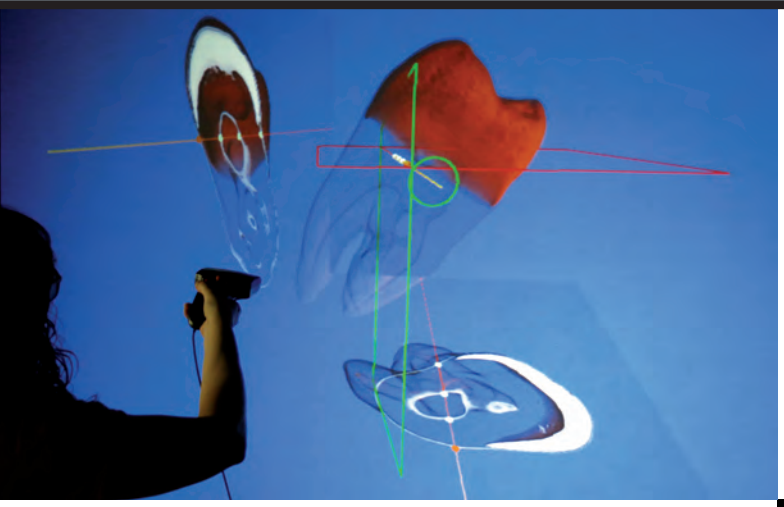
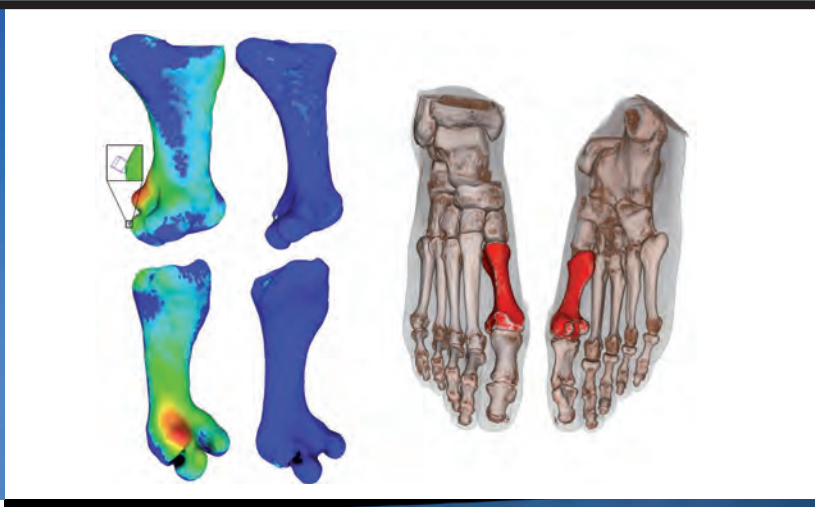
Doctoral thesis

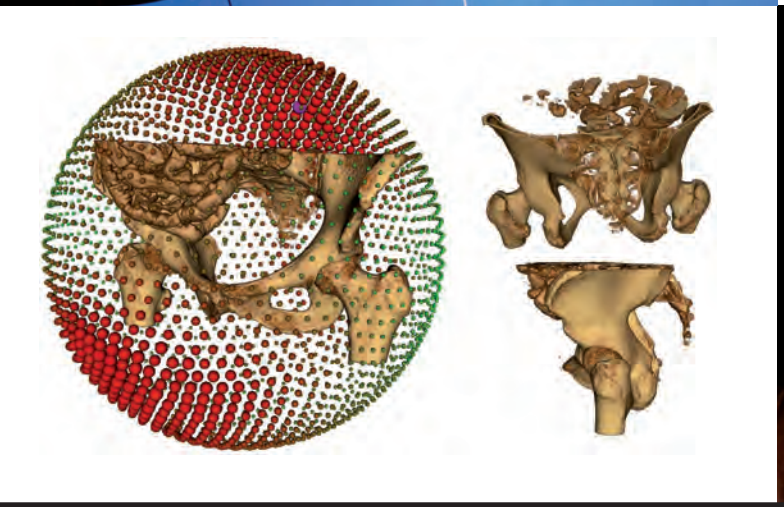
Advanced interaction techniques for medical models

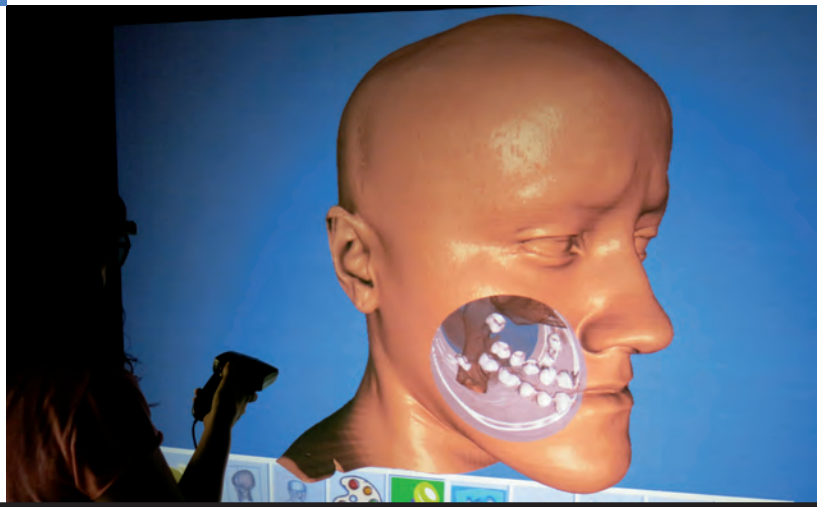
Eva Monclús Lahoya

Advisors: Isabel Navazo and Pere-Pau Vázquez









PhD Programme in Computing
Universitat Politècnica de Catalunya

Advanced interaction techniques for medical models

Doctoral Thesis

Eva Monclús Lahoya



Universitat Politècnica de Catalunya

Department of Software (LSI)

PhD Programme in Computing

Advisors: Isabel Navazo and Pere Pau Vázquez

Barcelona, June 2014

ABSTRACT

Advances in *Medical Visualization* allows the analysis of anatomical structures with the use of 3D-models reconstructed from a stack of intensity-based images acquired through different techniques, being Computed Tomography (CT) or Magnetic Resonance (MR) modalities two of the most common.

A general medical volume graphics application usually includes an exploration task which is sometimes preceded by an analysis process where the anatomical structures of interest are first identified.

The main objective of this thesis is the improvement of the user experience in the analysis and exploration of medical datasets. This improvement involves the development of efficient algorithms designed both under a user-centered perspective and taking into account the new computing capabilities of modern graphics hardware, in order to obtain high quality results in real-time.

