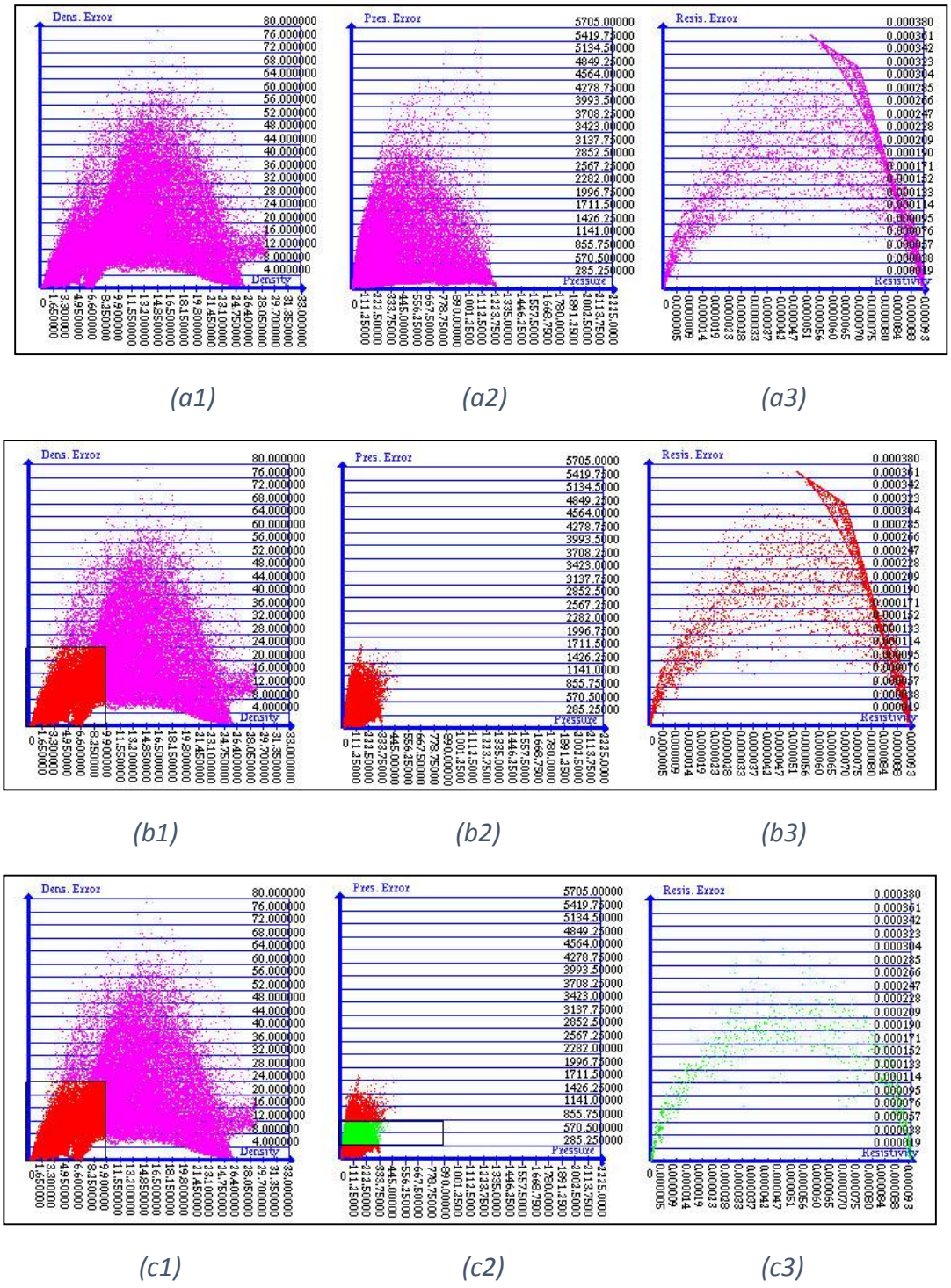


Figure 5.21: The results after the multivariate data's certain features are specified.

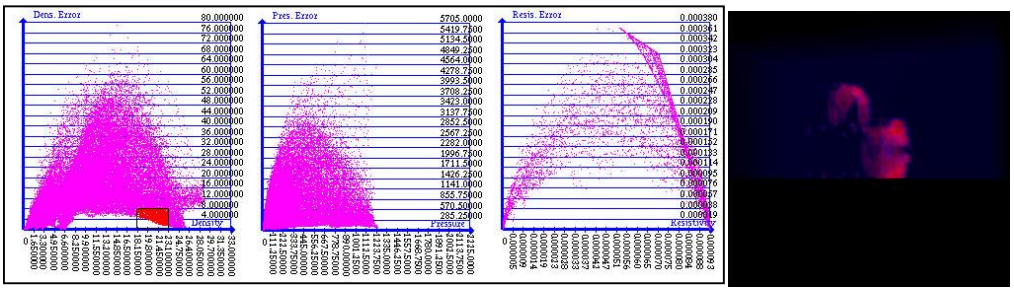
Figure 5.21(c3) illustrates the feature with respect to the resistivity variable and its associated errors of the multivariate data, after the features regarding their density

variable and its associated errors (the same as the one shown in Figure 5.21(b1)) as well as their pressure variable and its associated errors ($\begin{cases} 0 \leq pressure \leq 89 \\ 285.25 \leq pressure.error \leq 855.75 \end{cases}$) have been specified, as illustrated in Figures 5.21(c1) and (c2). In contrast to the feature shown in Figure 5.21(b3), we can see that the one shown in this figure consists of less data and lost some clustering patterns *i.e.* the clustering pattern appeared in the right-hand side.

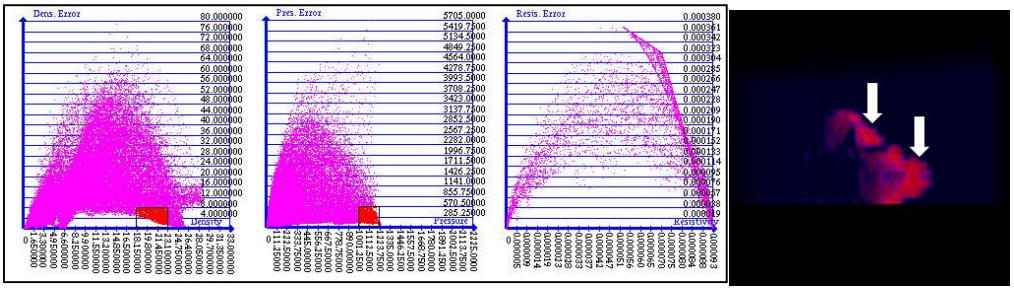
Figure 5.22 shows the results regarding the third question with the OR operation being enabled, and the data set used here is R_2. The images in the left three columns correspond to the logical scatter plot views, which present the multivariate features with respect to the density variable and its associated errors, the pressure variable and its associated errors, and the resistivity variable and its associated errors, respectively and used for feature specification. The three images in the rightmost column correspond to the feature localization view. We can see from the logical scatter plot views in Figure 5.22(a) that a feature regarding the density variable and its associated errors ($\begin{cases} 18.15 \leq density \leq 23.1 \\ 0 \leq density.error \leq 8 \end{cases}$) has been specified, and its corresponding spatial positions are clearly presented in the feature localization view in Figure 5.22(a). These positions are in fact those voxels' positions which match this feature. Similarly by observing the logical scatter plot views in Figure 5.22(b), we can see that in addition to the feature just mentioned, another feature regarding the pressure variable and its associated errors ($\begin{cases} 1001.25 \leq pressure \leq 1223.75 \\ 0 \leq pressure.error \leq 570.5 \end{cases}$) has been specified, and their corresponding spatial positions are illustrated in the feature localization view in Figure 5.22(b). These positions are in fact those voxels' positions which match either the first or the second feature, or both of them. In contrast to the feature localization view in Figure 5.22(a), we can see that the one in Figure 5.22(b) shows some additional substances *i.e.* the substances indicated by the arrows of this image. They imply the positions of those voxels which only match the second feature. By observing the logical scatter plot views in Figure 5.22(c) we can see that a third feature with respect to the resistivity variable and its associated errors ($\begin{cases} 0 \leq resistivity \leq 0.000023 \\ 0 \leq resistivity.error \leq 0.000247 \end{cases}$) has been specified in addition to the two

features just mentioned, and their corresponding spatial positions are shown in the feature localization view in Figure 5.22(c). These positions are in fact those voxels' positions which match the logical OR operation of all the three features.

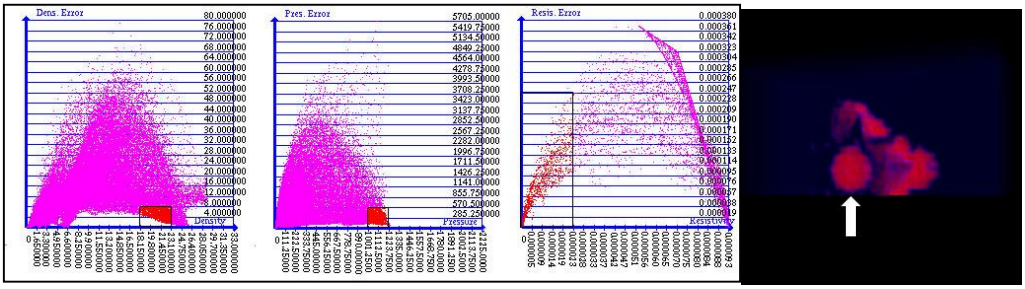
In contrast to the feature localization view in Figure 5.22(b), we can see that the one in Figure 5.22(c) shows an additional sphere substance, as indicated by the arrow. It corresponds to the positions of those voxels which only match the third feature.



(a)



(b)



(c)

Figure 5.22: The results after the multivariate data's certain features are specified in the feature space with the OR operation being enabled.

5.7 Summary

This chapter presented two relevant works, which take advantage of linked views and interactive specification to visualize errors associated to DVR. The first work is

the LVIS uncertainty visualization technique, which has fully implemented the three main steps of our framework and is designed to enable users to more accurately visualize errors in univariate 3D scalar data. The analysis results from our evaluation have validated that it is the most accurate technique among all evaluated uncertainty visualization techniques, although it took the longest time. As for those existing techniques, the Overlays technique appears to be more accurate. These findings can provide useful guidance for future uncertainty visualization design.

The second element of work explored is the ELVIS uncertainty visualization technique, which is extended from the LVIS technique and designed to enable users to effectively explore and analyze multivariate 3D scalar data and their associated errors. Such an extension is very meaningful to the field of uncertainty visualization, as little research has been addressed to visualize uncertainty in multivariate 3D scalar data. We show the relevant results by applying it to a specific case study. To validate its effectiveness, a further evaluation work is needed. This is considered as future work, which will be discussed in Chapter 7.

So far, our research only focuses on visualizing uncertainty generated from the data transformation that occurs as early as the data acquisition stage (see Section 1.1.1.1 for detail), in either isosurface rendering (corresponding to Chapter 4) or DVR (corresponding to this chapter). However, as shown in Figure 1.1, even the visualization itself may also include uncertainty. Uncertainty of visualization (see Section 1.1.1.1 for detail) will result in the visualization running in risks that mislead viewers' interpretation of data, or to make incorrect decisions from the data. In the next chapter, we will present relevant work to address such a problem.

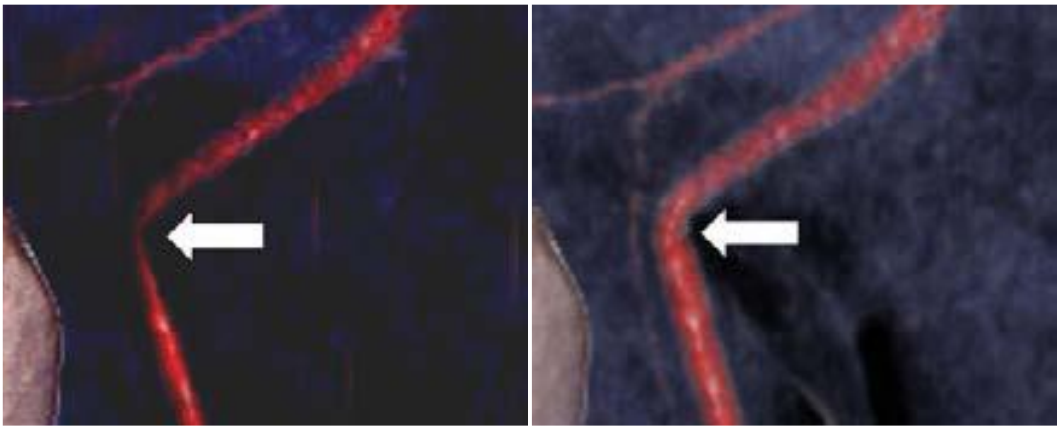
Chapter 6 Uncertainty Visualization of Material Intensity Classification in DVR Using Probabilistic Query Method

6.1 Introduction

This chapter will present a work in relation to the Probabilistic Query uncertainty visualization technique (hypothesis 3), as illustrated in Figure 3.13, and it corresponds to the content in (Ma et al., 2013c).

Modern visualization techniques have been successfully applied to the field of medical imaging and one of the typical examples is the DVR, which has in recent years matured into a routinely used tool for diagnosis (Lundstrom et al., 2007)(Lundstrom, 2007). Nevertheless, the DVR can sometimes be insufficient for diagnosis as some challenges still exist in its research. A specific challenge among them is the material intensity classification uncertainty involved in the DVR. This is caused probably because of two main reasons. First, different patients may have variations in their material intensity (Lundstrom et al., 2007)(Lundstrom, 2007). Second, different materials may share the same intensity. Thus for a specific diagnosis task, there is no a certain classification scheme that suits all patients. As a result of this challenge, for a specific diagnosis of a patient, different images that are based on different classification schemes may be generated from the DVR, which can have dramatic influences to the diagnostic decision. One example of this is the stenosis diagnosis in Figure 6.1, where a patient has to undergo a surgery when observe the image (as illustrated in Figure 6.1(a)) that is generated based on one classification scheme. But

when observe the image (as illustrated in Figure 6.1(b)) that is generated based on another classification scheme, we can see that the patient does not have vessel stenosis and thus does not need surgery.



(a) *(b)*

Figure 6.1: (a) A rendering based on one classification scheme indicates there is a stenosis. (b) Another rendering using another classification scheme reveals there is no vessel stenosis (Lundstrom et al., 2007)(Lundstrom, 2007).

Therefore, there is a need for radiologists and physicians to explore the material intensity classification uncertainty to study several possible image outputs in order to make a final diagnostic decision. In traditional DVR, this exploration of the material intensity classification uncertainty is typically achieved by repeated manual TF adjustments. However, such a method has two main drawbacks. First, the manual TF adjustments are performed by crudely perturbing the visualization parameters (which refer to color $\mathrm{RGB}\alpha$) in an uncontrollable manner (Lundstrom et al., 2007)(Lundstrom, 2007). Even for a diagnosis that has moderately complex TFs, it is unlikely for the radiologists and physicians to explore all its visualization parameter space. This may result in some alternative renderings that are very important to the diagnostic decision being missed. Second, there is no quantifiable information about the intensity classification uncertainty associated to each material that is appeared on the final images. As a result, when the radiologists and physicians try to make a diagnostic decision based on different output images from the DVR, they may not be able to determine which image should be used.

To address the above-mentioned two problems so as to improve the DVR as a better decision support tool, we performed the Probabilistic Query uncertainty visualization research, which consists of two main parts. First, the explicitly probabilistic TF model will be introduced, which defines the uncertainty domain as material intensity classification probability and enriches the original 3D scalar data to become the classified 3D scalar data (this process is shown in Figure 3.17) for subsequent visualization process. Second, the Probabilistic Query uncertainty visualization technique will be presented, which extracts the corresponding material features specified by a singular or compound query from the classified 3D scalar data and render them as a final image for diagnosis.

6.2 Explicitly Probabilistic TFs

In the context of this work, two main views on TFs can be distinguished. This first view refers to those TFs which directly map an intensity sample value s to a material color $\mathbf{c}$ that includes four-component vector $\mathrm{RGB}\alpha$, as described in Section 3.1.2.2. These TFs have typically mixed material intensity classification and appearance mapping together (Kniss et al., 2005). This is the currently dominating view introduced in most DVR research (Hansen and Johnson, 2004)(Engel et al., 2006). As such TFs do not explicitly involve any material intensity classification probability, we call them implicitly probabilistic TFs. The second view refers to the TFs which explicitly involve the process of mapping an intensity sample value s to a set of material probabilities. They are exploited by a small number of DVR research (Lundstrom et al., 2007)(Lundstrom, 2007)(Drebin, Carpenter and Hanrahan, 1988). To distinguish them with the implicitly probabilistic TFs, we call them explicitly probabilistic TFs.