On the analysis side, we have focused on the identification (*segmentation*) of the bones at joints, which is particularly challenging because the bones are very close to each other and their boundaries become ambiguous in CT images. We have concentrated our efforts on reaching maximum automation of the overall process. The proposed algorithm uses an example mesh of the same bone that has to be segmented, usually from a different person, to drive the segmentation process. The algorithm is based on an energy minimization scheme to deform the initial example mesh while following the well-defined features of the volume data to be segmented in a local and adaptive way. With this approach, the resulting mesh adapts to the volume features in the areas which can be unambiguously segmented, while taking the shape of the example mesh in regions which lack of relevant volume information.

We also present contributions on three different aspects of the exploration task: a best-view determination system and centering in virtual reality environments, a *focus+context* technique and a point selection method.

In medical practice it would often be very useful to have access to a quick pre-visualization of the involved medical dataset. We have proposed a new system which allows users to obtain a set of representative views in a short time and permits the generation of inspection paths at almost no extra cost. The technique relies on the use of a *Multiscale Entropy measure* for the generation of good viewpoints and uses a complexity-based metric, the *Normalized Compression distance*, for the calculation of the representative views set. Our proposal works upon a model (a raw volume dataset) classified through the definition of a transfer function. Starting from this minimal information, it automatically generates, both a set of representative views of the model and an exploration path that allows users to get an initial comprehension of the volume dataset before beginning the exploration task.

In the exploration of medical datasets, it is difficult to simultaneously visualize interior and exterior structures because the structures are commonly quite complex and it is easy to lose the context. We have developed a new interaction tool, the *Virtual Magic Lantern*, tailored to facilitate volumetric data inspection in a Virtual Reality environment. It behaves like a lantern whose illumination cone determines the region of interest. The region of interest is rendered using another transfer function providing a feature rich volume inspection experience. It addresses the occlusion management problem and facilitates the inspection of inner structures without the total elimination of the exterior structures, offering in this way, a *focus+context-based* visualization of the overall structures.

Finally, the analysis of medical datasets may require the selection of 3D points for measurements involving anatomical structures. Although there are well-established 3D object selection techniques for polygonal models, there is a lack of techniques specifically developed for volume datasets. We present a new selection technique for Virtual Reality setups which allows users to easily select anchor points in non necessarily segmented volume datasets rendered using Direct Volume Rendering. This new metaphor is based on the use of a ray emanating from the user, whose trajectory is enriched with its points of intersection with the on-the-fly determination of the isosurfaces along the ray path. Additionally, a visual feedback of the ray selection is offered through the use of two helper mirror views, in order to show occluded candidate points that would otherwise be invisible to the user without posterior and ad-hoc manipulation.

RESUM

Els avenços en la investigació en el camp de recerca de *Medical Visualization* permeten l'anàlisi de models volumètrics tridimensionals d'estructures anatòmiques, construïts a partir d'imatges mèdiques obtingudes mitjançant diferents tècniques de captació, essent la Tomografia Computeritzada (TC) una de les més habituals.

Generalment, les aplicacions informàtiques orientades a l'anàlisi d'aquest tipus de models, be sigui pel suport al diagnòstic, simuladors mèdics o per la planificació de processos quirúrgics, permeten l'exploració interactiva dels models volumètrics. Depenent de les estructures anatòmiques que es requereixi analitzar, pot ser necessari realitzar un procés d'identificació (segmentació) de les estructures per tal de possibilitar la seva posterior inspecció.

L'objectiu d'aquesta tesi és millorar l'eficiència i l'experiència de l'usuari, tant en la tasca de segmentació com d'exploració. Per tal d'assolir-ho, s'han desenvolupat diversos algorismes dissenyats sota una perspectiva centrada en l'usuari i fent servir els darrers avenços tecnològics de las targes gràfiques, el que ens permet obtenir resultats visuals de màxima qualitat en temps real.

Respecte de la tasca de segmentació, ens hem centrat en la identificació d'ossos ubicats en les articulacions, en models capturats mitjançant TC. La identificació d'aquest tipus d'estructures fent servir les tècniques clàssiques de segmentació pot arribar a ser molt feixuga, degut a que pot requerir de molta intervenció per part de l'usuari. La recerca realitzada en el marc de la tesi s'ha enfocat en assolir la màxima automatització possible del procés. La tècnica proposada empra una malla triangular d'exemple de l'os que es vol segmentar, que es farà servir per guiar tot el procés de segmentació. L'algorisme deforma de forma local i adaptativa aquesta malla, adaptant-la a la informació present en el model volumètric en les parts que de forma inambigua és pot determinar la seva frontera, i respectant al màxim la forma original de la malla en aquelles zones en el que el model volumètric presenta algun tipus d'incertesa en la definició de la frontera, ja sigui perquè l'estructura òssia apareix totalment unida a altres estructures de l'articulació o degut a que la informació capturada no presenta un frontera ben contrastada.

Respecte al procés d'exploració, aquesta tesis presenta resultats en dues vessants diferents. Per una banda, la generació automàtica de previsualitzacions de models volumètrics i per l'altra, el desenvolupament de noves tècniques d'interacció que facilitin la seva exploració en entorns de realitat virtual.

Oferir a l'usuari una previsualització ràpida del model volumètric que ha d'inspeccionar, pot ser de molta utilitat en la pràctica clínica. Aquesta tesi presenta una nova tècnica que permet obtenir en un temps acceptable un conjunt de vistes representatives, així com la generació automàtica d'una animació a l'entorn del model que facilita a l'usuari una ràpida comprensió del mateix. La tècnica desenvolupada utilitza una formulació de l'*entropia multiescala* pel càlcul de vistes informatives del model volumètric. A partir del conjunt de les vistes calculades i l'ús de la *distància de compressió normalitzada*, una mètrica del camp de la teoria de la complexitat, es calcula un subconjunt de vistes representatives.

Per altre banda, en l'exploració de models mèdics pot ser difícil la visualització simultània d'estructures internes i externes. Per abordar aquest problema s'ha desenvolupat una nova tècnica d'interacció anomenada *Virtual Magic Lantern*, pensada per a facilitar la inspecció d'aquests models en entorns de realitat virtual. Aquesta metàfora d'interacció es comporta com una llanterna de mà guiada per l'usuari, on el seu feix de llum determina una regió d'interès sobre el model volumètric. Aquesta regió d'interès serà visualitzada emprant una funció de transferència diferent a l'emprada per la resta del model, permetent d'aquesta manera la visualització de les estructures internes sense eliminar totalment la resta del model.

Per últim, en l'anàlisi de models volumètrics pot ser necessària la selecció de punts concrets per a poder realitzar algun tipus de mesuraments entre estructures anatòmiques. Depenent de com es visualitzi el model, determinar quin punt exactament vol seleccionar l'usuari pot no tenir un resultat únic. Per a solucionar aquest problema, aquesta tesi presenta una nova metàfora d'interacció en entorns de realitat virtual per a la selecció de punts en un model volumètric no necessàriament segmentat. Aquesta tècnica es basa en l'ús d'un raig originat en la mà de l'usuari, sobre el que es visualitzen els punts d'intersecció de les estructures anatòmiques que travessa. Donat que la superfície d'aquestes estructures no està explícitament definida en el model volumètric, s'ha requerit desenvolupar un càlcul ràpid i precís de la intersecció del raig amb aquestes estructures. Per oferir una visualització dels punts calculats sense cap tipus d'oclusió per part de les estructures anatòmiques visibles en el model volumètric, s'ha afegit a la visualització global la visualització de dos panells auxiliars en els quals es mostra el mateix model volumètric retallat de tal manera que els punts d'intersecció calculats siguin totalment visibles. D'aquesta forma, es facilita a l'usuari la selecció dels punts sense tenir que realitzar cap tipus de manipulació del model per tal d'obtenir una visualització en la qual els punts calculats siguin visibles.

RESUMEN

Los avances en la investigación en el área de *Medical Visualization* permiten el análisis de modelos volumétricos tridimensionales de estructuras anatómicas. Estos modelos se construyen a partir de imágenes médicas obtenidas mediante diferentes técnicas de captación, siendo la Tomografía Computerizada (TC) una de las más frecuentes.

Habitualmente, las aplicaciones informáticas orientadas al análisis de este tipo de modelos, bien sean para el soporte al diagnóstico, simuladores médicos o la planificación de procesos quirúrgicos, permiten la exploración interactiva de los modelos volumétricos. Dependiendo de las estructuras anatómicas que se precise analizar, puede ser necesario realizar un proceso de identificación (segmentación) de las estructuras anatómicas para posibilitar su posterior inspección.

El objetivo principal de esta tesis ha consistido en el desarrollo de nuevas técnicas informáticas que mejoren la experiencia del usuario en los procesos tanto de segmentación como de exploración de un modelo volumétrico. Para alcanzar dicho objetivo, ha sido necesario el desarrollo de algoritmos eficientes diseñados teniendo particularmente en cuenta al usuario final y explotando los últimos avances en la tecnología de las tarjetas gráficas para poder obtener resultados visuales de la máxima calidad en tiempo real.

En lo relativo al proceso de segmentación, nos hemos centrado en la identificación de las estructuras óseas ubicadas en articulaciones, en modelos capturados mediante TC. La identificación de este tipo de estructuras usando los métodos tradicionales de segmentación puede llegar a ser muy tediosa, debido a que puede necesitarse mucha intervención por parte del usuario. La investigación llevada a cabo ha tenido como objetivo principal el maximizar el grado de automatización en el proceso de segmentación de este tipo de estructuras. La técnica propuesta parte de un ejemplo de la estructura ósea (malla triangular) que se quiere segmentar, generada a partir de los datos o bien de otra persona o bien de la misma persona en otras circunstancias. A partir de este ejemplo el algoritmo deforma la malla de manera local y adaptativa, adaptandola a la información presente en el modelo volumétrico en aquellas zonas donde la frontera de la estructura está definida de forma no ambígua, y respetando la forma de la malla original en aquellas otras zonas en las cuales el modelo volumétrico presenta algún tipo de incertidumbre en la definición de la frontera, ya sea porque la estructura ósea aparece totalmente unida a otras estructuras óseas de la articulación o debido a que la información capturada no presenta una frontera bien contrastada.

En lo relativo al proceso de exploración, esta tesis presenta resultados en dos vertientes distintas.

Por un lado, la generación automática de una previsualización del modelo volumétrico y por el otro lado, el desarrollo de nuevas técnicas de interacción que faciliten la exploración de modelos volumétricos en entornos de realidad virtual.

Ofrecer al usuario una previsualización rápida del modelo volumétrico que ha de inspeccionar, puede ser de mucha utilidad en la práctica clínica. Esta tesis presenta un nuevo sistema que permite obtener en un tiempo razonable un conjunto de vistas representativas del modelo volumétrico, así como la generación de una animación alrededor del modelo que facilita al usuario una rápida comprensión del mismo. Las técnicas desarrolladas se basan en el uso de la *entropía multiescala* para el cálculo de vistas informativas del modelo volumétrico. A partir del conjunto de vistas calculadas y mediante el uso de la *distancia de compresión normalizada*, una métrica de Teoría de la Complejidad, se calcula un subconjunto de vistas representativas del modelo volumétrico.

Por otro lado, en la exploración de modelos volumétricos puede ser difícil visualizar simultáneamente estructuras anatómicas internas y externas. Esto es debido a que las estructuras son bastantes complejas, y es fácil perder la referencia respecto a otras estructuras anatómicas. En esta tesis se ha desarrollado una nueva técnica de interacción, bautizada como *Virtual Magic Lantern*, orientada a facilitar la inspección de modelos volumétricos en entornos de realidad virtual. Esta nueva metáfora de interacción se comporta como una linterna de mano guiada por el usuario, cuyo haz de luz define sobre el modelo volumétrico una región de interés. Esta región de interés será visualizada utilizando una función de transferencia diferente a la usada para el resto del modelo, posibilitando de esta manera la inspección de estructuras internas sin eliminar totalmente el resto del modelo.

En el análisis de modelos médicos puede ser necesaria la selección de puntos concretos para poder realizar algún tipo de medición entre estructuras anatómicas. Dependiendo del tipo de visualización del modelo, determinar qué punto exactamente quiere seleccionar el usuario puede no tener un resultado único. Para solucionar este problema, se presenta una nueva metáfora de interacción en entornos de realidad virtual para la selección de puntos anatómicos de un modelo volumétrico no necesariamente segmentado. Esta técnica se basa en el uso de un rayo originado en la mano del usuario, sobre el que son visualizados los puntos de intersección de las estructuras anatómicas que atraviesa. Dado que la superficie de estas estructuras anatómicas no está explícitamente representada en el modelo volumétrico, se ha requerido desarrollar un cálculo preciso y rápido de la intersección del rayo con estas estructuras. Para ofrecer una visualización de los puntos calculados sin ningún tipo de oclusión por parte de las estructuras anatómicas existentes en el modelo, se ha añadido a la visualización global la visualización de dos paneles auxiliares en los cuales se muestra el mismo modelo volumétrico recortado de tal manera que sean completamente visibles el conjunto de los puntos. De esta forma, se facilita al usuario la selección de los puntos calculados sin tener que realizar ningún tipo de manipulación del modelo para poder obtener una visualización en la que los puntos calculados sean visibles.