The TF model exploited in this work belongs to the explicitly probabilistic TFs, which is originally proposed by (Lundstrom et al., 2007)(Lundstrom, 2007) and only used for material intensity classification - that is, the appearance mapping is kept separately from the explicitly probabilistic TFs. This is different from the TFs that belong to the first view, as mentioned above. It can be defined as a two-step approach. First, the user defines the classifying function $\tilde{p}_m(s)$ which maps different intensity values to

different material likelihood (note that an intensity value can be mapped to more than one material likelihood) according to their experience to the original 3D scalar data to be classified, through the familiar TF GUI interface, as illustrated in Figure 6.2. The range of the likelihood for each material is [0.0, 1.0] and it indicates how likely the intensity values to be this material. A simple form of the classifying function is trapezoids, as illustrated in Figure 6.2.

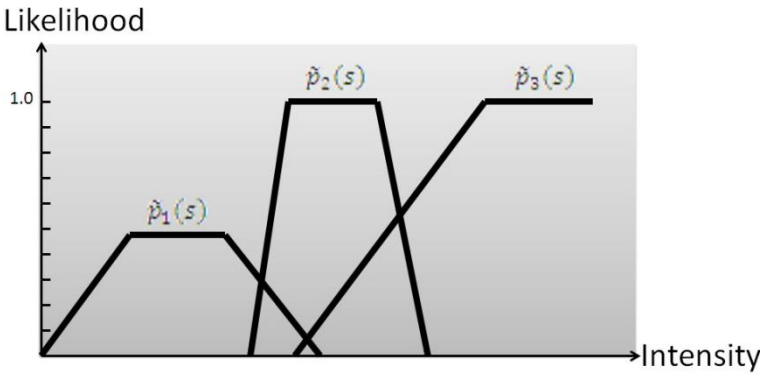


Figure 6.2: An explicitly probabilistic TF, where the material likelihood is defined using trapezoids in the familiar TF GUI interface.

Second, all material likelihoods specified for the intensity values need to be normalized so that their sum should be 1. We call the normalized material likelihood as material probability and formula 6.1 presents the normalization:

$$p_m(s) = \frac{\tilde{p}_m(s)}{\sum_{m'=0}^{M} \tilde{p}_{m'}(s)} \quad (6.1),$$

where $p_m(s)$ represents material m's probability for intensity s. $\tilde{p}_m(s)$ represents material m's likelihood specified by the user for intensity s. $\sum_{m'=0}^{M} \tilde{p}_{m'}(s)$ represents the sum of all M materials' likelihoods for intensity s. In particular, transparent regions refer to null material or air (m = 0) and its likelihood is implicitly defined as formula 6.2:

$$\tilde{p}_0(s) = \max(0.0, 1.0 - \sum_{m=1}^{M} \tilde{p}_m(s)) \quad (6.2).$$

As a result of the explicitly probabilistic TF model, the original univariate 3D scalar data are enriched to the classified 3D scalar data, with its each intensity-specific voxel corresponding to one or multiple material probability.

Figure 6.3 shows a specific example about how a concrete explicitly probabilistic TF is used to compute the material probability and enriches an original univariate 3D scalar data set to a classified 3D scalar data set. Figure 6.3(a) illustrates the original univariate 3D scalar data set with its each voxel including a specific scalar value *i.e.* the voxel indicated by the red rectangle incorporates a scalar value 120. Figure 6.3(b) illustrates the concrete explicitly probabilistic TF that is specified by the user in the TF GUI (corresponding to the first step mentioned above) and used to compute the material probability (corresponding to the second step mentioned above) associated to each intensity-specific voxel of the original univariate 3D scalar data set. Take those voxels whose scalar value equals to 120 as example, the process of computing their corresponding material probability can be summarized as follows: (1) from the explicitly probabilistic TF we can see that the intensity scalar value 120 corresponds to two materials A and B, and both material likelihoods $\tilde{p}_A(120)$ and $\tilde{p}_B(120)$ are 0.6. Their corresponding null material likelihood $\tilde{p}_0(120)$ can be computed using formula 6.2, as illustrated in formula 6.3:

$$\tilde{p}_0(120) = \max(0.0, 1.0 - (\tilde{p}_A(120) + \tilde{p}_B(120))) \quad (6.3),$$

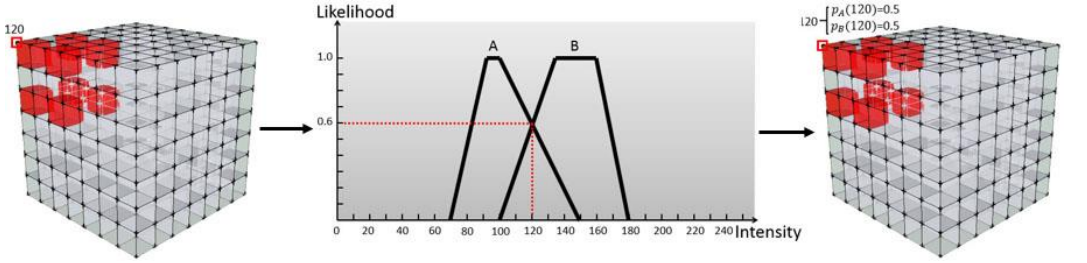
and the result equals to 0.0. (2) given that $\tilde{p}_A(120)$, $\tilde{p}_B(120)$ and $\tilde{p}_0(120)$, we can use formula 6.1 to compute their corresponding material A's probability $p_A(120)$ and material B's probability $p_B(120)$, as illustrated in formula 6.4 and 6.5:

$$p_A(120) = \frac{\tilde{p}_A(120)}{\tilde{p}_A(120)+\tilde{p}_B(120)+\tilde{p}_0(120)} \quad (6.4),$$

$$p_B(120) = \frac{\tilde{p}_B(120)}{\tilde{p}_A(120)+\tilde{p}_B(120)+\tilde{p}_0(120)} \quad (6.5),$$

and the results of $p_A(120)$ and $p_B(120)$ are 0.5.

Therefore, for any voxels whose intensity scalar value equals to 120 in the original univariate 3D scalar data set, they correspond to two materials A and B, and both $p_A(120)$ and $p_B(120)$ are 0.5. This forms the classified 3D scalar data set with enriched material probability information, as illustrated in Figure 6.3(c).



(a) *(b)* *(c)*

Figure 6.3: A specific example about how a concrete explicitly probabilistic TF is used to compute the material probability and enriches an original univariate 3D scalar data set to a classified 3D scalar data set.

6.3 Probabilistic Query

This section will present the Probabilistic Query uncertainty visualization technique, which enables users to extract and display the features of materials according to their classification probability through customizable and interactive query. Its input data are the classified 3D scalar data set generated from the explicitly probabilistic TFs.

Query-based visualization is not a totally new concept in the visualization community, and its basic idea is derived from the database query. However, there is a subtle difference between the two. The query-based visualization typically searches the whole data space and assigns every voxel of the data with appropriate appearance properties according to the specified query, and finally returns an image that indicates the feature to be extracted. In comparison, the database query typically searches the whole database and returns the matched records. Over the past few years, some research has been conducted for query-based visualization. Doleisch, Gasser and Hauser (2003) described a feature querying framework that allows the user to obtain insight from computational simulation data. Johnson and Huang (2009) presented a See Distribution Query (SeeDQ) framework that can readily apply to arbitrary spatial data sets and support on time-variant and multivariate data to extract the features previously inaccessible in general feature detection tools. The difference between their works and our work is that we explore the query-based visualization in the context of uncertainty visualization.

6.3.1 Overview of Workflow

Figure 6.4 shows the workflow (simplified from Figure 3.17 and only focuses on the uncertainty visualization main step rather than the entire framework) of the proposed Probabilistic Query uncertainty visualization technique, given that the classified 3D scalar data set. We can see from the figure that in total of two steps are involved in the visualization. First, the user needs to specify a either singular or compound query (this include information about which materials, under what material classification probability criteria, etc.) to tell what voxels (or features) he or she wants to extract and visualize from the classified 3D scalar data set. Additionally, the user needs to specify the concrete colors for the singular or compound query. As a result, each voxel of the classified 3D scalar data set is assigned with an appropriate color RGBα according to the user specified query and query colors. This forms the colored 3D scalar data set. Second, the volume ray-casting rendering (described in Section 3.1.2.2) is applied to the colored 3D scalar data set to generate the final 2D image that displays the user expected voxels (or features).

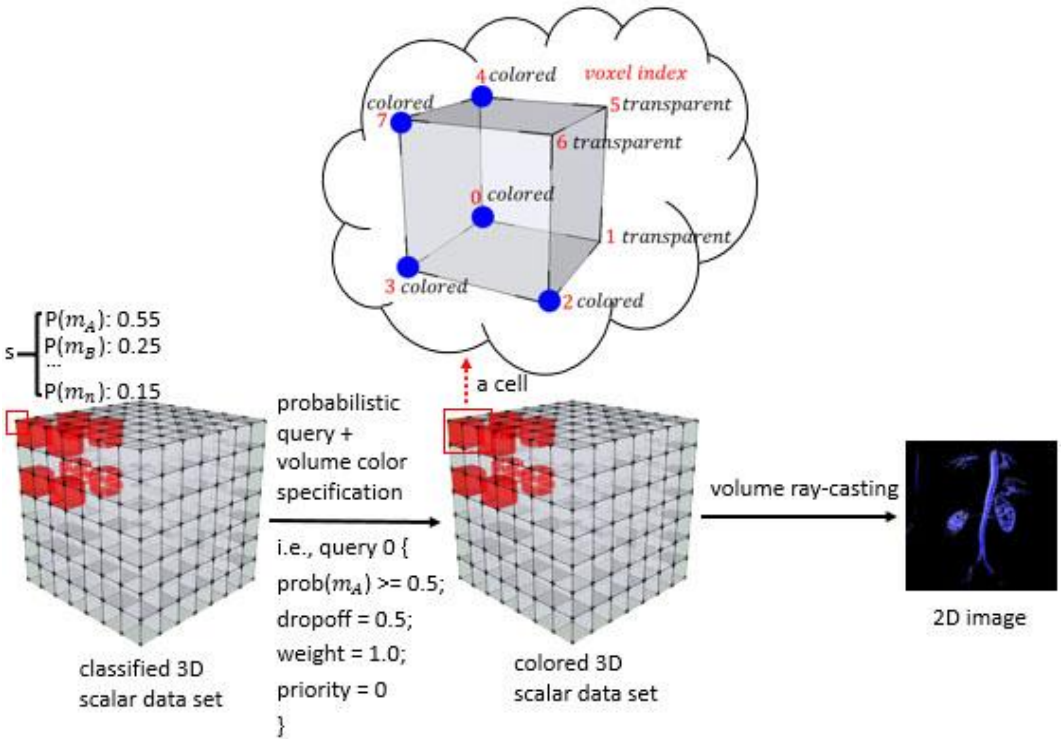


Figure 6.4: Overview of the Probabilistic Query workflow.

6.3.2 Query Statements

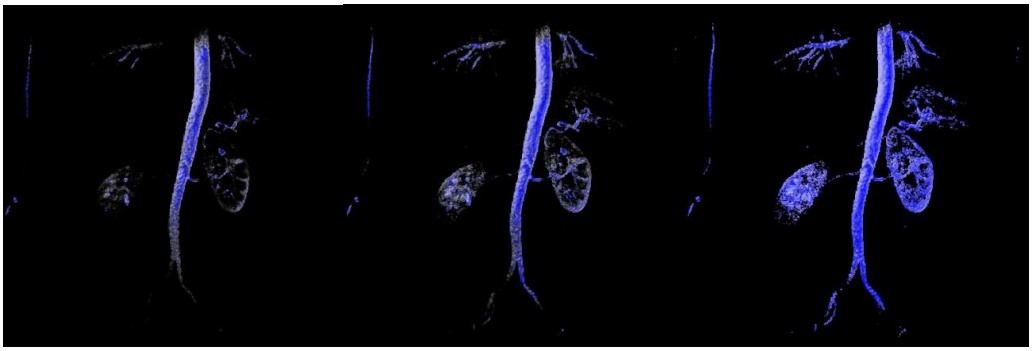
To query a classified 3D scalar data that is derived from an explicitly probabilistic TF, a specific query has to be formed first. This section will introduce four query statements which are used together to form a query. These query statements are designed in a way to facilitate the feature extraction and exploration, as well as give more insight. For any of them, it should be specified by the user according to his application needs. The final query can be used in a singular or compound form on the classified 3D scalar data to extract the expected features.

6.3.2.1 Probability Inequality

The first statement of a query is the probability inequality, as shown in Figures 6.5(a), which is used to set the probability criteria of an interested feature to be extracted. It involves three parameters to be specified by the user. On the left-hand side of the inequality, the user needs to specify the material interested to query. It could be any one material existing in the currently explored data set. On the right-hand side of the inequality, the user needs to specify a target value of probability that is interested to explore for the queried material. This could be as simple as a constant value, or as complicated as an arithmetic expression that mixes multiple material variables' probabilities with constant values to formulate more complex target. Finally, the user needs to specify an inequality operator that indicates how the queried material's probability relates to the target's probability. All standard inequality operators are supported. Also, the $==$ operator is supported to extract a material's corresponding feature whose probability exactly equals to the targeted one specified on the right-hand side of the probability inequality.

The user can quantitatively explore the classification probabilities of a queried material, for a given explicitly probabilistic TF by adjusting the three parameters. As a result of the statement, the corresponding voxels which meet the specified probability inequality will be extracted from 3D data. For example, $prob(m_1) \geq 0.9$ will extract those voxels whose material m_1's probabilities are greater than or equal to 0.9. $prob(m_1) \geq prob(m_2) \times 2$ will find those voxels whose material m_1's probability is greater than or equal to twice than material m_2's probability. One thing is worth mentioning that for queries with $<$ or $\leq$ inequality operators *i.e.*

$prob(m_1) \leq 0.3$, a default constraint condition $prob(m_1) > 0.0$ will be automatically added to the queries. This is to make sure that our queries make sense, as we only want to query the voxels which have possibility to incorporate the queried material, rather than those voxels which do not include the queried material i.e. $prob(m_1) == 0.0$.



query 0 {
prob(vessel) ≥ 0.975;
dropoff = 0.5;
weight = 0.7;
priority = 0
}

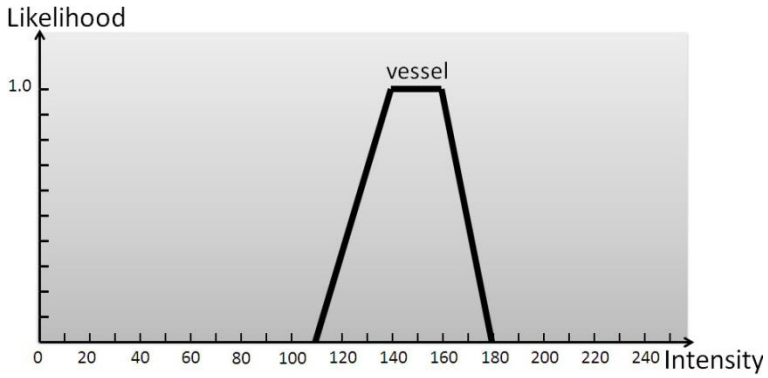
(a)

query 1 {
prob(vessel) ≥ 0. 5;
dropoff = 0.5;
weight = 0.7;
priority = 0
}

(b)

query 2 {
prob(vessel) ≥ 0. 033;
dropoff = 0.5;
weight = 0.7;
priority = 0
}

(c)



(d)

Figure 6.5: The effects of three singular queries on the MR kidney data set with different target probability.

Figure 6.5 illustrates an example of quantitative exploration of classification probabilities for a single vessel material by adjusting target probability of the probability inequality statement, and Figure 6.5(d) shows its corresponding explicitly probabilistic TF. Three specified queries with different target probabilities 0.975, 0.5

and 0.033 have been made for this example, as shown in the bottom three text boxes in Figures 6.5(a), (b) and (c), and the top three images illustrate their corresponding results. In each image, the voxels whose vessel material's probabilities are greater than or equal to the target probability are displayed, and from the three images we can see that with the target probability becoming smaller (query criteria becoming looser), more voxels are rendered as vessels *i.e.* the two kidneys in Figure 6.5(c) are rendered as vessels (the reason that kidneys can be rendered as vessels is because both materials have overlapping intensities, and it is not possible to separate these two materials based on the intensities).