ACKNOWLEDGEMENTS

The work presented in this thesis has been possible thanks to the help of a lot of people. Specially my advisors: Isabel Navazo and Pere-Pau Vázquez. Both know how much I appreciate them. This is my thesis, but, as I always say, this is also our thesis: without their help it would not have been possible to finish it. They have encouraged and supported me in all the dimensions of its development, from technical and intellectual issues to personal and emotional aspects.

My special thanks also go to Pere Brunet, Toni Chica and Àlvar Vinacua. It was a pleasure to work with them in the development of one of the contributions of my thesis. I also want to thank Carlos Andújar, who provided his experience in the area of Virtual Reality whenever he was called for.

All the medical doctors I have collaborated with deserve to be mentioned as well. When they share their problems or their wishes with us, they put our minds to work looking for solutions. Additionally, some of them have been so kind to participate in the evaluation of most of the contributions presented in this thesis. To all of them, my most sincere acknowledgment.

Since I began to work in the *Secció d'Informàtica Gràfica* (SIG) lab until now in the *Research Center for Visualization, Virtual Reality and Graphics Interaction* (ViRVIG), I have met different people, from research collaborators, to simply lab mates, and some friends too. I appreciate the different kind of support they have provided, from coding help, ideas, and emotional support to the funny times we have spent together. Some of them have also volunteered for taking part in different user studies. I am really grateful to all of them.

Finally, my special thanks for Manu Vivó, who has been such a good and patient teacher when my maths failed.

CONTENTS

Contents ix

List of Tables xii

List of Figures xiii

1 Introduction 1
- 1.1 Motivation . . . 4
- 1.2 Addressed problems and contributions . . . 4
- 1.3 About this document . . . 5

2 Example-Guided Segmentation 7
- 2.1 Related work . . . 8
- 2.2 Example-guided bone joints segmentation . . . 10
  - 2.2.1 Formal problem statement . . . 12
- 2.3 Implementation details . . . 13
  - 2.3.1 Generation of the driving distance field . . . 13
  - 2.3.2 Generation of the Laplacian coordinates . . . 17
  - 2.3.3 Example-guided segmentation algorithm . . . 18
- 2.4 Results . . . 20
- 2.5 Conclusions . . . 26

3 Automatic Good Viewpoint Selection 29
- 3.1 Related work . . . 30
- 3.2 Process overview . . . 33
- 3.3 Best view determination . . . 34
  - 3.3.1 Preliminaries . . . 34
  - 3.3.2 Application of the Multiscale Entropy to Volume Graphics . . . 35
  - 3.3.3 Efficient adaptive algorithm . . . 39
  - 3.3.4 Results . . . 42
    - Comparison with other methods in literature . . . 44

3.4 Representative views selection . . . 45
3.4.1 Preliminaries . . . 45
3.4.2 Representative views selection algorithm . . . 46
3.4.3 Results . . . 47
Optimization . . . 48
3.5 Exploration path construction . . . 50
3.5.1 Exploration path construction algorithm . . . 50
3.5.2 Results . . . 51
3.6 Conclusions . . . 52

4 Virtual Magic Lantern: an interaction metaphor for enhanced medical data inspection 55
4.1 Related Work . . . 56
4.2 The Virtual Magic Lantern metaphor . . . 58
4.2.1 Definition of the region of interest with the lantern . . . 60
4.2.2 Lantern guidance . . . 62
4.3 Implementation details . . . 62
4.3.1 VML implementation . . . 63
4.3.2 VMW implementation . . . 65
4.4 Results . . . 68
4.4.1 VML in medicine . . . 70
4.4.2 User studies . . . 72
Initial findings . . . 72
Findings in the medical domain . . . 73
4.5 Conclusions . . . 76

5 DAAPMed: a data-aware picking technique for medical models 79
5.1 Related work . . . 80
5.2 A data-aware anchor point selection for medical models . . . 82
5.2.1 Heisenberg effect . . . 85
5.3 Implementation details . . . 86
5.3.1 Automatic detection of the ray-isosurface intersection . . . 86
5.3.2 Helper Views: Visual feedback framework . . . 88
5.3.3 Performance . . . 90
5.4 User study . . . 91
5.4.1 Design details of the Clipping Plane technique in Virtual Reality . . . 92
5.4.2 Test design . . . 93
Data preparation . . . 94
Subjects and procedure . . . 95
5.4.3 Statistical results . . . 96

$Test_1$ results 97
$Test_2$ results 100
5.4.4 Post-questionnaire results 102
5.4.5 Discussion 103
Shake filtering 103
5.5 Conclusions 104

6 Conclusions **107**
6.1 Future research 109
6.2 Publications 111

Bibliography **113**

List of Tables

2.1 Information of the chosen bones to segment, and characteristics of the reference meshes. . 21
2.2 Information of the volume models to segment and the structure which will be segmented in each of them. . . . 21
2.3 Execution times for the algorithm . . . 26
2.4 Quantitative results for the algorithm . . . 26

3.1 Resolution and voxel dimension of the tested models . . . 36
3.2 The best and worst views according to the Multiscale Entropy measure for different viewports 37
3.3 The best and worst views information according to our measure for different viewports . . . 38
3.4 Performance of the adaptive approach with respect to the loading time . . . 42
3.5 Comparison of the obtained views using the brute force and the adaptive approaches 43
3.6 Performance of the representative views selection algorithm . . . 48
3.7 Performance of the representative views selection algorithm with the RGB and the grey scale strategies . . . 48
3.8 Comparison of the representative views selection with the RGB and the grey scale strategies 49

4.1 Performance of the classical ray casting with different data sets . . . 68
4.2 Performance of the VML and VMW metaphors . . . 69
4.3 Questionnaire to be filled out by the participants. . . . 74

5.1 Performance of the DAAPMed technique . . . 91
5.2 Statistical analysis of the relevant variables for $Test_1$ . . . 97
5.3 Statistical analysis of the relevant variables for $Test_2$ . . . 101
5.4 Results of the evaluation performed to analyze the steady-hands mechanism . . . 104