6.3.2.2 Approximate Matching

Displaying more voxels rather than those that strictly match every probability inequality often leads to more insights into the data, and in this way, users could receive feedback even for queries that return few voxels. In addition, users may be interested in those voxels which meet a query to a certain degree. A binary cutoff between matching and nonmatching voxels does not allow users to specify fuzzy queries that achieve above-mentioned goals, and the resulting visualization may have annoyingly sharp edges. Thus, to enable fuzzy queries and improve users' feedback, a fuzzy matching statement is supported into every query.

The implementation of the fuzzy matching can be summarized as follows: first of all, the probability inequality statement of a query is evaluated, which returns a value that indicates how far a voxel's probability relevant to the queried material is from satisfying the inequality. The value is called the *distance* of the inequality, notated as Δ. If the inequality is completely satisfied, the Δ is 0. Otherwise, the Δ is calculated as the absolute value of the difference between the left-hand side and the right-hand side of the probability inequality. This can be mathematically formulated as formula 6.6:

$$\Delta = \begin{cases} 0, & inequality\ is\ met \\ \left| prob_{v_i}(m) - prob(target) \right|, & inequality\ is\ not\ met \end{cases} \quad (6.6),$$

where v_i represents the i^{th} voxel of a 3D data. m represents the queried material. $prob_{v_i}(m)$ represents the material m's probability in voxel v_i. $prob(target)$ represents the target probability specified by the user. Take the query in Figure 6.5(b) as an example, if a voxel does comprise greater than or equal to 0.5 of the vessel material, the Δ is assigned to 0. If it only comprises 0.3 of the vessel material, then the Δ is assign to $|0.3 - 0.5| = 0.2$.

Second, the Δ together with the *dropoff* parameter (see the text box in Figure 6.5(b) for an example) that is specified by the user for the query are then fed into an exponential decay function, as shown in the following formula 6.7, which generates a fuzzy matching score for the query i:

$$score_i = e^{\frac{(-1)}{dropoff_i^2} \times \Delta} \quad (6.7),$$

where $score_i$ represents the fuzzy matching score for query i. $dropoff_i$ represents the *dropoff* parameter for query i, and it indicates how loosely the query can be matched.

The reason to use formula 6.7 to express the fuzzy matching score rather than to use a linear function is because it could offer better control over more various scoring. In addition, the formula is defined over $[0, +\infty)$ which can be used for any possible Δ, and it has a score range of $[0,1]$ that can be mapped directly to the opacity value of every voxel to determine its appearance or disappearance. A final benefit of the formula is that its scores can be adjusted by various *dropoff* parameters. That is to say, by specifying different *dropoff* parameters, the loosely match degree of a query can be controlled. Figure 6.6 shows an experiment about how different scores can be acquired by various *dropoff* values, and we can see from this figure that large *dropoff* parameters allow more loosely matched voxels being displayed, while small ones only allow strictly matched voxels being displayed. The last thing is worth mentioning that the *dropoff* parameter in formula 6.7 is squared, and in this way a magnification effect can be achieved *i.e.* large *dropoff* parameters will become larger, while small ones will become smaller.

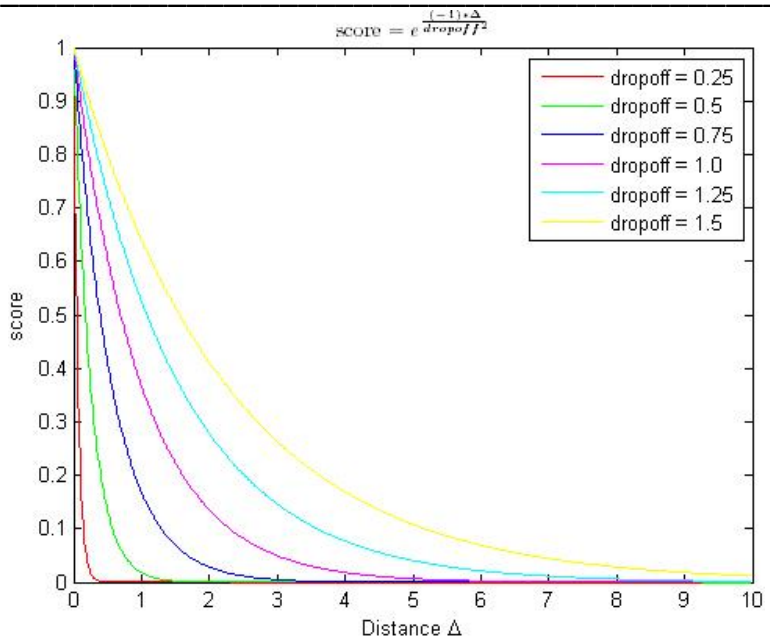
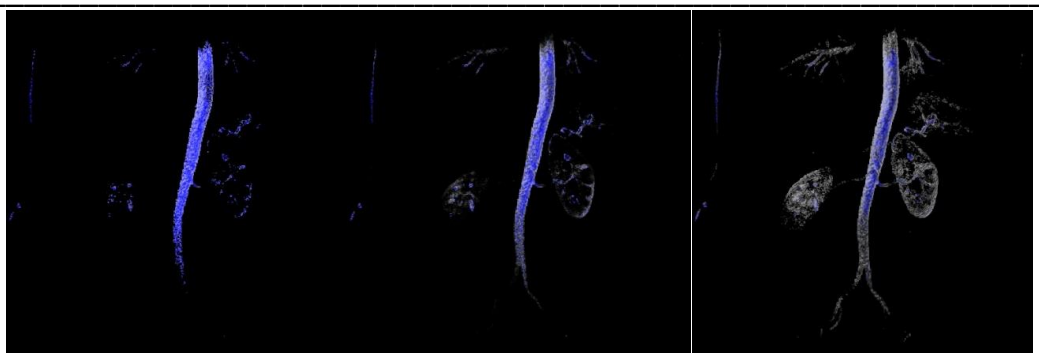


Figure 6.6: The fuzzy matching scores with different dropoff parameters. A large dropoff parameter offers a looser match, while a small one offers a more accurate match.

Figure 6.7 illustrates an example about the fuzzy matching effect, and its corresponding explicitly probabilistic TF is shown in Figure 6.5(d). This example is composed of three queries with the same parameters but the *dropoff*, as shown in the three text boxes in Figures 6.7(a), (b) and (c), and their resulting visualization are shown in the top three images. We can see that in contrast to the query in Figure 6.7(a), the one in Figure 6.7(b) has a bigger *dropoff* value 0.5, which indicates a looser match, is allowed. Thus, the image shown in Figure 6.7(b) has more voxels being rendered in gray (here we assigned blue to voxels that exactly meet the probability inequality, and we assigned gray to voxels that meet the fuzzy matching, but not meet the probability inequality. In this way, we allow the user to make a distinction between voxels depending on whether they meet the exactly or only fuzzy matching. Section 6.3.3 will introduce the details of query color specification) than the ones shown in Figure 6.7(a). Similarly, we can see that in contrast to the query in Figure 6.7(b), the one in Figure 6.7(c) has a looser *dropoff* value 1.75. Thus, its corresponding image shown in Figure 6.7(c) renders more gray voxels that meet the fuzzy matching than the ones rendered in Figure 6.7(b).



query 0 {
prob(vessel) ≥ 0.95;
dropoff = 0.001;
weight = 0.5;
priority = 0
}

query 1 {
prob(vessel) ≥ 0.95;
dropoff = 0.5;
weight = 0.5;
priority = 0
}

query 2 {
prob(vessel) ≥ 0.95;
dropoff = 1.75;
weight = 0.5;
priority = 0
}

(a) *(b)* *(c)*

Figure 6.7: The effects of three singular queries on the MR kidney data set with different dropoff parameters.

6.3.2.3 Weight of Query

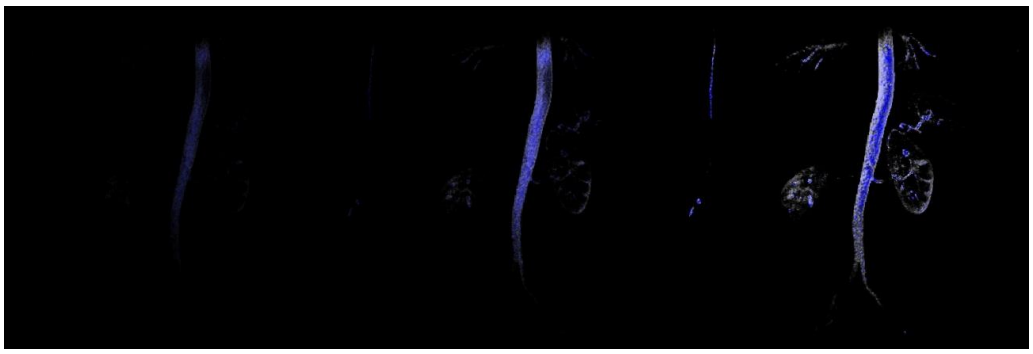
As illustrated in Figure 6.7(a), every singular query can generate a visualization which reveals the result of a query. These singular queries can be combined together to form a more powerful compound query, which presents a more complex visual result. However, as in (Keim and Kriegel, 1994), these singular queries may have different importance to the user. Therefore, in addition to the *dropoff* parameter, every query supports a *weight* parameter, which indicates its importance to the compound query. For every query, its weight is used to directly modulate its fuzzy matching score and can be set to any positive value. Consequently the final score of a compound query can be calculated as the weighted sum of the fuzzy matching scores of all singular queries, as shown in formula 6.8:

$$score = clamp(0,1,\sum_{i\in Q} weight_i \times score_i) \qquad (6.8),$$

where $clamp$ is used to limit the $score$ range to be $[0, 1]$. If the $score$ is greater than 1, then the $clamp$ will cut its value to be 1. If the $score$ is smaller than 0, then the $clamp$ will cut its value to be 0. Q represents the set of singular queries that form the compound query. $weight_i$ represents the *weight* parameter for query i,

specified by the user. $score_i$represents the fuzzy matching score for query i, defined in formula 6.7. *score* represents the final score of the compound query.

As the weight parameter can be any positive value and it does not guarantee formula 6.8 can be summed to unity. Therefore the final *score* of the compound query has to be clamped between 0 and 1 so that it can be used directly as the opacity value of every voxel for subsequent DVR.



(a)	(b)	(c)

query 0 {
prob(vessel) ≥ 0.95;
dropoff = 0.5;
weight = 0.025;
priority = 0
}
	query 1 {
	prob(vessel) ≥ 0.95;
	dropoff = 0.5;
	weight = 0.125;
	priority = 0
	}
		query 2 {
		prob(vessel) ≥ 0.95;
		dropoff = 0.5;
		weight = 1.0;
		priority = 0
		}

Figure 6.8: The effects of three singular queries on the MR kidney data set with different weight parameters.

Figure 6.8 illustrates an example of the effect of the *weight* parameter, and its corresponding explicitly probabilistic TF is shown in Figure 6.5(d). This example consists of three singular queries with the same parameters but the *weight*, as shown in the three text boxes in Figures 6.8(a), (b) and (c), and their resulting visualization are shown in the top three images. We can see that compared to the query in Figure 6.8(a), the one in Figure 6.8(b) has a bigger *weight* value 0.125 that indicates a higher opacity value. Thus the voxels appeared in Figure 6.8(b) are more opaque (less transparent) than those appeared in Figure 6.8(a). We can also see that the query in Figure 6.8(c) has the biggest *weight* value 1.0 among all the three queries, thus the voxels appeared in Figure 6.8(c) are most opaque among all the three images.

6.3.2.4 Priority of Query

The last statement of a query is the rendering priority, as shown in the text box in Figure 6.8(a). It is used when a compound query occurs (for a singular query, it does not take any effect) and determines which query's color should be assigned to the final color of a voxel when this voxel concurrently met multiple singular queries that are incorporated in the compound query. Typically, the voxel's final color is determined by the color of a query which has the highest priority (provided that the voxel met this query). When the priorities of multiple singular queries are equal, this voxel's color will be assigned by a new user specified color (not colors of these existing queries) which indicates it concurrently met multiple singular queries. As a result, the user has opportunities to explore alternative renderings for what-if scenarios. In addition, he is able to reveal the maximum possible rendering boundary of a singular query.

Figure 6.9 shows a specific example to illustrate this, and its corresponding explicitly probabilistic TF is shown in Figure 6.9(d). This example consists of three compound queries, as shown in the text boxes in Figures 6.9(a), (b) and (c). We can see that every compound query incorporates two singular queries: query 0 and query 1. While query 0 corresponds to the kidney material and is used to extract voxels whose kidney material's probability met its criteria, query 1 corresponds to the vessel material and is used to extract those voxels whose vessel material's probabilities met its criteria. The three compound queries are actually three variations of the same compound query, with different *priority* parameters.

The compound query in Figure 6.9(a) illustrates a case of the *priority* parameters where *priority(query 0) == priority(query 1)*, and its corresponding results are shown in the image above it. The red parts of the image indicate those voxels which only met query 0 for material kidney, and the blue parts of the image indicate those voxels which only met query 1 for material vessel. In addition, we can see that there are some yellow parts on the image, and this indicates those voxels which met both query 0 (for kidney) and query 1 (vessel). Based on the image, the user may feel interested to further explore two what-if scenarios: (1) if those voxels which met

both query 0 (for kidney) and query 1 (vessel) are only query 1 (vessel), what will the results look like? (2) if those voxels are only query 0 (kidney), what will the results look like? The corresponding results of these two what-if scenarios can be easily revealed by simply adjusting the *priority* parameters of these two queries, as shown in Figures 6.9(b) and (c).

The compound query in Figure 6.9(b) corresponds to the scenario (1), and its top image shows the corresponding results. By adjusting *priority(query 0) < priority(query 1)*, we actually specified that for those voxels which met both query 0 and query 1, their final colors will be determined by the query 1's color (as it has the highest priority). As a result, those yellow parts shown in the top image in Figure 6.9(a) are rendered as blue, as we assumed only material vessel exists in those voxels. The resulting image in fact formulates the maximum possible rendering boundary of vessel, which corresponds to such a case: as long as a voxel met the query 1, it will be rendered as material vessel (blue). Obviously, this is the maximum possible rendering for material vessel and it corresponds to the most blue voxels. Any blue voxels revealed by other renderings for material vessel will be less than the amount of this case. We believe that the maximum possible rendering boundary for a query will be very valuable for decision making because it clearly and visually depicts the decision boundary for the query.

The compound query in Figure 6.9(c) corresponds to the scenario (2) and its top image shows the corresponding results. By adjusting *priority(query 0) > priority(query 1)*, we specified that for those voxels which met both query 0 and query 1, their final colors will be determined by the query 0's color. As a result, those yellow parts shown in the image in Figure 6.9(a) are rendered as red (as we assumed only material kidney exists in those voxels), which reveals the maximum possible rendering boundary for material kidney.

6.3.3 Query Color Specification

The query color specification is designed in such a way that it allows the user to inversely determine which queries of a compound query are matched by simply

examining the colors of the final image. As a result, the user can obtain more in-depth feedback through analysis of the final image.



```
query 0 {
prob(kidney) ≥ 0.05;
dropoff = 0.001;
weight = 0. 5;
priority = 0
}
query 1 {
prob(vessel) ≥ 0.05;
dropoff = 0.001;
weight = 0.5;
priority = 0
}
```

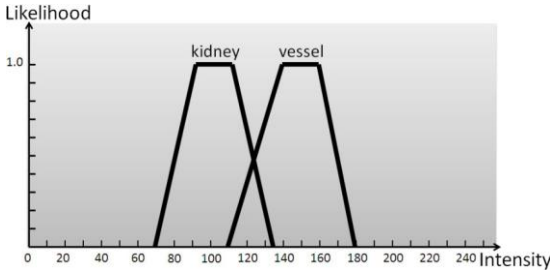
(a)

```
query 0 {
prob(kidney) ≥ 0.05;
dropoff = 0.001;
weight = 0. 5;
priority = 0
}
query 1 {
prob(vessel) ≥ 0.05;
dropoff = 0.001;
weight = 0.5;
priority = 1
}
```

(b)

```
query 0 {
prob(kidney) ≥ 0.05;
dropoff = 0.001;
weight = 0. 5;
priority = 1
}
query 1 {
prob(vessel) ≥ 0.05;
dropoff = 0.001;
weight = 0.5;
priority = 0
}
```

(c)



(d)

Figure 6.9: The effects of three compound queries on the MR kidney data set with different combination of priority parameters.