List of Figures

1.1 Stages in a general medical volume graphics application. . . . 2

2.1 2D views of complex bony structures . . . 8
2.2 Scheme of the whole segmentation process . . . 11
2.3 Steps for the driving distance field computation. . . . 14
2.4 Preprocess steps of the algorithm . . . 16
2.5 Step_1 of the optimization algorithm . . . 18
2.6 Results of the algorithm applied to the *Metatarsal* bone . . . 22
2.7 Resulting segmentation of the *Metatarsal* bone shown in the context of the whole dataset . 22
2.8 Results of applying the algorithm to the *Calcaneus* bone . . . 23
2.9 Results of applying the algorithm to the *Phalanx* bone . . . 24
2.10 Resulting segmentation of the *Phalanx* bone shown in the context of the whole dataset. Left image shows the initial mesh after the initial coarse registration and right image shows the resulting mesh. . . . 24
2.11 Results of the algorithm applied to the *5-Metatarsal* bone . . . 25
2.12 Resulting segmentation of the 5-*Metatarsal* bone shown in the context of the whole dataset 25

3.1 Workflow of the proposed system . . . 33
3.2 Examples of the entropy measure for a dense set of views . . . 36
3.3 Mesh subdivision scheme . . . 39
3.4 Estimation of the maximum reachable entropy inside a triangle . . . 40
3.5 Comparison of the obtained views with other methods from literature . . . 44
3.6 Examples of our similarity measure for the head model . . . 47
3.7 Scheme followed to build the two exploration paths . . . 50
3.8 Exploration paths around some of the tested models . . . 51
3.9 Analysis of a region of interest around the kidney . . . 52

4.1 Image of *Ars Magna Lucis et Umbrae* where the Magic Lantern is depicted. . . . 58
4.2 The Virtual Magic Lantern metaphor. . . . 59
4.3 Inspection of a medical dataset using the VML and the VMW metaphors. . . . 60
4.4 Cylindrical and prism shapes of the interaction tool. . . . 61

4.5 Example of the use of a finite version of the lantern shapes . . . 61
4.6 Lantern guidance scheme. . . . 62
4.7 The two cone-based focal inspection regions. . . . 63
4.8 VML ray casting scheme. . . . 64
4.9 Boundary shading for noise avoidance. . . . 64
4.10 Implementation of the VMW metaphor. . . . 66
4.11 Examples of the VMW metaphor. . . . 67
4.12 Two models rendered with two different rendering styles. . . . 67
4.13 Application layout. . . . 70
4.14 Showroom presented at the Hospital Universitari Arnau de Vilanova (2010). . . . 71
4.15 Inspection of the model used in the user study. . . . 73
4.16 Post-questionnaire results from questions Q_1, Q_2, Q_3, Q_6 and Q_7 . . . 75
4.17 Post-questionnaire results from questions Q_4 and Q_5 . . . 75
4.18 Post-questionnaire results of questions evaluating personal preference. . . . 76

5.1 Block diagram illustrating the workflow of the DAAPMed technique . . . 84
5.2 User interacting with a model using DAAPMed. The blow-up in the figure magnifies the visualization of the candidate selection points in one of the Helper Views. . . . 85
5.3 Illustration of the add-in incorporated to prevent Heisenberg effect of spatial interaction . . . 86
5.4 Plots showing the isosurfaces detected with our measure . . . 88
5.5 Helper Views: Visual feedback framework . . . 89
5.6 Immersive virtual reality setup used for the user study. . . . 92
5.7 Porting the clipping plane technique for point selection to VR. . . . 93
5.8 The training and testing datasets used in $Test_1$ . . . 94
5.9 Model used in task T_3 . . . 95
5.10 Scheme followed in $Test_1$ and $Test_2$ . . . 96
5.11 Images showing the description of task T_2 . . . 96
5.12 Results of the completion task timings for $Test_1$ . . . 97
5.13 Input device footprints chart . . . 98
5.14 Input device footprints split chart . . . 99
5.15 User footprint chart . . . 100
5.16 Accuracy by technique . . . 101
5.17 Results of the completion task timings for $Test_2$ . . . 101
5.18 Post-questionnaire results . . . 102

6.1 Image showing the integration of DAAPMed technique in VML metaphor. . . . 110
6.2 Images showing the *Zoom-in-Place* technique. . . . 110

List of Algorithms

1 Optimization algorithm. 20
2 Adaptative subdivision algorithm. 41
3 Fragment shader pseudocode of the VML metaphor 65
4 Shake filtering algorithm. 103

CHAPTER 1

INTRODUCTION

Volume Graphics is concerned with the modeling, synthesis and manipulation of volumetric datasets. The typical structure of a volumetric dataset is a delimited region of the space in which some measurements have been taken at several sampling positions. There are a lot of fields in science and industry for which the data used has a volumetric nature. For example, the comprehension of the air flow around planes and cars in aerodynamics, or the analysis of seismic data for the study of a terrain in geoscience. In the medical domain, the analysis of 3D medical images has become commonplace, partially replacing the classical 2D X-ray images. Our research has focused on the area of medical applications, although some of the proposed algorithms could be adapted to other fields.

Medical Visualization is a sub-field of *Volume Graphics* concerned with applications based mainly on the visualization of 3D medical images to ease clinical diagnosis, treatment, therapy planning and medical education. The basic acquisition modalities are: Computed tomography (CT), Magnetic Resonance (MR), Ultrasound Imaging (US) and Nuclear Imaging (NI) such as PET and SPECT modalities [PB13]. Medical image data usually consists of a stack of individual images. Each image represents a thin slice of the captured anatomical structures. Volumetric data combines individual images into a 3D representation on a 3D grid. In a more formal way, a volumetric dataset is a set V of samples (x, y, z, v), where v represents the value of some measurable data property at a certain 3D location (x, y, z). The value v can be either mono-valued, as in X-ray absorption in CT, or multi-valued, representing, for instance, results from multiple scanning modalities, such as anatomical (CT, MRI) and functional (PET, fMRI). The value v may also be scalar, such as CT, or vectorial, representing for instance the fiber structure of the brain in Diffusion Tensor imaging (DTI). In addition, the volume data may be time-varying, in this case V becomes a set of samples (x, y, z, t, v). In general, the samples may be taken at purely random locations in space, but in most cases the set V is isotropic containing samples taken at regularly spaced intervals along three orthogonal axes. When the spacing between

samples along each axis is a constant – maybe a different one to each axis –, we say that the set V is anisotropic. Since the set of samples is located on a regular grid, a 3D array is typically used to store the values. This representation is usually called *Voxel Model*. The property value v at non-sampled positions is usually computed by trilinear interpolation from the voxels – sampled values.

This thesis focuses on structured rectilinear models where samples (scalar & mono-valued) are located at regularly spaced intervals. The volume dataset is considered anisotropic and the trilinear interpolation scheme is used for calculating new samples points.