For a compound query, the number of priorities is determined by the number of singular queries incorporated in the compound query, starting by the lowest priority 0. For example, if a compound query consists of N singular queries, then the number of priorities for this case will be N, enumerated as $0,1,2,\cdots,N-1$. This gives a sufficient resolution to distinguish the priority of every singular query.

To efficiently save the memory, a compound query can be compactly represented by a bit field in the memory, with each individual bit corresponding to a singular query incorporated within it. Every query has its bit turned on (is set to 1) if its probability inequality is fully satisfied (Δ is 0), otherwise has its bit turned off (is set to 0). After performing the compound query on every voxel, different combinations of $0s$ and $1s$ for the bit field can be acquired. When interpreted as an integer, the range of these combinations is between [0, $2^Q - 1$], where Q is the set of singular queries within the compound query. These different combinations are finally mapped to a lookup table for color specification (use the integer value of a combination as the index to the lookup table), and this forms the color scheme for the final image.

Figure 6.10(b) shows an example about the query color specification, and its corresponding compound query is shown in Figure 6.10(a). This example is composed of a compound query with two singular queries: query 0 and query 1. While query 0 is used to extract those voxels whose material kidney's probability meet the query, query 1 is used to extract those voxels whose material vessel' probability meet the query. Thus, this compound query can be compactly represented by a bit field with two bits, with each bit corresponding to a singular query. The two bits generate in total of 4 (2^Q, where Q is 2) different combinations, with a 0 bit indicating that the corresponding query is not matched, and a 1 bit indicating that the corresponding query is matched. Every combination corresponds to an integer value within a range [0, 3], which can be used as an index to look for the corresponding color in the lookup table that is specified by the user. Figure 6.10(c) shows the corresponding results. By examining the colors of this image, the user can inversely know which queries of this compound query are matched *i.e.* the red parts correspond to the areas where query 0 is matched, and the blue parts correspond to the areas where query 1 is matched. Moreover, the yellow parts correspond to the areas where both query 0 and query 1 are matched.

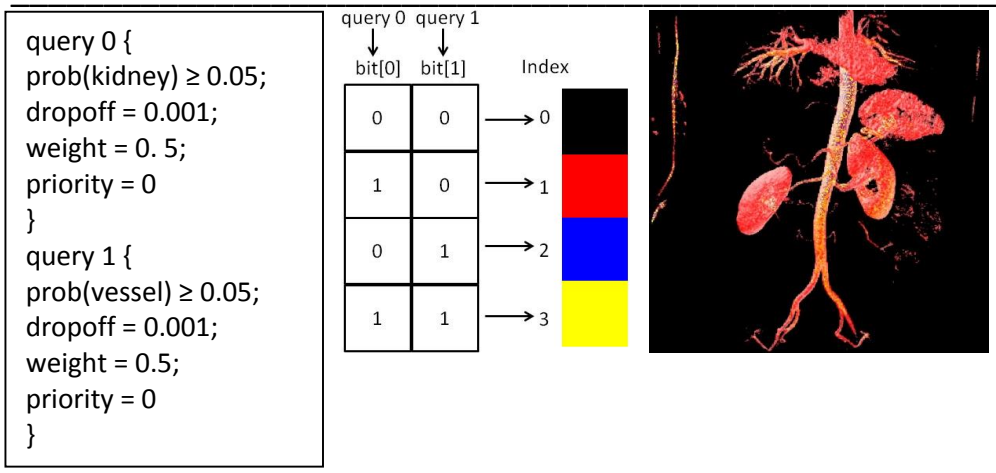


Figure 6.10: An example about how to specify the query color.

6.3.4 Query Volume Rendering

Either a singular or a compound query queries the entire classified 3D scalar data in a voxel by voxel manner. As a result, the colored 3D scalar data is acquired, with its every voxel being assigned with an appropriate color and opacity value with respect to the query. The colored 3D scalar data is then rendered to generate the final image. In this research the classic volume ray-casting algorithm is chosen for the rendering. However, instead of utilizing the most common trilinear interpolation, the easy and fast nearest neighbor interpolation is exploited. This is because the trilinear interpolation could introduce non-specified intermediate colors to the final image when it interpolates the color of an interior sample from its surrounding 8 voxels, and this may prevent the user from inversely determining which queries of a compound query are matched by simply examining the colors of the final image. Compared to the trilinear interpolation, the nearest neighbor interpolation determines the color of an interior sample using the color at its nearest voxel. Thus it will not introduce any non-specified colors to the final image and allows the user to inversely determine the match of queries by examining the colors of the final image. However, a small drawback of the nearest neighbor interpolation is that it typically generates an image with discontinuous color boundaries. To acquire an image that does not include non-specified intermediate colors while has smooth color boundaries, a more advanced interpolation scheme needs to be explored. For

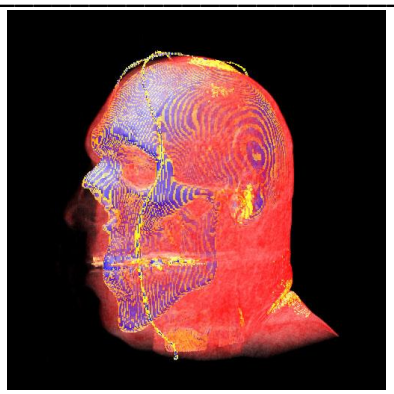
now, the nearest neighbor interpolation is utilized, which remains the capability for the user to inversely determine the match of queries, but at a cost of the image quality (discontinuous).

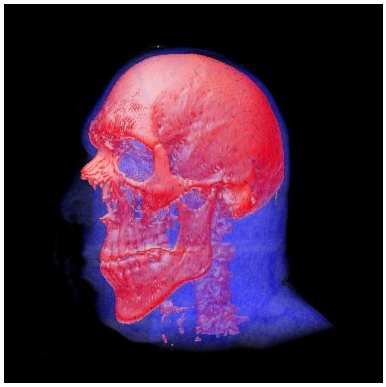
6.3.5 Single Material Query and Multiple Material Query

For a given explicitly probabilistic TF, a singular query can only be used to extract a singular feature from one type of material at a time, as the previous examples illustrated in Figure 6.5, Figure 6.7 and Figure 6.8. As for a compound query, it can be used to extract either multiple features from multiple types of materials (as the previous examples shown in Figure 6.9 and Figure 6.10, with each singular query of a compound query extracts a feature from one material) or multiple features from the same type of material. The latter is illustrated by the two examples shown in Figures 6.11(a) and (b), and their corresponding explicitly probabilistic TF is shown in Figure 6.11(c).

We can see from the compound query in Figure 6.11(a) that it consists of two singular queries, with each query having specific criteria and both concurrently querying the bone material. Consequently three different features of the bone material are extracted, as shown in the top image in Figure 6.11(a). The red parts of the image correspond to the query 0's feature where material bone's probability is less than and equal to 0.75. The blue parts of the image correspond to the query 1's feature where material bone's probability is greater than and equal to 0.55. Finally there are some yellow parts on the image (meet both query 0 and query 1), which correspond to the feature where material bone's probability is between 0.55 and 0.75.

The compound query in Figure 6.11(b) shows another example to extract multiple features from the singular bone material and its corresponding results are shown in the top image in Figure 6.11(b). The red parts of the image correspond to the query 0's feature where material bone's probability equals to 1.0, while the blue parts of the image correspond to the query 1's feature where material bone's probability is less than and equal to 0.5.



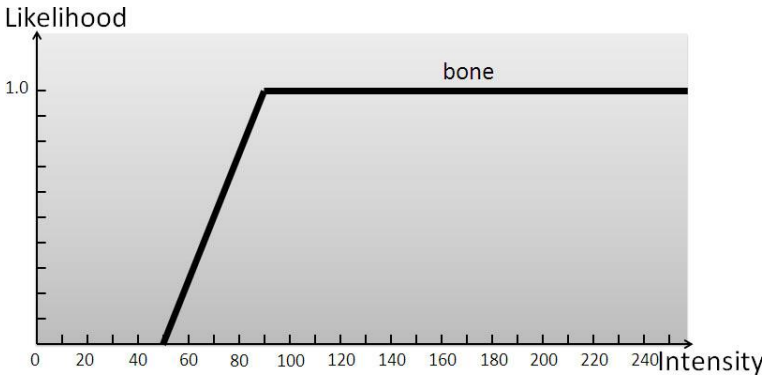


```
query 0 {
prob(bone) ≤ 0.75;
dropoff = 0.2;
weight = 0. 005;
priority = 0
}
query 1 {
prob(bone) ≥ 0.55;
dropoff = 0.2;
weight = 0.8;
priority = 0
}
```

(a)

```
query 0 {
prob(bone) == 1.0;
dropoff = 0.2;
weight = 1;
priority = 0
}
query 1 {
prob(bone) ≤ 0.5;
dropoff = 0.2;
weight = 0.02;
priority = 0
}
```

(b)



(c)

Figure 6.11: Two examples of extracting multiple features from the same material.

In summary, the probabilistic query technique offers a versatile tool to quantitatively visualize the data classification probability within DVR. It can be used to extract various features of a singular material or multiple materials through different query combinations.

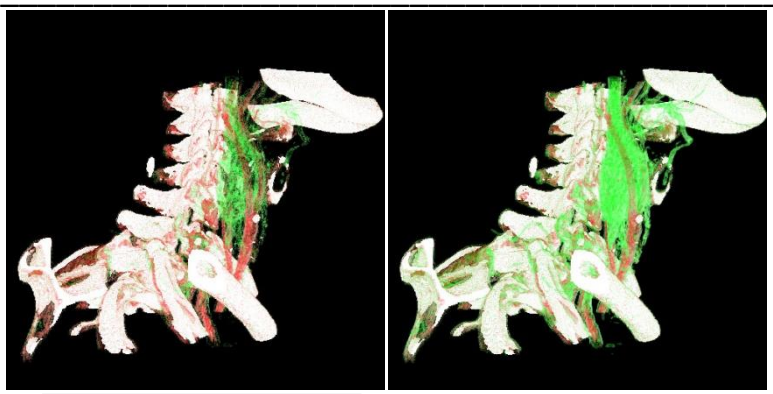
6.3.6 Case Studies

This section will present three case studies by applying the proposed Probabilistic Query uncertainty visualization technique in the domain of medical imaging. In particular, the previous two case studies demonstrate how the proposed Probabilistic Query uncertainty visualization technique could possibly be used as a better decision support tool for the radiologists and physicians to make their diagnostic decision, as opposed to the traditional DVR. The last case study demonstrates the versatility of this technique.

6.3.6.1 Case Study 1: Thyroid Tumor Assessment

The first case study focuses on the thyroid tumor assessment, where the tumor extent in relation to right carotid arteries is crucial for the diagnosis. The data set exploited in this case study is a CT scan of the thyroid tumor in human being's neck, provided by Claes Lundstrom from the Center for Medical Image Science and Visualization (CMIV), Linkoping University. Its original dimensions are $512 \times 512 \times 512$, with 16 bits per voxel. For simplicity, we used ImageJ (National Institutes of Health, 2013) to convert it to a $512 \times 512 \times 355$ data set, with 8 bits per voxel.

Figures 6.12(a) and (b) show two renderings from our proposed Probabilistic Query uncertainty visualization technique, and Figure 6.12(c) shows their corresponding explicitly probabilistic TF. The red divergent object on the right of these renderings indicates the carotid arteries, and the green object indicates the thyroid tumor. The remaining white object indicates the bones inside the neck. From these two renderings we can see that the tumor extents are quite different. The two text boxes below the two renderings are their corresponding queries, which reveal the classification probability of each material appeared on these renderings. Such quantitative information provides very valuable decision support for diagnosis. For example, when radiologists and physicians assess the thyroid tumor extent shown on these two renderings for their tumor removal surgery, it is very clear for them that the thyroid tumor appeared on the left rendering has higher probability (≥ 0.9) than the one appeared on the right rendering (≥ 0.65). Therefore, they may choose the left rendering as their actual surgery foundation.

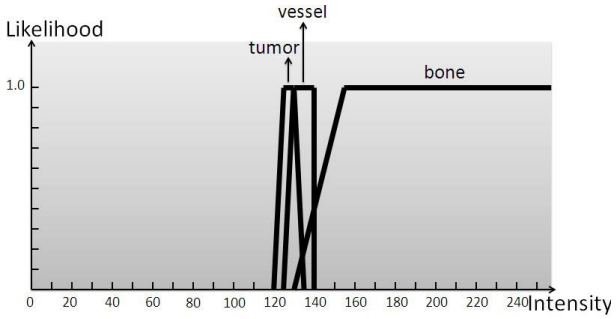


```
query 0 {
prob(tumor) ≥ 0.9;
dropoff = 0.02;
weight = 0.2;
priority = 0;
}
query 1 {
prob(arteries) ≥ 0.75;
dropoff = 0.02;
weight = 0.2;
priority = 0;
}
query 2 {
prob(bone) == 1.0;
dropoff = 0.02;
weight = 1;
priority = 0;
}
```

(a)

```
query 0 {
prob(tumor) ≥ 0.65;
dropoff = 0.02;
weight = 0.2;
priority = 0;
}
query 1 {
prob(arteries) ≥ 0.75;
dropoff = 0.02;
weight = 0.2;
priority = 0;
}
query 2 {
prob(bone) == 1.0;
dropoff = 0.02;
weight = 1;
priority = 0;
}
```

(b)

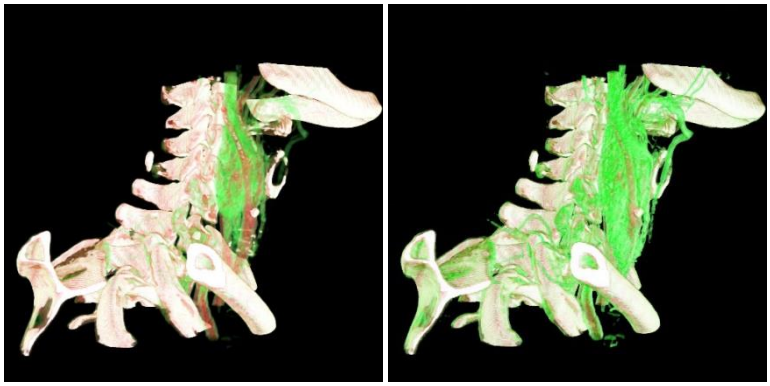


(c)

Figure 6.12: The Probabilistic Query uncertainty visualization technique applied to the CT thyroid tumor data set, with the green object indicating the thyroid tumor, the red object indicating the carotid arteries, and the white object indicating the bones inside the neck.

Such a decision cannot be made by assessing the renderings of tumor extent (as shown in Figures 6.13(a) and (b)) generated by manual adjustments of the TF of the

traditional DVR, as they do not provide any quantifiable information about the material classification uncertainty.



(a) *(b)*

Figure 6.13: Two renderings of the CT thyroid tumor data set, obtained by manual TF adjustments of the traditional DVR.

6.3.6.2 Case Study 2: Stenosis Assessment

The second case study focuses on the stenosis assessment, where the diagnosis task is to determine whether there is vessel stenosis, and according to (Persson et al., 2004), an inadequate visualization can give an incorrect impression of the vessels that affects the medical assessment. The data set exploited in this case study is a MR renal angiography (Roettger, 2012), whose dimensions are $384 \times 512 \times 80$, with 8 bits per voxel.

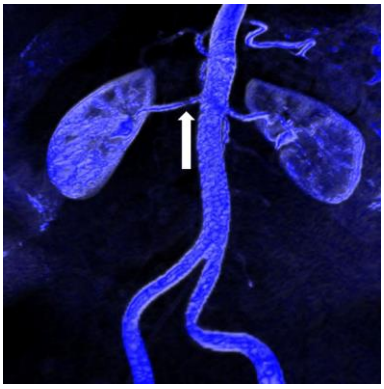
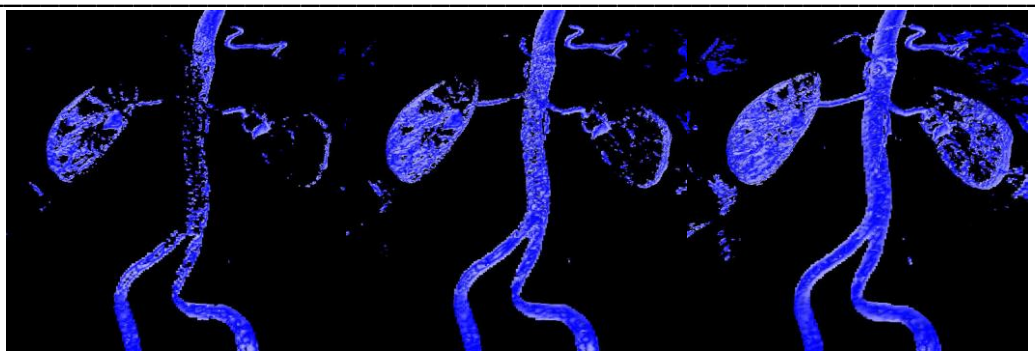


Figure 6.14: A rendering of a MR renal angiography from the traditional DVR, with a suspected stenosis denoted by the white arrow.



```
query 0 {
prob(vessel) == 1.0;
dropoff = 0.1;
weight = 1;
priority = 0;
}
```

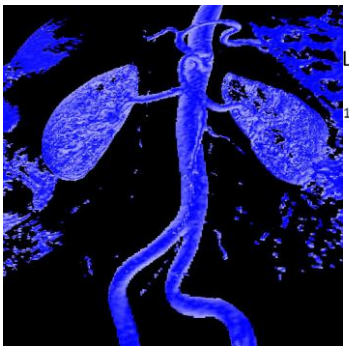
(a)

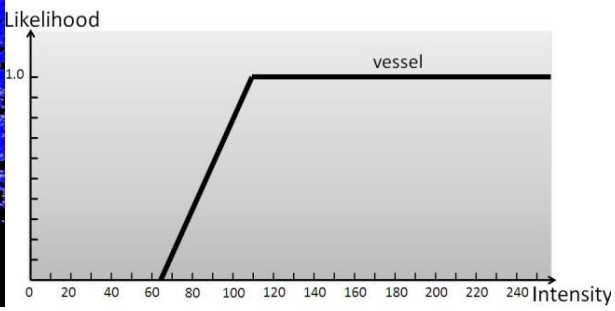
```
query 1 {
prob(vessel) ≥ 0.85;
dropoff = 0.1;
weight = 1;
priority = 0;
}
```

(b)

```
query 2 {
prob(vessel) ≥ 0.625;
dropoff = 0.1;
weight = 1;
priority = 0;
}
```

(c)






```
query 3 {
prob(vessel) ≥ 0.33;
dropoff = 0.1;
weight = 1;
priority = 0;
}
```

(d)

(e)

Figure 6.15: The Probabilistic Query uncertainty visualization technique applied to the MR renal angiography.

Figure 6.14 shows a rendering of the MR renal angiography from the traditional DVR, based on one classification scheme. We can see from this figure that there may be a vessel stenosis, as indicated by the white arrow. Based on this rendering, a surgery may be performed, which has the risk to threaten the patient's life. Experienced radiologists and physicians have learned to use manual TF adjustments of the traditional DVR to question this classification scheme (Lundstrom et al.,

2007)(Lundstrom, 2007). However, as described in Section 6.1, such a method only enables them to explore the material intensity classification uncertainty under an uncontrollable manner, which may result in some alternative renderings that are very important to the final diagnostic decision being missing. Therefore, the radiologists and physicians may not be able to make the correct diagnostic decision, based on the incomplete renderings.

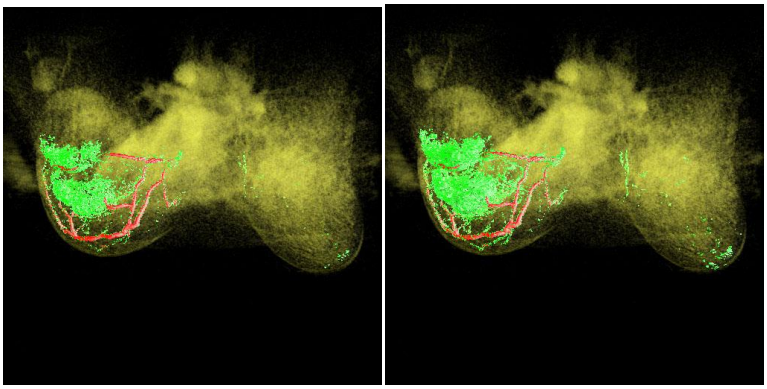
Figures 6.15(a), (b), (c) and (d) show four renderings of the MR renal angiography that correspond to four specific queries (in text boxes) from the proposed Probabilistic Query uncertainty visualization technique. Figure 6.15(e) shows their corresponding explicitly probabilistic TF. We can see from these queries that how the Probabilistic Query technique enables users to explore the vessel material classification probability in a controllable and quantitative manner – by simply adjusting the probability inequality. This technique makes sure that all alternative renderings in relation to the vessel classification probability could be queried and revealed, for a given explicitly probabilistic TF. Therefore, it could expand the decision support for radiologists and physicians who use the traditional DVR, and assist them to make the correct decision. Taking the stenosis assessment as an example, by observing the four renderings revealed in Figure 6.15, we can see that there is no vessel stenosis.

6.3.6.3 Case Study 3: Breast Tumor Assessment

The last case study takes four examples to show the versatility of the Probabilistic Query uncertainty visualization technique applied to the breast tumor assessment. The first example makes use of the technique to reveal alternative breast tumor extent in relation to the breast. The second example utilizes it to reveal alternative correlations between the breast tumor and its surrounding vessels. The third example shows its capability for “what-if” scenarios exploration, as well as for the maximum possible tumor rendering boundary display. Finally, the fourth example shows its capability for complex feature localization, in which case we want to search in the 3D object space where certain complex features that have been specified in

the query are presented. The data set exploited here is about tumor in breast (Roettger, 2012), whose dimensions are $448 \times 448 \times 208$, with 8 bits per voxel.

Figure 6.16(a) to Figure 6.16(d) shows four alternative renderings in relation to the first example from the Probabilistic Query technique. These four renderings are generated from four compound queries with different classification probabilities of the tumor material, as shown in the four text boxes in Figure 6.16(a) to Figure 6.16(d). Their corresponding explicitly probabilistic TF is shown in Figure 6.16(e). We can see that every rendering reveals the appearance and extent of the breast tumor at a specific probability level.

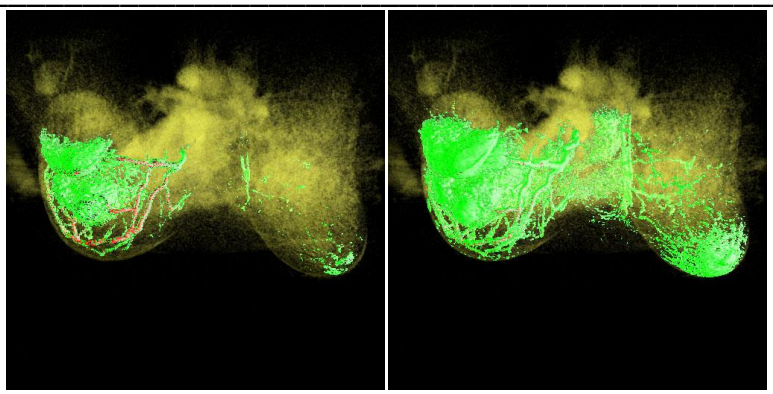


```
query 0 {
prob(breast) == 1.0;
dropoff = 0.15;
weight = 0.02;
priority = 0;
}
query 1 {
prob(tumor) == 1.0;
dropoff = 0.15;
weight = 1;
priority = 0;
}
query 2 {
prob(vessel) == 1.0;
dropoff = 0.15;
weight = 1;
priority = 0;
}
```

(a)

```
query 0 {
prob(breast) == 1.0;
dropoff = 0.15;
weight = 0.02;
priority = 0;
}
query 1 {
prob(tumor) ≥ 0.9;
dropoff = 0.15;
weight = 1;
priority = 0;
}
query 2 {
prob(vessel) == 1.0;
dropoff = 0.15;
weight = 1;
priority = 0;
}
```

(b)

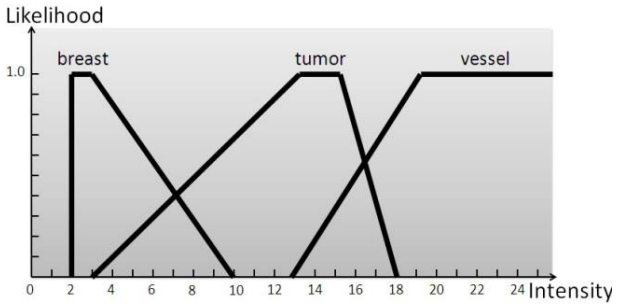


```
query 0 {
prob(breast) == 1.0;
dropoff = 0.15;
weight = 0.02;
priority = 0;
}
query 1 {
prob(tumor) ≥ 0.75;
dropoff = 0.15;
weight = 1;
priority = 0;
}
query 2 {
prob(vessel) == 1.0;
dropoff = 0.15;
weight = 1;
priority = 0;
}
```

(c)

```
query 0 {
prob(breast) == 1.0;
dropoff = 0.15;
weight = 0.02;
priority = 0;
}
query 1 {
prob(tumor) ≥ 0.35;
dropoff = 0.15;
weight = 1;
priority = 0;
}
query 2 {
prob(vessel) == 1.0;
dropoff = 0.15;
weight = 1;
priority = 0;
}
```

(d)

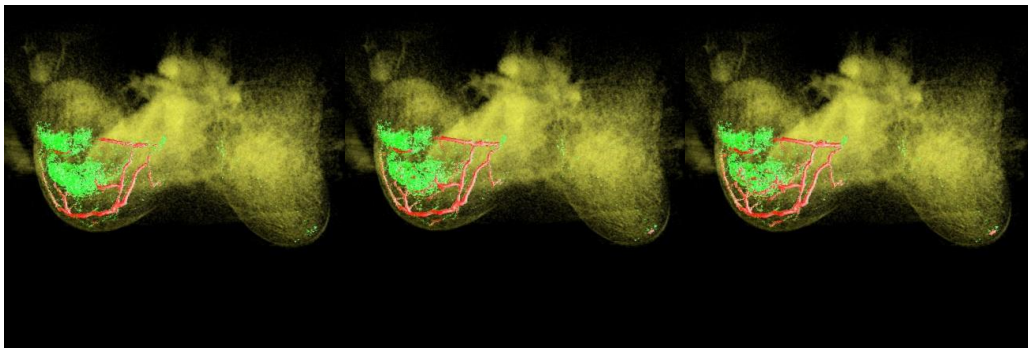


(e)

Figure 6.16: The Probabilistic Query uncertainty visualization applied to examine the breast tumor extent in relation to the breast. The green material indicates the breast tumor, and the red material indicates the vessels. The yellow material indicates the breast.

Figure 6.17(a) to Figure 6.17(c) illustrate three alternative renderings in relation to the second example from the Probabilistic Query technique. These renderings are generated from three compound queries with different classification probabilities of

the vessel material, as shown in the three text boxes in Figure 6.17(a) to Figure 6.17(c). Their corresponding explicitly probabilistic TF is shown in Figure 6.16(e). We can see that every rendering reveals different correlations between the tumor and its surrounding vessels at a specific probability level. By looking at Figure 6.17(a), we can see that there are very few red vessels inside the bottom green tumor. Compared to it, we can see that the bottom green tumor in Figure 6.17(b) incorporates a little bit more red vessels, while the one in Figure 6.17(c) incorporates much more red vessels.



```
query 0 {
prob(breast) == 1.0;
dropoff = 0.15;
weight = 0.02;
priority = 0;
}
query 1 {
prob(tumor) == 1.0;
dropoff = 0.15;
weight = 0.7;
priority = 0;
}
query 2 {
prob(vessel) == 1.0;
dropoff = 0.15;
weight = 1;
priority = 0;
}
```

(a)

```
query 0 {
prob(breast) == 1.0;
dropoff = 0.15;
weight = 0.02;
priority = 0;
}
query 1 {
prob(tumor) == 1.0;
dropoff = 0.15;
weight = 0.7;
priority = 0;
}
query 2 {
prob(vessel) ≥ 0.5;
dropoff = 0.15;
weight = 1;
priority = 0;
}
```

(b)

```
query 0 {
prob(breast) == 1.0;
dropoff = 0.15;
weight = 0.02;
priority = 0;
}
query 1 {
prob(tumor) == 1.0;
dropoff = 0.15;
weight = 0.7;
priority = 0;
}
query 2 {
prob(vessel) ≥ 0.325;
dropoff = 0.15;
weight = 1;
priority = 0;
}
```

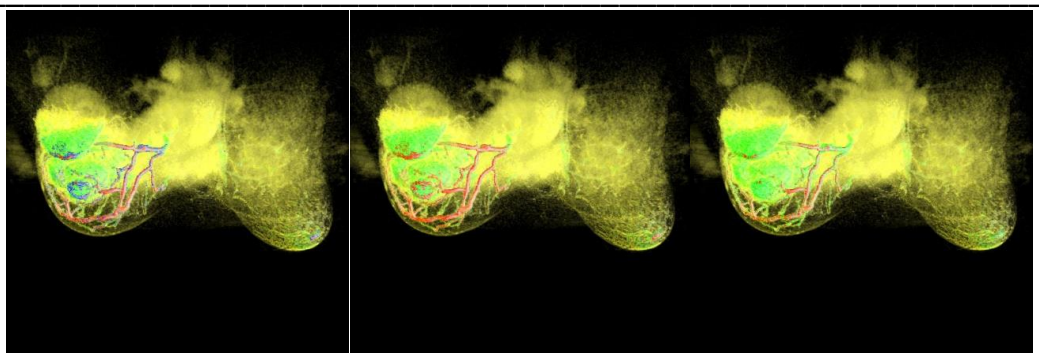
(c)

Figure 6.17: The Probabilistic Query uncertainty visualization applied to examine the correlations between the tumor and its surrounding vessels.

Figure 6.18(a) to Figure 6.18(d) shows four alternative renderings with respect to the third example from the Probabilistic Query technique. These renderings are

generated from four compound queries with different priorities for the breast material, tumor material and vessel material, as shown in the four text boxes in Figure 6.18(a) to Figure 6.18(d). Their corresponding explicitly probabilistic TF is shown in Figure 6.16(e). We can see that every rendering reveals different correlations among breast, tumor and vessels. Beginning with Figure 6.18(a), we can see that there are some blue parts on this image. They indicate those voxels which met both query 1 for the tumor material and query 2 for the vessel material. Based on this image, radiologists and physicians may feel interested to ask two "what-if" questions for further exploration: (1) if those voxels only consist of the vessel material, what is the possible extent of the tumor? (2) if those voxels only consist of the tumor material, what is the possible extent of the tumor? The answers of these two "what-if" questions can be easily revealed by adjusting the priorities of the tumor material and the vessel material in the queries (as shown in the two text boxes in Figures 6.18(b) and (c)), and Figures 6.18(b) and (c) illustrate the corresponding results to the first and second "what-if" question, respectively. In contrast to these two images, we can see that the tumor in Figure 6.18(c) incorporates less red vessels than the one in Figure 6.18(b). Thus, its tumor extent is slightly bigger than the tumor extent in Figure 6.18(b).