The main processes involved in a general medical volume graphics application [ENMM99] are depicted in Figure 1.1.

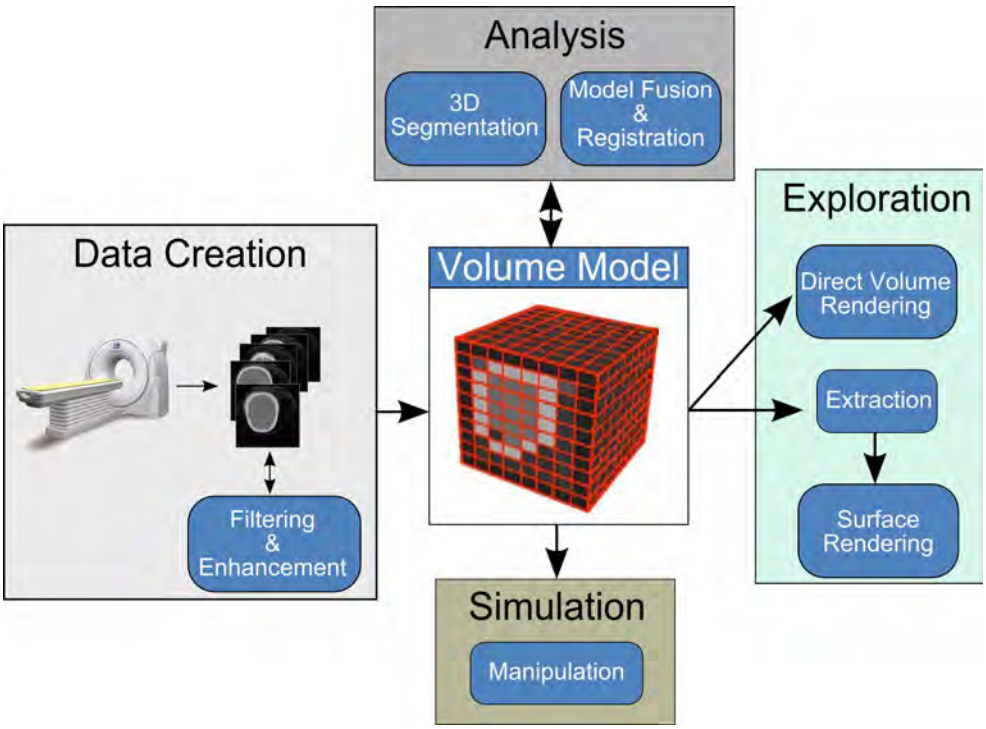


Figure 1.1: Stages in a general medical volume graphics application.

- *Data creation* refers to the processes involved in the construction, starting from the acquired data, of a volume model that can be quantified, visualized and manipulated. Once the information has been acquired, a series of 2D slices are generated. Before the construction of the *Voxel model*, the image data set may need to undergo several preprocessing steps, such as: distortion correction, and filtering enhancement. Depending on the kind of the application and the nature of the captured information, the construction of the volume dataset may consists of simply filling the *Voxel model* with the 2D image data or it may needs a more sophisticated structures such as a multiresolution approach.

- *Analysis* refers to all the processes involved in detecting structures of interest from the volumetric model and subsequently characterizing and analyzing them. In literature, this general concept of finding, extracting and characterizing is called *segmentation* [SF00]. Moreover, when more than one modality is captured, it is useful to have all the modalities referenced at the same coordinate system. This process is called *registration*. Actually, the registration problem is very close to the segmentation problem, so some of the paradigms applied to segmentation can be also applied to the registration problem. Once the different modalities have been registered, the term *fusion* refers to the establishment of a function that derive useful complimentary information from the different modalities.

- The objective of the *Exploration* process is to synthesize an image in an appropriate manner that conveys the structural and dynamic characteristics of the volume model, while also supporting the user interaction. Volume models can be visualized by directly projecting the volume data to the screen (*Direct Volume Rendering*, DVR) or by generating an intermediate representation: point set, contour set or polygonal isosurfaces, which allows the use of classical rendering algorithms, such us *surface rendering*.

- *Simulation* includes the processes involved in the manipulation of the volume model. One of its main purposes is the training of future medical doctors in surgery procedures. Simulation is also used for several surgery planning procedures. Surgery simulators have been developed to simulate the behaviour of soft tissue, the interaction of surgical devices with soft tissue, etc. In many surgical tasks, the tactile sense plays a very important role; haptic interfaces allow the simulation of tactile feedback by computing forces that represent the interaction between surgical devices and the patient's anatomy.

User interaction is required in almost all of the processes described above. For example, in the *Exploration* task, the user frequently guides the process. In the *Analysis* stage, depending on the anatomical structure users are interested in, user intervention may be necessary in order to guide the segmentation task. *Interaction* is centered on providing techniques that facilitate user's goals and tasks, designed under a user-centered perspective, having as a main objective the improvement of the usability of the interaction methods making more comfortable, easier and faster the user experience when using them.

The acceptance of computer aided applications by medical doctors critically depends on issues like performance, robustness, accuracy and usability. So that, the validation and clinical evaluation of volume graphics techniques are very important in order to guarantee its integration in the daily-work of medical doctors.

Although the use of *Virtual Reality* (VR) in medical applications is just starting, the development of some commercial VR platforms is a reality (see for example the development of the Dextroscope platform [Bra]), and some disciplines, such as neurosurgery, have adopted the use of VR in its clinical practice. In fact, there is a growing interest in interaction research of volume models for immersive

virtual environments and immersive visualizations, starting to be considered a specific sub-field in the Virtual Reality research area.

1.1 Motivation

As described in the previous section, a general medical volume graphics application usually includes an exploration task which is sometimes preceded by an analysis process where the anatomical structures of interest are first identified.

The thesis research has been centered in the context of the applications developed by the *Modeling, Visualization, Interaction and Virtual Reality* (MOVING) research group of the *Universitat Politècnica de Catalunya*, in collaboration with some specialists from the *Hospitals de la Vall d'Hebron* of Barcelona. This collaboration has led to the problems addressed and their medical application.

The main objective of this thesis is to improve the user experience in the *Analysis* and *Exploration* blocks of a medical application (see Figure 1.1). The term *user experience* involves the development of efficient algorithms designed under a user-centered perspective, and also taking the new computing capabilities into account to obtain high quality results in real-time. Concretely, this thesis has focused on improving the user experience in the *Analysis* phase by minimizing the amount of user intervention required to segment a structure of interest. Additionally, the research in this thesis has been focused on enhancing the user experience in the *Exploration* task using *Direct Volume Rendering (DVR)* both in virtual reality and desktop-based setups.

Throughout the development of all the techniques proposed in this thesis, it has been taken into account the improvement of efficiency, the quality of the obtained results and the usability of the different tasks carried out by the user.

1.2 Addressed problems and contributions

The work presented in this thesis has addressed the following problems:

- Concerning the *Analysis*, we have focused on the segmentation of bones at joints, where captured data usually exhibits unclear boundaries between different tissues which often lead to misclassification of structures. We have concentrated our efforts on reaching maximum automation of the overall process. Our proposal [CMB+12] is a model-based approach guided by deformation techniques inspired both in Geometric Processing techniques and in volume region-based information. Chapter 2 details it.

- Regarding the *Exploration*, we have focused in three different clue points:

  - Optimal selection of viewpoints is an important task in order to improve the understanding of the inspected dataset. Chapter 3 describes a new method based on entropy measures that improves the automation of the process of good viewpoints generation. The objective

is to offer an interesting set of views before starting the inspection task [VMN08]. The new technique allows users to obtain a quick previsualization of a volumetric dataset in a short time using an automatic-fashion algorithm.

– For the inspection of volumetric models in VR environments, we have tackled two different aspects:

  * In Volume Rendering, it is difficult to simultaneously visualize interior and exterior structures because the structures are commonly quite complex and it is easy to lose the context. Chapter 4 presents a new interaction tool, named *Virtual Magic Lantern*, for improving and helping users in the task of medical data inspection [MDNV09]. It addresses the occlusion management problem, facilitating the inspection of inner structures without the total elimination of the exterior structures, offering in this way a context-based visualization of the overall structures.

  * The analysis of medical datasets may require the selection of 3D points for several tasks, such as the measurements of anatomical structures. Performing this kind of task is often difficult and tedious, as well as very time consuming. Although there is a well-established field of research in 3D object selection techniques, there is a lack of techniques specifically developed for volume datasets. The main objective has been to provide an easy-to-use tool for the fast and accurate selection of 3D anchor points in VR environments. The proposed technique is called *DAAPMed* (Data-Aware Anchor Point selection tool for Medical Models) [MVN13], which is based on the use of a ray emanating from the user's hand, whose trajectory is enriched with the information on the points of intersection with the structures traversed by it. Also, in order to avoid its occlusion with the medical dataset, visual feedback of the ray position is offered through the use of mirror views. This approach is described in Chapter 5.

1.3 About this document

The remainder of this document is organized as follows: each of the addressed problems and contributions are presented as a separated chapter, where the first section briefly reviews the related work on the corresponding topic of research and next sections present the proposed technique in depth. Chapter 2 is devoted to the specific segmentation problem addressed. Chapter 3 presents the automatic selection of representative views. Chapter 4 presents the *Virtual Magic Lantern* technique. Chapter 5 describes the *DAAPMed* technique. Finally, in Chapter 6, the conclusions and the future work are discussed.

CHAPTER 2

EXAMPLE-GUIDED SEGMENTATION

In *Medical Visualization*, the segmentation process is an important, challenging, and current problem. It is of relevance for surgery planning, simulations, training and diagnosis, among other applications. Despite the advances in medical imaging systems, the complexity of anatomical structures, along with the lack of contrast, the presence of artifacts, missing data, and the fact that the sampled values do not always map bijectively to tissues, make the automatic segmentation of medical images quite complex. No single segmentation technique may identify all anatomical structures, and often medical experts must guide the segmentation with their knowledge about anatomy.

We have focused our research in the specific domain of the segmentation of bones located at joints. This kind of segmentation is particularly challenging because these bones are very close to each other, which can make bone boundaries ambiguous in Computed Tomography (CT) images. This chapter presents a model-based algorithm guided by deformation techniques inspired both by Geometric Processing techniques and by volume region-based information. We have developed a quasi-automatic technique which provides an accurate segmentation of the bone structure of interest.

The rest of the chapter is organized as follows. Section 2.1 summarizes the relevant literature in the area of volume segmentation focused on the subject we have addressed. Section 2.2 presents an overview of the algorithm. Section 2.3 details its implementation. Results are discussed in Section 2.4. Finally, Section 2.5 presents the conclusions.

2.1 Related work

Due to the high complexity of the segmentation process, a vast number of papers have addressed this problem under very different perspectives. Some of them focus on the segmentation of a specific structure, for example the segmentation of the liver surface [SDM+01], while others try to cover a larger number of different structures as in [Erd12]. In [Erd12] readers can find a good survey of the segmentation techniques.

Many segmentation methods have been presented to provide either automated or semi-automated segmentation of bones in CT images. It is traditionally accomplished by thresholding [KEK03] and seeded region growing [AB94]. This techniques are fairly successful in general since bony structures have greater Hounsfield values than those of the surrounding soft tissues. However, automatic segmentation of a bone could be a challenging task, due to several difficulties, including: a) non-uniformity of bone tissue, b) narrow inter-bone regions, and c) diffused and weak boundaries. Depending on the bony structure – pelvis and femur – it is seldom possible to find a a threshold that is less than the values of all the bones and greater than the values of the other tissues [CZW+13]. When talking about bones located at joints – such as foot bones –, the difficulties are due to the fact that bones at joints are too close to each other, which can make bone boundaries ambiguous. Left image of Figure 2.1 shows a CT image of the foot where it is possible to observe the fuzzy boundary of some foot bones. Right image shows how a region-based segmentation using a single seed may not identify completely a bone (the seed used are coloured in blue) or may join different bones (the seed used in this case are coloured in red).