In addition to the two "what-if" questions, radiologists and physicians may also want to know the maximum possible tumor rendering boundary for the given compound query. The answer to this question can also be easily revealed by simply assigning the tumor material with the highest priority, and Figure 6.18(d) shows the corresponding results. Compared to other alternative renderings, we can see that this image has the maximum tumor extent. This gives the radiologist and physicians useful information to make an "extreme" diagnostic decision *i.e.* the maximum tumor extent they could remove from the breast, for the given query. Figure 6.18(e) to Figure 6.18(h) illustrate some more alternative renderings based on other priority combinations with respect to the given compound query.



```
query 0 {
prob(breast) ≥ 0.1;
dropoff = 0.15;
weight = 0.02;
priority = 2;
}
query 1 {
prob(tumor) ≥ 0.4;
dropoff = 0.15;
weight = 0.2;
priority = 0;
}
query 2 {
prob(vessel) ≥ 0.2;
dropoff = 0.15;
weight = 1;
priority = 0;
}
```

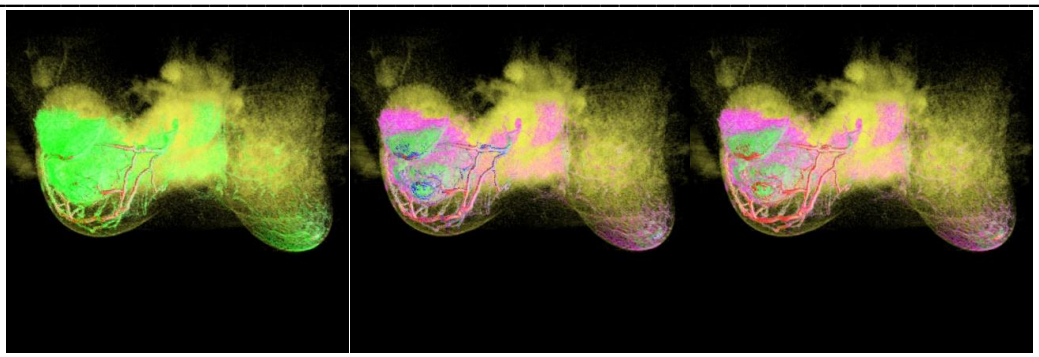
(a)

```
query 0 {
prob(breast) ≥ 0.1;
dropoff = 0.15;
weight = 0.02;
priority = 2;
}
query 1 {
prob(tumor) ≥ 0.4;
dropoff = 0.15;
weight = 0.2;
priority = 0;
}
query 2 {
prob(vessel) ≥ 0.2;
dropoff = 0.15;
weight = 1;
priority = 1;
}
```

(b)

```
query 0 {
prob(breast) ≥ 0.1;
dropoff = 0.15;
weight = 0.02;
priority = 2;
}
query 1 {
prob(tumor) ≥ 0.4;
dropoff = 0.15;
weight = 0.2;
priority = 1;
}
query 2 {
prob(vessel) ≥ 0.2;
dropoff = 0.15;
weight = 1;
priority = 0;
}
```

(c)



```
query 0 {
prob(breast) ≥ 0.1;
dropoff = 0.15;
weight = 0.02;
priority = 0;
}
query 1 {
prob(tumor) ≥ 0.4;
dropoff = 0.15;
weight = 0.2;
priority = 2;
}
query 2 {
prob(vessel) ≥ 0.2;
dropoff = 0.15;
weight = 1;
priority = 0;
}
```

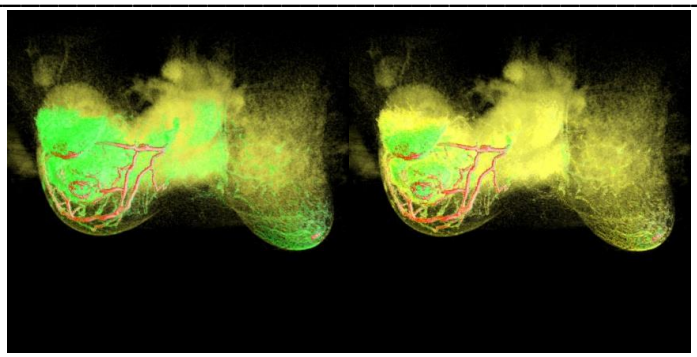
(d)

```
query 0 {
prob(breast) ≥ 0.1;
dropoff = 0.15;
weight = 0.02;
priority = 0;
}
query 1 {
prob(tumor) ≥ 0.4;
dropoff = 0.15;
weight = 0.2;
priority = 0;
}
query 2 {
prob(vessel) ≥ 0.2;
dropoff = 0.15;
weight = 1;
priority = 0;
}
```

(e)

```
query 0 {
prob(breast) ≥ 0.1;
dropoff = 0.15;
weight = 0.02;
priority = 0;
}
query 1 {
prob(tumor) ≥ 0.4;
dropoff = 0.15;
weight = 0.2;
priority = 0;
}
query 2 {
prob(vessel) ≥ 0.2;
dropoff = 0.15;
weight = 1;
priority = 2;
}
```

(f)



```
query 0 {
prob(breast) ≥ 0.1;
dropoff = 0.15;
weight = 0.02;
priority = 0;
}
query 1 {
prob(tumor) ≥ 0.4;
dropoff = 0.15;
weight = 0.2;
priority = 1;
}
query 2 {
prob(vessel) ≥ 0.2;
dropoff = 0.15;
weight = 1;
priority = 2;
}
```

(g)

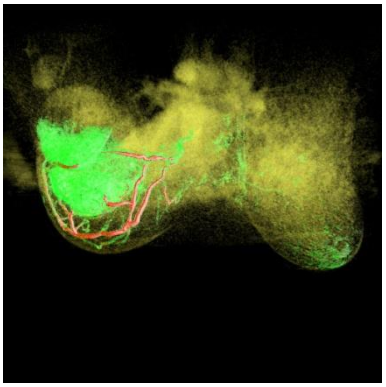
```
query 0 {
prob(breast) ≥ 0.1;
dropoff = 0.15;
weight = 0.02;
priority = 1;
}
query 1 {
prob(tumor) ≥ 0.4;
dropoff = 0.15;
weight = 0.2;
priority = 0;
}
query 2 {
prob(vessel) ≥ 0.2;
dropoff = 0.15;
weight = 1;
priority = 2;
}
```

(h)

Figure 6.18: The Probabilistic Query uncertainty visualization applied to reveal results of (1) the what-if scenarios and (2) the maximum possible tumor rendering boundary, which may be interested to the radiologists and physicians.

Figure 6.19 shows the results with respect to the fourth example from the Probabilistic Query technique. Its corresponding explicitly probabilistic TF is shown in Figure 6.16(e). From the compound query shown in the text box in Figure 6.19 we can see that there are three features to be extracted. First, we want to know where those voxels which are fully composed of breast material are appeared? Second, we want to know where those voxels whose tumor material's probability is two times greater than their breast material's probability are presented? Third, we want to know those voxels whose vessel material's probability is 2.5 times greater than their tumor material's probability are appeared? From the image in Figure 6.19 we can see that where these features are located. The yellow parts denote the positions of the

first feature, and the green parts indicate the positions of the second feature. Finally, the red parts indicate the positions of the third feature.



```
query 0 {
prob(breast) == 1.0;
dropoff = 0.15;
weight = 0.02;
priority = 0;
}
query 1 {
prob(tumor) > 2*prob(breast);
dropoff = 0.15;
weight = 0.2;
priority = 0;
}
query 2 {
prob(vessel) > 2.5*prob(tumor);
dropoff = 0.15;
weight = 1;
priority = 0;
}
```

Figure 6.19: An example to show the Probabilistic Query uncertainty visualization's capability for complex feature localization.

6.4 Summary

This chapter presented a Probabilistic Query uncertainty visualization technique, which addressed two issues (see Section 6.1) about the exploration of material intensity classification uncertainty, caused by manual TF adjustments of the traditional DVR. First, it enables users to explore the material intensity classification probability in a controllable manner through customizable and interactive query, and thus is able to reveal all alternative renderings for a given explicitly probabilistic TF. Second, its resulting renderings always correspond to the customizable queries, and thus are able to give indication about the classification probability information associated to the materials appeared on these renderings. Consequently it could provide radiologists and physicians with better decision support for diagnosis, as opposed to the traditional DVR. This have been illustrated by the three case studies in the domain of medical imaging.

At this point, we have presented all our research relevant to this thesis, either visualization of uncertainty or uncertainty of visualization (see Section 1.1.1.1 for detail). In the next chapter, we will sum up the entire thesis and discuss about the future work.

Chapter 7 Conclusions and Future Work

7.1 Conclusions

As described in Chapter 1, uncertainty is pervasive in 3D scalar data and can be introduced at any stage of the visualization pipeline, but it is often overlooked by most volume visualization research. In addition, there has been a recognized need for uncertainty visualization in 3D scalar data, as such visualization can faithfully represent data and thus enables users to draw correct conclusions or make correct decisions and interpretation from the data. Therefore, this thesis focuses on the research of uncertainty visualization in 3D scalar data.

As mentioned in Chapter 1, according to the visualization pipeline stage in which the uncertainty is introduced, two types of uncertainty visualization can be distinguished. Visualization of uncertainty focuses on the uncertainty introduced in either the data acquisition stage or the data transformation that occurs during the data acquisition, while uncertainty of visualization focuses on the uncertainty introduced in either the data visualization stage or the data transformation that occurs during the visualization. By keeping this in mind, three specific research hypotheses have been formulated, which cover either visualization of uncertainty or uncertainty of visualization, as itemized below:

• **Hypothesis 1:** the proposed Texture uncertainty visualization technique enables users to better identify scalar and error data and provides reduced visual overload and more appropriate brightness than four state-of-the-art uncertainty visualization techniques, as demonstrated using a perceptual effectiveness user study.

The work relevant to this hypothesis belongs to the visualization of uncertainty, as its uncertainty is generated in the data transformation stage that occurs during data acquisition. Three main steps of our framework (as introduced in Chapter 3) have

been applied to fully address this hypothesis. First, during the uncertainty data modeling main step, the uncertainty is clearly defined as errors between the original univariate 3D scalar data set and any of its lower resolution data sets, and can be generated from a two-step modeling approach (MR modeling and quantification). This generates the lower resolution univariate 3D data, which serve as the basic input data for the subsequent Texture uncertainty visualization technique, as well as those 4 commonly used techniques (these techniques are isosurface rendering-based uncertainty visualization techniques) chosen for the final evaluation. Second, during the uncertainty visualization main step, the Texture uncertainty visualization technique is proposed, which includes two types of textures and can be used to visualize the errors generated from the first step in isosurface rendering. Third, during the evaluation main step, a formal user study is conducted, which compares the Texture uncertainty visualization technique with other 4 commonly used uncertainty visualization techniques at 4 perceptual aspects that are considered to be important for uncertainty visualization, based on users' subjective rating.

As a result, this work provides the following contribution: although the Texture uncertainty visualization technique (including two types of textures) is proved to have some utilities in certain perceptual aspects, they are not as good as the existing hue and texture opacity uncertainty visualization techniques. In addition, the perceptual effectiveness of the 4 commonly used uncertainty visualization techniques is revealed. These findings could provide useful guidance for future uncertainty visualization design.

• **Hypothesis 2:** the proposed LVIS uncertainty visualization technique enables users to better search max/min scalar and error data than four state-of-the-art uncertainty visualization techniques, as demonstrated using a perceptual effectiveness user study.

The work relevant to this hypothesis also belongs to the visualization of uncertainty, as its uncertainty is modeled in the same way as the one in hypothesis 1. The same three main steps of our framework have been applied to fully address this hypothesis. First, during the uncertainty data modeling main step, the uncertainty is clearly defined as errors (the same as the definition in hypothesis 1) and the same two-step

modeling approach can be applied to generate the lower resolution univariate 3D data, which are used as the input data for the subsequent LVIS uncertainty visualization technique, as well as those 4 commonly used techniques (these techniques are DVR-based uncertainty visualization techniques, which are different from the 4 techniques chosen in hypothesis 1) chosen for the evaluation. Second, during the uncertainty visualization main step, the LVIS uncertainty visualization is proposed, which takes advantage of linked views and interactive specification together to clearly visualize the errors in DVR. Third, during the evaluation main step, a formal user study is designed and performed, which is intended to compare the perceptual effectiveness of the LVIS techniques with other 4 commonly used uncertainty visualization techniques, based on users' task completion accuracies and time.

The analysis results from the evaluation suggest that the proposed LVIS technique enables users to more accurately complete the 4 perception-related searching tasks than other 4 commonly used uncertainty visualization techniques. Therefore, we further extended it to another work – ELVIS, which addresses the insufficient research subarea in uncertainty visualization for multivariate 3D scalar data. Currently, two main steps of our framework have been implemented for this work. First, the uncertainty data modeling main step has been implemented, which clearly defines the uncertainty as errors between an original multivariate 3D scalar data set and any of its lower resolution data sets and extends the two-step modeling approach (previously used to generate the lower resolution univariate 3D data) to generate the lower resolution multivariate 3D data for the ELVIS use. Second, the actual ELVIS uncertainty visualization technique has been implemented, which has been applied to the data generated from the first main step to show its application and the corresponding results have been reported. To validate its effectiveness, the third main step – evaluation is needed to be performed. This is considered as future work and will be discussed it in the next section.

To sum up, the above-mentioned two works provide the following contribution: a LVIS uncertainty visualization is developed, which has better perceptual effectiveness than other 4 commonly used uncertainty visualization techniques. In

addition, it is extended to the ELVIS, which can be used to visualize uncertainty in multivariate 3D scalar data. Furthermore, the perceptual effectiveness of the 4 commonly used uncertainty visualization techniques are revealed. These could provide useful guidance for future uncertainty visualization design.

• **Hypothesis 3:** the proposed Probabilistic Query uncertainty visualization technique, in comparison to traditional DVR methods, enables radiologists and physicians to better identify possible alternative renderings relevant to a diagnosis and the classification probabilities associated to the materials appeared on these renderings; this leads to improved decision support for diagnosis, as demonstrated in the domain of medical imaging.

The work in relation to this hypothesis can be classified as uncertainty of visualization, as it deals with the uncertainty that is introduced during the data transformation that occurs as part of the visualization. Two main steps of our framework have been implemented for this work. First, the uncertainty data modeling main step has been implemented, which clearly defines the uncertainty as the material intensity classification probability, and takes advantage of the explicitly probabilistic TFs to quantify it. This generates the classified 3D scalar data, which serve as the input data for subsequent Probabilistic Query uncertainty visualization technique. Second, the actual Probabilistic Query technique has been implemented, which enables users to extract material probability features that correspond to their customizable and interactive query and finally render these features as images for diagnosis. This technique has been applied to specific case studies in the domain of medical imaging to illustrate its possible advantages of providing radiologists and physicians with better decision support for diagnosis, as opposed to the traditional DVR. To validate this, a further evaluation work in cooperation with radiologists and physicians is needed. This is considered as future work and will be discussed in the next section.

To summarize, this work provides the following contribution: a Probabilistic Query uncertainty visualization technique is proposed, which is very promising to provide radiologists and physicians with better decision support for diagnosis than the traditional DVR.

7.2 Future Work

Some future work needs to be undertaken for the research described in this thesis. Currently both the LVIS and ELVIS uncertainty visualization techniques only enable users to interactively visualize all scalar values and their associated error values of the 3D data within the entire user-specified ROI. This will be insufficient for the case where users only want to visualize scalar values and their associated errors within the ROI at a specific depth. Therefore, it would be a very good idea to integrate the slicing visualization technique (either orthogonal slicing or oblique slicing, please refer to (Moller, 2010) for explanation) into them so that such an issue can be solved. For example, by specifying a specific depth value, users can visualize all scalar values and their associated errors on the slice plane at that depth within the ROI. Integrating such a technique could meet some more sophisticated cases, for example, users could visualize the scalar values and their associated errors on an arbitrarily oriented slice within the ROI.

As mentioned in Chapter 5, for the ELVIS uncertainty visualization technique that focuses on uncertainty visualization in multivariate 3D scalar data, a formal evaluation work is needed to validate its effectiveness. To our knowledge, there has not been research reported to address the evaluation of uncertainty visualization for multivariate 3D scalar data. This is probably because most uncertainty visualization works only focus on univarate 3D scalar data, and little research has been focused on multivariate 3D scalar data. Therefore, we consider both uncertainty visualization in multivariate 3D scalar data and its relevant evaluation as two future directions that need further study. As described in (Brodlie, Osorio and Lopes, 2012), advances in multifield visualization (which itself has been identified as one top visualization research problem (Johnson, 2004)) may help uncertainty visualization in multivariate 3D scalar data, because an uncertainty measure such as standard deviation can be treated as an extra variable. Therefore, it is necessary to expand our knowledge to incorporate multifield visualization, which may provide good solutions to address the two above-mentioned directions. Additionally, another issue that needs to be addressed for the ELVIS is its PC component, which is currently implemented in a very basic form, and thus suffers from over-plotting when a large ROI is specified, resulting

in a view that is too cluttered to show trends or structures. To solve this issue, a more advantaged PC implementation is required, as the ones proposed by (Johansson et al., 2005) (Bendix, Kosara and Hauser, 2005)(Novotny and Hauser, 2006).

As mentioned in the last section, for the Probabilistic Query uncertainty visualization technique, a formal evaluation is needed to validate its advantages of providing radiologists and physicians with better decision support for diagnosis, as opposed to the traditional DVR. This involves a user study that randomly divides the radiologists and physicians who participated in the study into two groups, say $\mathrm{group1}$ and $\mathrm{group2}$. For the subjects in $\mathrm{group1}$, we ask them to use the Probabilistic Query to complete a carefully designed diagnosis task. For the subjects in $\mathrm{group2}$, we ask them to use the DVR to complete the same diagnosis task. By statistically analyzing their task completion accuracies, we can validate our conclusion. Such a user study is considered as future work, as currently there has been no cooperation set up with the hospital and thus it is difficult to recruit the expected subjects. In addition, as described in Section 6.3.4, the nearest neighbor interpolation is currently exploited in the Probabilistic Query technique, which could effectively avoid introducing any non-specified intermediate colors and thus enables the user to inversely determine the matched queries from the final image. However, its drawback is to generate discontinuous color boundaries. To obtain a smoother image, a more advanced interpolation scheme should be explored, as the one proposed in (Johnson and Huang, 2009). Moreover, currently we interact with the Probabilistic Query technique directly through the code. An interactive easy-to-use user interface is expected to be implemented, which should not only allow users to easily input and adjust the query statement parameters, but also give them useful guidance and feedback to set and refine these parameters, according to their specific application needs *i.e.* showing certain material's probability histogram to users would be helpful for them to identify the interested probability to query.

Currently, the four uncertainty visualization techniques including the Texture, LVIS, ELVIS and Probabilistic Query are all implemented based on Central Processing Unit (CPU). Therefore the speed for rendering the 3D scalar data could be relatively slow.

To increase the rendering speed and achieve a more effective interaction, a Graphics Processing Unit (GPU)-accelerated implementation of these techniques is necessary.

Finally, there are two follow-on works that we believe are worth pursuing. First, it would be useful to extend the Probabilistic Query technique to visualize the PDF in 3D scalar data (where each voxel of the data incorporates a random variable, say F, with its PDF, $\mathrm{g(f)}$). A specific application of this case could be, for example, when assessing the temperature in short-term weather forecasting, we can make the following query, as illustrated in formula 7.1:

$$\mathrm{prob}(35℃ \leq \mathrm{F} \leq 50℃) > 50\% \qquad (7.1),$$

to visualize those areas that may have a high temperature $(35℃ - 50℃)$, whose probability of occurrence is greater than 50%. Second, through our Probabilistic Query study it is clear how visualization could involve uncertainty and lead users to make incorrect interpretations, conclusions or decisions. To improve the visualization as a more reliable scientific computing tool, we need to not only visually depict the uncertainty in visualization, but also verify its process, like what we do for other computational sciences. This leads to a new concept of “verifiable visualization”, which has been recently identified as a significant research direction by some leading experts (Kirby and Silva, 2008)(Nathaniel and Ma, 2011).

References

Akiba, H., Ma, K. L. (2007) A Tri-Space Visualization Interface for Analyzing Time-Varying Multivariate Volume Data. In Proc. of IEEE-VGTC Symposium on Visualization, pp. 115-122. Eurographics Association.

Akiba, H., Ma, K. L., Chen, J. H., Hawkes, E. R. (2007) Visualizing Multivariate Volume Data from Turbulent Combustion Simulations. Journal of Computing in Science and Engineering, Vol. 9. No. 2, pp. 76-83.

Alexandre, D. S., Tavares, J. M. R. S. (2010) Introduction of Human Perception in Visualization. International Journal of Imaging and Robotics, Vol. 4, No. A10.

Bayer, B. E. (1973) An Optimum Method for Tow-level Rendition of Continuous-tone Pictures. IEEE International Conference on Communications, vol. 1, pp. 11-15.

Bendix, F., Kosara, R., Hauser, H. (2005) Parallel Sets: Visual Analysis of Categorical Data. In Proc. of the IEEE Symposium on Information Visualization, pp. 133-140.

Benjamini, Y. (1988) Opening the Box of A Boxplot. The American Statistician, Vol. 42, No. 4, pp. 257-262.

Bentley, J. P. (2009) Two-way ANOVA in SPSS. Available at: http://www.mcsr.olemiss.edu/mathematica/Two-factorANOVAinSPSS.pdf.

Borgo, R., Cignoni, P., Scopigno, R. (2004) Simplicial-based Techniques for Multi-resolution Volume Visualization: An Overview. Geometric Modeling for Scientific Visualization, pp. 309-328.

Brodlie, K. (2006) Envisioning Information – Scientific Visualization Scalar 3D Data. Lecture Notes. School of Computing, University of Leeds, pp. 1-46.

Brodlie, K. W., Carpenter, L. A., Earnshaw, R. A., Gallop, J. R., Hubbold, R. J., Mumford, A. M., Osland, C. D., and Quarendon, P. (1992) Scientific Visualization – Techniques and Applications. Springer-Verlag.

Brodlie, K., Osorio, R. A., Lopes, A. (2012) A Review of Uncertainty in Data Visualization. Expanding the Frontiers of Visual Analytics and Visualization, pp. 81 – 109. Springer London. DOI: 10.1007/978-1-4471-2804-5_6.

Brown, R. (2004) Animated Visual Vibrations as An Uncertainty Visualization Technique. In Proc. of International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia, pp.84-89.

Brownlee, C., Patchett, J., Lo, L. T., DeMarle, D., Mitchell, C., Ahrens, J. and Hansen, C. D. (2012) A Study of Ray Tracing Large-scale Scientific Data in Parallel Visualization Applications. In Proc. of Eurographics Symposium on Parallel Graphics and Visualization, pp. 51-60. Eurographics Association.

Cassell, J., Vihjalmsson, H. (1999) Fully Embodies Conversational Avatars: Making Communicative Behaviors Autonomous. Autonomous Agents and Multiagent Systems, Vol. 2, No. 1, pp. 45-64.

Cedilnik, A., Rheingans, P. (2000) Procedural Annotation of Uncertainty Information. In Proc. of IEEE Visualization, pp. 77-84.

Celebi, O. C. (2013) Scientific Visualization and 3D Volume Rendering. Online Tutorial, available at: http://www.byclb.com/TR/Tutorials/volume_rendering/.

Chan, W. W-Y. (2006) A Survey on Multivariate Data Visualization. Technical Report. Department of Computer Science and Engineering, Hong Kong University of Science and Technology.

Chen, C.M. (2005) Top Unsolved Information Visualization Problems. IEEE Computer Graphics and Applications. Vol. 25, No. 4, pp. 12-16.

Choonpradub, C., McNeil, D. (2005) Can the Box Plot be Improved? Songklanakarin Journal of Science and Technology, Vol. 27, No. 3, pp. 649-657.

Cignoni, P., Floriani, L. D., Montani, C., Puppo, E., Scopigno, R. (1994) Multiresolution Modeling and Visualization of Volume Data Based on Simplicial Complexes. In Proc. of the Symposium on Volume Visualization, pp. 19-26.

Cignoni, P., Rocchini, C., Scopigno, R. (1998) Metro: Measuring Error on Simplified Surfaces. Computer Graphics Forum, Vol. 17, No. 2, pp. 167-174.

Community Coordinated Modeling Center. (2007) OpenGGCM, available at: http://ccmc.gsfc.nasa.gov/models/modelinfo.php?model=OpenGGCM.

Coninx, A., Bonneau, G. P., Droulez, J., Thibault, G. (2011) Visualization of Uncertainty Scalar Data Fields Using Color Scales and Perceptually Adapted Noise. In Proc. of Applied Perception in Graphics and Visualization, pp. 59-66.

Dachille, F. (1997) Volume Visualization Algorithms and Architectures. Research Proficiency Exam, SUNY at Stony Brook.

Daubechies, I. (1992) Ten Lectures on Wavelets, SIAM, Philadelphia, Pennsylvania.

Davis, T. J., Keller, C. P. (1997) Modeling and Visualizing Multiple Spatial Uncertainties. Computers and Geosciences, Vol. 23, No. 4, pp. 397-408.

Dinesha, V., Adabala, N., Natarajan, V. (2012) Uncertainty Visualization Using HDR Volume Rendering. The Visual Computer. Vol. 28, No. 3, pp. 265-278. Springer-Verlag. DOI: 10.1007/s00371-011-0614-7.

Dix, A.. (2013) Introduction to Information Visualization. Information Retrieval Meets Information Visualization, LNCS 7757, pp. 1-27.

Djurcilov, S., Kim, K., Lermusiaux, P., Pang, A. T. (2001) Volume Rendering Data with Uncertainty Information. In Proc. of the 3rd Joint Eurographics, pp. 243-252.

Djurcilov, S., Kim, K., Lermusiaux, P., Pang, A. T. (2002) Visualizing Scalar Volumetric Data with Uncertainty. Computer & Graphics. Vol. 26, No. 2, pp. 239-248.

Doleisch, H., Gasser, M., Hauser, H. (2003) Interactive Feature Specification for Focus+Context Visualization of Complex Simulation Data. In Proc. of the Symposium on Data Visualization, pp. 239-248. Eurogaphics Association.

Doleisch, H., Mayer, M., Gasser, M., Priesching, P., Hauser, H. (2005) Interactive Feature Specification for Simulation Data on Time-Varying Grids. In Proc. of Simulation and Visualization, pp. 291-304.

Drebin, R. A., Carpenter, L., Hanrahan, P. (1988) Volume Rendering. Computer Graphics, Vol. 22, No. 4, pp. 65-74.

Ehlschlaeger, C. R., Shortridge, A. M., Goodchild, M. F. (1997) Visualizing Spatial Data Uncertainty Using Animation. Computers in GeoSciences. Vol. 23, No. 4, pp. 387-395.

Engel, K., Hadwiger, M., Kniss, J., Rezk-Salama, C., Weiskopf, D. (2006) Real-Time Volume Graphics. A. K. Peters Ltd.

Esty, W. W., Banfield, J. D. (2003) The Box-percentile Plot. Journal of Statistical Software, Vol. 8, No.17, pp. 1-14.

Feng, D. (2010) Visualization of Uncertainty Multivariate 3D Scalar Fields. PhD Dissertation. Department of Computer Science, University of North Carolina at Chapel Hill.

Feng, D., Kwock, L., Lee, Y., Taylor II, R. M. (2010) Matching Visual Saliency to Confidence in Plots of Uncertain Data. IEEE Transactions on Visualization and Computer Graphics. Vol. 16, No. 6, pp. 980-989.

Foulks, A., Bergeron, R. D. (2009) Uncertainty Visualization in the VisIt Visualization Environment. In Proc.of Visualization and Data Analysis. Vol. 7243, pp. 72430-72440.

Fryxell, B., Olson, K., Ricker, P., Timmes, F. X., Zingale, M., Lamb, D. Q., MacNeice, P., Rosner, R., Truran, J. W. and Tufo, H. (2000) FLASH: An Adaptive Mesh Hydrodynamics Code for Modeling Astrophysical Thermonuclear Flashes. The Astrophysical Journal Supplement Series. Vol. 131, No. 1, pp. 273-334.

Garland, M. (1999) Multiresolution Modeling: Survey and Future Opportunities. In Proc. of Eurograohics, State of the Art Report, pp. 111-131.

Gasser, M. (2004) Fast Focus + Context Visualization of Large Scientific Data. Technical Report. VRVis Research Center, pp. 1-8.

Ge, Y., Li, S., Lakhan, V. C., Lucieer, A. (2009) Exploring Uncertainty in Remotely Sensed Data with Parallel Coordinates. International Journal of Applied Earth Observation and Geoinformation. Vol. 11, No. 6, pp. 413-422.

Gordon, A., Glazko, G., Qiu, X., Yakovlevs, A. (2007) Control of the Mean Number of False Discoveries, Bonferroni and Stability of Multiple Testing, pp. 179-190.

Grigoryan, G. and Rheignans, P. (2004) Point-Based Probabilistic Surfaces to Show Surface Uncertainty. IEEE Transactions on Visualization and Computer Graphics. Vol. 10, No. 5, pp. 564-573.

Grigoryan, G. and Rheingans, P. (2002) Probabilistic Surface: Point Based Primitives to Show Surface Uncertainty. In Proc. of IEEE Visualization, pp. 147-153.

Guthe, S., Wand, M., Gonser, J., Straer, W. (2002) Interactive Rendering of Large Volume Data Sets. In Proc. of IEEE Visualization, pp. 53-60.

Hansen, C. D., Johnson, C. R. (2004) The Visualization Handbook. Academic Press. ISBN: 9780080481647.

Haroz, S., Heitmann, K. (2008) Seeing the Difference between Cosmological Simulations. IEEE Computer Graphics and Applications. Vol. 28, No. 5, pp. 37-45.

Haroz, S., Ma, K. L., Heitmann, K. (2008) Multiple Uncertainties in Time-Variant Cosmological Particle Data. In Proc. of IEEE Pacific Visualization Symposium, pp. 207-214.

He, H. Q., Yang, L., Xu, Q. (2011) Multidimensional Uncertainty Visualization with Parallel Coordinate and Star Glyph. International Journal of Digital Content Technology and Its Applications, Vol. 5, No. 6, pp. 412-420.

Heitmann, K., Lukic, Z., Fasel, P., Habib, S., Warren, M. S., White, M., Ahrens, J., Ankeny, L., Armstrong, R., O'Shea, B., Eicher, P. M., Spingel, V., Stadel, J. and Trac, H. (2007) The Cosmic Code Comparison Project. [Online] http://www.citebase.org/abstract?id=oai:arXiv.org:0706.1270.

Heitmann, K., Ricker, P. M., Warren, M. S., Habib, S. (2005) Robustness of Cosmological Simulations I: Large Scale Structure. The Astrophysical Journal. Vol. 160, pp. 28-58.

Hinchley, K., Tullio, J., Pausch, R., Proffit, D., Kassell, N. (1997) Usability Analysis of 3D Rotation Techniques. In Proc. of ACM UIST, pp. 1-10.

Hintze, J. L., Nelson, R. D. (1998) Violin Plots: A Box Plot-density Trace Synergism. The American Statistician, Vol. 52, No. 2, pp. 181-184.

Inselberg, A. (1985) The Plane with Parallel Coordinates. The Visual Computer, Vol. 1, No. 4, pp. 69-91.

Inselberg, A. (2009) Parallel Coordinates: Visual Multidimensional Geometry and Its Applications. Springer. ISBN 978-0387215075.

Inselberg, A., Dimsdale, B. (1990) Parallel Coordinates: A Tool for Visualizing Multi-dimensional Geometry. In Proc. of the 1st Conference on Visualization, pp. 361-378.

Interrante, V. (2000) Harnessing Natural Textures for Multivariate Visualization. IEEE Computer Graphics and Applications. Vol. 20, No. 6, pp. 6-11.

Johansson, J., Ljung, P., Jern, M., Cooper, M. (2005) Revealing Structure within Clustered Parallel Coordinates Displays. In Proc. of the IEEE Symposium on Information Visualization, pp. 125-132.

Johnson, C. R. and Huang, J. (2009) Distribution-Driven Visualization of Volume Data. IEEE Transactions on Visualization and Computer Graphics, Vol. 15. No. 5, pp. 734-746.

Johnson, C. R. (2004) Top Scientific Visualization Research Problems. IEEE Computer Graphics and Applications. Vol. 24, No. 4, pp. 13-17.

Johnson, C. R. and Sanderson, A. R. (2003) A Next Step: Visualizing Errors and Uncertainty. IEEE Computer Graphics and Applications. Vol. 23, No. 5, pp. 6-10.

Johnson, C. R., Moorhead, R., Munzner, T., Pfister, H., Rheingans, P., Yoo, T. S. (2006) NIH-HSF Visualization Research Challenges Report. IEEE Computer Society.

Jones, C., Ma, K. L., Ethier, S., Lee, W. L. (2008) An Integrated Exploration Approach to Visualizing Multivariate Particle Data. Journal of Computing in Science and Engineering, Vol. 10, No. 4, pp. 20-29.

Joy, K. I. (2009) Massive Data Visualization: A Survey. Mathematical Foundations of Scientific Visualization, Computer Graphics, and Massive Data Exploration, pp. 285-302.

Jurgen, H., Manner, R., Knittel, G., Strasser, W. (1995) Three Architectures for Volume Rendering. International Journal of the Eurographics Association, pp. 111-122.

Keim, D. A., Kriegel, H. P. (1994) VisDB: Database Exploration Using Multidimensional Visualization. IEEE Computer Graphics and Applications, Vol. 14, No. 5, pp. 40-49.

Kirby, R.M., Silva, C. T. (2008) The Need for Verifiable Visualization. IEEE Computer Graphics and Applications, Vol. 28, No. 5, pp. 78-83.

Kniss, J. M., Uitert, R. V., Stephens, A., Li, G. S., Tasdizen, T. (2005) Statistically Quantitative Volume Visualization. In Proc. of IEEE Visualization, pp. 287-294.

Laerd Statistics. (2013a) One-way MANOVA Using SPSS. Available at: https://statistics.laerd.com/spss-tutorials/one-waymanova-using-spss-statistics.php.

Laerd Statistics. (2013b) ANOVA with Repeated Measurements Using SPSS. Available at: https://statistics.laerd.com/spsstutorials/one-way-anova-repeated-measures-using-spssstatistics.php.

Laerd Statistics. (2013c) Two-way ANOVA in SPSS. Available at: https://statistics.laerd.com/spss-tutorials/two-way-anova-using-spss-statistics.php.

Laramee, R. S., Kosara, R. (2007) Future Challenges and Unsolved Problems. Human-Centered Visualization Environments, pp. 231-254.

Law, A. (1996) Exploring Coherency in Parallel Algorithms for Volume Rendering. PhD Thesis. Department of Computer and Information Science. The Ohio State University.

Levoy M. (2008) The Stanford Volume Data Archive. Available at: http://www-graphics.stanford.edu/data/voldata/.

Levoy, M. (1990a) Efficient Ray Tracing of Volume Data. ACM Transactions on Graphics, Vol. 9, No. 3, pp. 245-261.

Levoy, M. (1990b) Volume Rendering by Adaptive Refinement. The Visual Computer, Vol. 6, No. 1, pp. 2-7.

Li, Z. N., Drew, M. S. (2004) Fundamentals of Multimedia. School of Computing Science, Simon Fraser University. Pearson Prentice Hall.

Lin, C. (2003) Big and Little Endian. Lecture Notes. Computer Science, University of Maryland. Available at: http://www.cs.umd.edu/class/sum2003/cmsc311/Notes/Data/endian.html.

Linsen, L., Long, T. V., Rosenthal, P. (2009) Linking Multidimensional Feature Space Cluster Visualization to Multifield Surface Extraction. IEEE Computer Graphics and Applications, Vol. 29, No. 3, pp. 85-89.

Linsen, L., Long, T. V., Rosenthal, P., Rosswog, S. (2008) Surface Extraction from Multi-field Particle Volume Data Using Multi-dimensional Cluster Visualization. IEEE Transaction on Visualization and Computer Graphics, Vol. 14, No. 6, pp. 1483-1490.

Lodha, S., Sheehan, B., Pang, A. T., Wittenbrink, C. (1996) Visualizing Geometric Uncertainty of Surface Interpolants. In Proc.of the Conference on Graphics Interface, pp. 238-245.

Lorensen, W. E., Cline, H. E. (1987) Marching Cubes: A High Resolution 3D Surface Construction Algorithm. Computer Graphics, Vol. 21, No. 4, pp. 163-169.

Lundstrom, C. (2007) Efficient Medical Volume Visualization – An Approach Based on Domain Knowledge. PhD dissertation. Department of Science and Technology, Linkoping University, Sweden.

Lundstrom, C., Ljung, P., Persson, A., and Ynnerman, A. (2007) Uncertainty Visualization in Medical Volume Rendering Using Probabilistic Animation. IEEE Transactions on Visualization and Computer Graphics. Vol. 13, No. 6, pp. 1648-1655.

Ma, J., Murphy, D., O'Mathuna, C., Hayes, M., Provan, G. (2012a) Visualizing Uncertainty in Multi-resolution Volumetric Data Using Marching Cubes. In Proc. of Advanced Visual Interfaces, pp. 489-496. ACM New York. Capri Island (Naples), Italy. (index: EI)

Ma, J., Murphy, D., O'Mathuna, C., Hayes, M., Provan, G. (2012b) Iso-surface Rendering based Uncertainty Visualization for Multi-resolution Volume Data with Regular Grids. In Proc. of Information Visualization Theory and Application, pp. 751-754. SciTePress. Rome, Italy. (index: EI)

Ma, J., Murphy, D., O'Mathuna, C., Hayes, M., Provan, G. (2012c) Quantitatively Visualizing Uncertainty Information Using Volume Ray-casting Rendering, Linked View and Scatter Plot for Volumetric Data. In Proc. of the 13th Computer Graphics and Imaging, pp. 189-196. ACTA Press. Grete, Greece. (index: EI)

Ma, J., Murphy, D., O'Mathuna, C., Hayes, M., Provan, G. (2012d) Analyzing and Visualizing Multivariate Volumetric Scalar Data and Their Errors. In H. Carr and S, Czanner, editors, the 10th Theory and Practice of Computer Graphics, pp. 61-68. Eurographics Association. Rutherford Appleton Laboratory, UK. (index: EI)

Ma, J., Murphy, D., Provan, G., O'Mathuna, C., Hayes, M. (2013a) The Evaluation of Perceptual Effectiveness of Isosurface Rendering-based Uncertainty Visualization Techniques for 3D Scalar Data. In S. Czanner and W. Tang, editors, the 11th Theory and Practice of Computer Graphics, pp. 25-32. Eurographics Association. University of Bath, UK. (index: EI)

Ma, J., Murphy, D., Provan, G., O'Mathuna, C., Hayes, M. (2013b) The Evaluation of Direct Volume Rendering-Based Uncertainty Visualization Techniques for 3D Scalar Data. International Journal of Image and Graphics. World Scientific Publishing Company. (index: DBLP Bibliography Server, io-port.net) (Under review)

Ma, J., Murphy, D., Provan, G., O'Mathuna, C., Hayes, M. (2013c) Uncertainty Visualization of Material Intensity Classification in Direct Volume Rendering Using Probabilistic Query. Journal of Visual Languages and Computing. Elsevier. (index: SCI; Impact factor: 0.783) (Under review)

Ma, K. L. (2001) Large-scale Data Visualization. IEEE Computer Graphics & Applications, pp. 22-23.

MacEachren, A. M., Robinson, A., Hopper, S., Gardner, S., Murray, R., Gahegan, M., Hetzler, E. (2005) Visualizing Geospatial Informaiton Uncertainty: What We Know and What We Need to Know. Cartography and Geographic Information Science. Vol. 32, No. 3, pp. 139-160.

Mallat, S. G. (1989) A Theory for Multiresolution Signal Decomposition: The Wavelet Representation. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 11, pp. 674-693.

McCartney, T. P. (1997) A Usability Study of End-user Construction of Direct Manipulation User Interfaces. Software-Concepts and Tools, Vol. 18, No. 2, pp. 63-72.

McGill, R., Tukey, J. W., Larsen, W. A. (1978) Variations of Box Plots. The American Statistician, Vol. 32, No. 1, pp. 12-16.

Meibner, M., Pfister, H., Westermann, R., Wittenbrink, C. M. (2000) Volume Visualization and Volume Rendering Techniques. Tutorial No. 6 in Eurographics. Switzerland, pp. 1-36.

Moller, T. (2010) Visualization. Lecture Notes. School of Computer Science. Simon Fraser University.

Nathaniel, F., Ma, K. L. (2011) Reliable Visualization: Verification of Visualization based on Uncertainty Analysis. IEEE Visualization Conference.

National Institutes of Health. (2013) ImageJ – Image Processing and Analysis in Java. Available at: http://rsbweb.nih.gov/ij/.

Newman, T. S. and Lee, W. (2004) On Visualizing Uncertainty in Volumetric Data: Techniques and Their Evaluation. Journal of Visual Languages and Computing. Vol. 15, No. 6, pp. 463-491.

Novotny, M., Hauser, H. (2006) Outlier-preserving Focus+Context Visualization in Parallel Coordinates. In Proc. of the Conference on Visualization. IEEE Transactions on Visualization and Computer Graphics, Vol. 12, No. 5, pp. 893-900.

Olston, C. and Mackinlay, J. D. (2002) Visualizing Data with Bounded Uncertainty. In Proc. of the IEEE Symposium on Information Visualization, pp. 37-40.

Osorio, R. A., Brodlie, K. (2008) Contouring with Uncertainty. In Proc. of Theory and Practice of Computer Graphics, pp. 59-66.

Pang, A. T. (2008) Visualizing Uncertainty in Natural Hazards. In Risk Assessment, Modeling and Decision Support (Book), Vol. 14, pp. 261-294.

Pang, A. T., Wittenbrink, C. M., Lodha, S. K. (1997) Approaches to Uncertainty Visualization. The Visual Computer. Vol. 13, No. 8, pp. 370-390.

Papaioannou, G. (2010) Advanced Computer Graphics Course – Volume Rendering. Lecture Notes. Department of Informatics. Athens University of Economics & Business, pp. 1-46.

Patin, F. (2003) An Introduction to Digital Image Processing. Available at: http://www.dim.uchile.cl/~grios/apuntes/imagenes/ImageProc.pdf.

Persson, A., Dahlstrom, N., Engellau, L., Larsson, E. M., Brismar, T., Smedby, O. (2004) Volume Rendering Compared with Maximum Intensity Projection for Magnetic Resonance Angiography Measurements of the Abdiminal Aorta. Acta Radiologica, Vol. 45, pp. 453-459.

Piringer, H., Kosara, R., Hauser, H. (2004) Interactive Focus + Context Visualization with Linked 2D/3D Scatterplots. In Proc. of Coordinated and Multiple Views in Exploratory Visualization, pp. 49-60.

PixelsTech.net (2012) Gaussian Blur Algorithm, available at: http://www.pixelstech.net/article/index.php?id=1353768112.

Pothkow, K., Hege, H. C. (2011) Positional Uncertainty of Isocontours: Condition Analysis and Probabilistic Measures. IEEE Transactions on Visualization and Computer Graphics. Vol. 17, No. 10, pp. 1393-1406.

Pothkow, K., Weber, B., Hege, H. C. (2011) Probabilistic Marching Cubes. Computer Graphics Forum. Vol. 30, No. 3, pp. 931-940.

Potter, K. (2003) Three-dimensional Line Textures for Interactive Architectural Rendering. Master thesis. Computer Science, School of Computing, the University of Utah, USA.

Potter, K. (2010a) Open Questions in Uncertainty Visualization. Presentation in CScADS Workshop. University of Utah. July 2010, pp. 1- 27.

Potter, K. (2010b) The Visualization of Uncertainty. PhD dissertation. Computer Science, School of Computing, the University of Utah, USA.

Potter, K. (2011) Uncertainty Visualization State of the Art. Presentation in USA/South America Symposium on Stochastic Modeling & Uncertainty Quantification. University of Utah. August 2011, pp. 1-24.

Potter, K., Kniss, J., Riesenfeld, R., Johnson, C. R. (2010) Visualizing Summary Statistics and Uncertainty. In Computer Graphics Forum, Vol. 29, No. 3, pp. 823-831.

Potter, K., Rosen, P., Johnson, C. R. (2012) From Quantification to Visualization: A Taxonomy of Uncertainty Visualization Approaches. In Andrew Dienstfrey and Ronald Boisvert, editors. Uncertainty Quantification in Scientific Computing, Vol. 377, pp. 226-249. Springer Berlin Heidelberg.

Potter, K., Wilson, A., Bremer, P. T., Williams, D., Doutriaux, C., Pascucci, V., Johnson, C. R. (2009) Ensemble-vis: A Framework for the Statistical Visualization of Ensemble Data. In Proc. of IEEE Workshop on Knowledge Discovery from Climate Data: Prediction, Extremes, pp. 233-240.

Raeder, J. (1995) Global MHD Simulations of the Dynamics of the Magnetosphere: Weak and Strong Solar Wind Forcing. In Proc. of the Second International Conference on Substorms, pp. 561-568.

Rhodes, P.J., Laramee, R. S., Bergeron, R. D., Sparr, T. M. (2003) Uncertainty Visualization Methods in Isosurface Rendering. In Proc. of Eurographics, pp. 83-88. The Eurographics Association.

Rickenberg, R., Reeves, B. (2000) The Effects of Animated Characters on Anxiety, Task Performance, and Evaluation of User Interfaces. In Proc. of Computer-Human Interaction, pp. 49-56.

Roettger, S. (2012) The Volume Library, available at: http://schorsch.efi.fh-nuernberg.de/data/volume/.

Sanyal, J., Zhang, S., Bhattacharya, G., Amburn, P., Moorhead, R. J. (2009) A User Study to Compare Four Uncertainty Visualization Methods for 1D and 2D Datasets.

IEEE Transactions on Visualization and Computer Graphics. Vol. 15, No. 6, pp. 1209-1218.

Sanyal, J., Zhang, S., Dyer, J., Mercer, A., Amburn, P., Moorhead, R. J. (2010) Noodles: A Tool for Visualization of Numerical Weather Model Ensemble Uncertainty. IEEE Transactions on Visualization and Computer Graphics. Vol. 16, No. 6, pp. 1421-1430.

Skeels, M., Lee, B., Smith, G., Robertson, G. (2010) Revealing Uncertainty for Information Visualization. Information Visualization. Vol. 9, No. 1, pp. 70-81.

Spence, B. (2007) Information Visualization: design for interaction (2nd edition). Imperial College, London, Prentice Hall.

Swan, J. E. (1998) Object Order Rendering of Discrete Objects. PhD Thesis. Department of Computer and Information Science. The Ohio State University.

Tang, S. Z. (1999) 3D Data Visualization. Tsinghua University Press.

Tao, Y., Liu, Y., Friedman, C., Lussier, Y. A. (2004) Information Visualization Technique in Bioinformatics during the Postgenomic Era, pp. 237-245.

Theisel, H. (2005) Scientific Visualization – Visualization of Multiparameter Data. Lecture Notes. Max Planck Institute for Informatics. Saarland University. Germany, pp. 1-14.

Thompson, D., Levine, J. A., Bennett, J. C., Bremer, P. T., Gyulassy, A., Pascucci, V., Pebay, P. P. (2011) Analysis of Large-scale Scalar Data Using Hixels. In Proc. of IEEE Symposium on Large-scale data Analysis and Visualization, pp. 23-30.

Thomson, J., Hetzler, B., MacEachren, A., Gahegan, M., and Pavel, M. (2005) A Typology for Visualizing Uncertainty. In Proc. of the SPIE, Visualization and Data Analysis, pp.146-157.

Tufte, E. R. (1983) The Visual Display of Quantitative Information. Graphics Press.

Tukey, J. W. (1977) Exploratory Data Analysis. Addison-Wesley.

Warren, M. S. and Salmon, J. K. (1993) A Parallel Hashed Oct-tree N-body Algorithm. Supercomputing, pp. 12-21.

Westermann, R. (1994) A Multiresolution Framework for Volume Rendering. In Proc. Symposium on Volume Visualization, pp. 51-58.

Wijk, J. J. V. (2005) The Value of Visualization. IEEE Visualization, pp. 79 – 86.

Wijk, J. J. V. (2006) Views on Visualization. IEEE Transactions on Visualization and Computer Graphics, Vol. 1, No. 4, pp. 421-432.

Wittenbrink, C., Pang, A. T., Lodha, S. (1995) Verity Visualization: Visual Mapping. Technical Report, University of California at Santa Cruz.

Wolfram Research Inc. (2014) Scatter Diagram, available at: http://mathworld.wolfram.com/ScatterDiagram.html.

Wong, P. C., Bergeron, R. D. (1994) 30 Years of Multidimensional Multivariate Visualization. In Proc. of Scientific Visualization, pp. 3-33.

Wong, P. C., Bergeron, R. D. (1995) Authenticity Analysis of Wavelet Approximation in Visualization. In Proc. of IEEE Visualization, pp. 184-191.

Wong, P. C., Shen, H. W., Johnson, C. R., Chen, C. M., Ross, R. B. (2012) The Top 10 Challenges in Extreme-scale Visual Analytics. IEEE Computer Graphics and Applications. Vol. 32, No. 4, pp. 63-67.

Xie, Z. X., Huang, S. P., Ward, M. O., Rundensteiner, E. A. (2006) Exploratory Visualization of Multivariate Data with Variable Quality. In Proc. of IEEE Symposium on Visual Analysis Science and Technology, pp. 183-190.

Yagel, R. (1996) Classification and Survey of Algorithms for Volume Viewing. SIGGRAPH tutorial notes.

Yang, N. (2006) A Service-Oriented Visualization of Medical Data. Master thesis. Computer Science, the University of Amsterdam.

Zehner, B., Watanabe, N., Kolditz, O. (2010) Visualization of Gridded Scalar Data with Uncertainty in Geosciences. Computers & Geosciences. Vol. 36, No. 10, pp. 1268-1275.