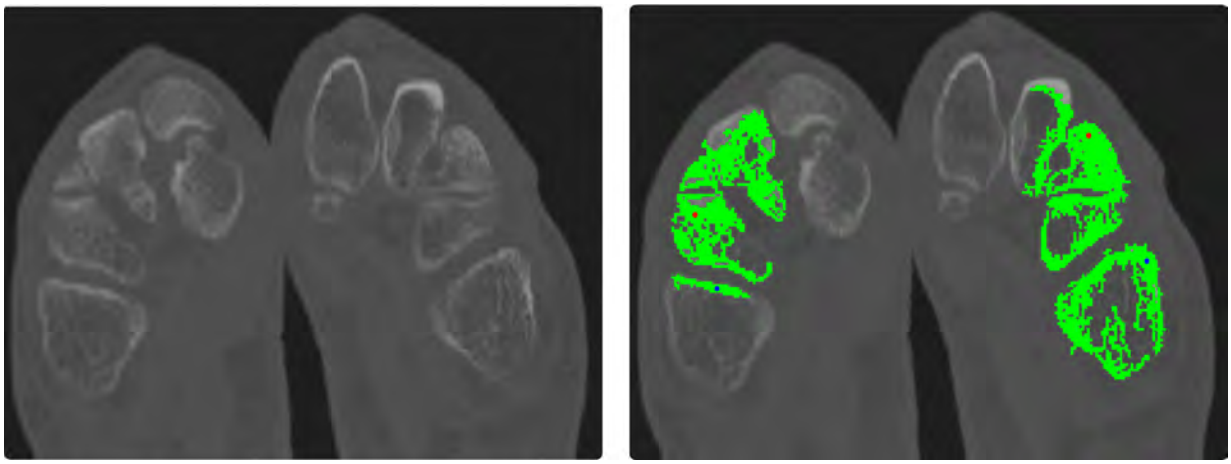


Figure 2.1: 2D CT view of complex bony structures. Left image shows a 2D view of several foot bones. Right image shows an example of four executions of a region-growing algorithm using as seed, the pixels marked in red and blue. Notice how, separated bones are segmented as an unique region and also how due to non-uniformity of bone tissue, the segmentation of a bone may require a lot of user intervention.

Segmentation algorithms can be classified according to different criteria [HD11, LUS+08]. Hu *et*

al. [HGM09] present and discuss some general segmentation techniques and categorize them into four groups: region-based (thresholding, region growing, clustering,..), boundary-based (deformable models), hybrid and model-based. Region-based and boundary-based techniques exploit within-region similarities and between-regions differences, respectively, whereas hybrid techniques use both region and boundary features. Model-based techniques deform a template that reflects the anatomy of a specific structure to segment a new scan.

Cheng *et al.* in [CZW+13] combine region-based techniques (thresholding and morphological filters) with a refinement scheme using gradient information to locate the accurate positions of the vertices of a triangulated bone surface obtained using the region-based technique. Their method was specially designed for the femoral head and the acetabulum bone from CT images.

Deformable models are curves or surfaces defined in an image domain that change their shape under the influence of forces. The forces are *internal*, from the curve or surface itself, and *external*, from the image data. Deformable models were introduced by Kass *et al.* [KWT88] and generalized to 3D by Terzopoulos *et al.* [TWK88]. Since then, different approaches have been published which propose new representations and deformation algorithms which allow the incorporation of changes to the topology of the initial shape, and offer improvements in efficiency and robustness [MT96, MDA01, HGM09]. Although deformable models can be customized to segment specific structures, in the presence of missing data, fuzzy boundaries or artifacts, they require the help of a medical expert to complete the segmentation. Lorigo *et al.* in [LFG+98] incorporate texture-based information into a geodesic active contours framework to segment bones in 2D MR volume images. Sebastian *et al.* [STCK03] combined active contours, region growing and region competition for the segmentation of carpal bones of the wrist in 3D CT images. The main limitation of the previous techniques is that they employ slice-by-slice strategies, needing user intervention for each slice, at least, for starting the evolution of the active contour framework.

Model-based (or atlas-based) methods aim to introduce medical knowledge into the segmentation algorithms. They usually consist of two steps. First, the model is approximately located in the 3D volume dataset; then the shape (and appearance) of the model is optimized to perform the segmentation. The two best known general approaches are *constrained deformable models*, which use a strong shape based on a simple example, and *point-based statistical models*, which store knowledge about the principal modes of variation of the template shape. Heimann *et al.* [HM09] presented a complete survey of 3D statistical shape models. Model-based algorithms are among the most robust methods when images are noisy or include artifacts. The major drawbacks are that statistical models require a large collection of training images and many shape parameters for complex structures. Although additional constraints result in a higher robustness, they also limit the accuracy of the final result. Liu *et al.* [LUS+08] state that no segmentation framework (not even model-based) may yield the level of precision, accuracy and efficiency that is required for the segmentation of the bones at a joint in MR and CT images. They propose a strategy for intra-patient segmentation based on a segmentation of a bone in one position performed by an operator using the live wire-based method. Subsequently, they use this model to search the same bone in other positions (images) by minimizing an energy function

that utilizes both boundary and region-based information. The minimization process calculates the rigid transformation that has to be applied to the live wire based segmentation. Although the results they obtain are quite good in terms of accuracy, we do not have to forget that the operator needs high amount of time to perform the initial segmentation.

Boonsuk [Boo09] present an approach to automatically segment the bone joint structures. The method employs a generic CAD model as anatomical knowledge to substitute that of medical experts. First of all, they proceed to segment the bone joint structure using a traditional segmentation approach based on region-growing. They state that in this initial segmentation the bones of a joint will be merged, so the proper bone boundaries will be lost. After that, they register both models (the generic CAD model and the reconstruction of the initial segmentation) and proceed to detect and repair the *welded regions* (those regions where the bones are joined due to ambiguous boundaries in the images). This last process is carried out into the cross-sections of the objects by analyzing geometrically the set of contours defined. Although they state their methodology can be applied to all the bone joints, they only give results about the joint of the pelvis with the femur.

Although a very extensive literature exists, there is still room for improvement in the reconstruction of bone joints. Most of the previous works in the subject of bone segmentation focused in one specific bone structure, being the femur head the most attacked problem. Only the work of Liu [LUS+08] deals with the segmentation of foot bones. This is a challenging problem due to the proximity of bones at the joint, partial volume effects, and other imaging modality-specific factors that confound boundary contrast. The new approaches should tend to minimize the user intervention and provide ever increasing fidelity of the results.

2.2 Example-guided bone joints segmentation

Our objective has been to develop a model-based technique that automatically segments a specific bone of a CT volume dataset taking as a guiding a high-quality mesh segmentation of the bone of interest. The main idea consists of the deformation of an *example mesh* until it matches the relevant volume features. The deformation process will not only be based on the patient's captured volume information, but also on the geometric shape of the original example mesh. With this approach, the resulting mesh adapts to the volume features in the areas which can be unambiguously segmented, while taking the shape of the example mesh in regions which lack of relevant volume information. The novelty of our strategy comes from the use of some geometrical properties of the *example* model in order to guide the algorithm.

An overview of the algorithm is presented in Figure 2.2. It starts from a volume data (V) to be segmented and an already existing segmentation (polygonal mesh) of the same organ in another dataset, usually from a different person. This mesh is called *reference mesh* (M). First of all, an approximate 3D registration between the reference mesh M and the volume V is performed (this is the only step requiring user intervention). After this coarse alignment of M with V, a pre-process step computes a suitable attracting field (called *Driving Distance Field* (DDF)) to the volume features. Next, the algorithm works

by minimizing an energy function adapting the shape of the polygonal mesh M to the volume features while maintaining the global shape of the reference mesh.

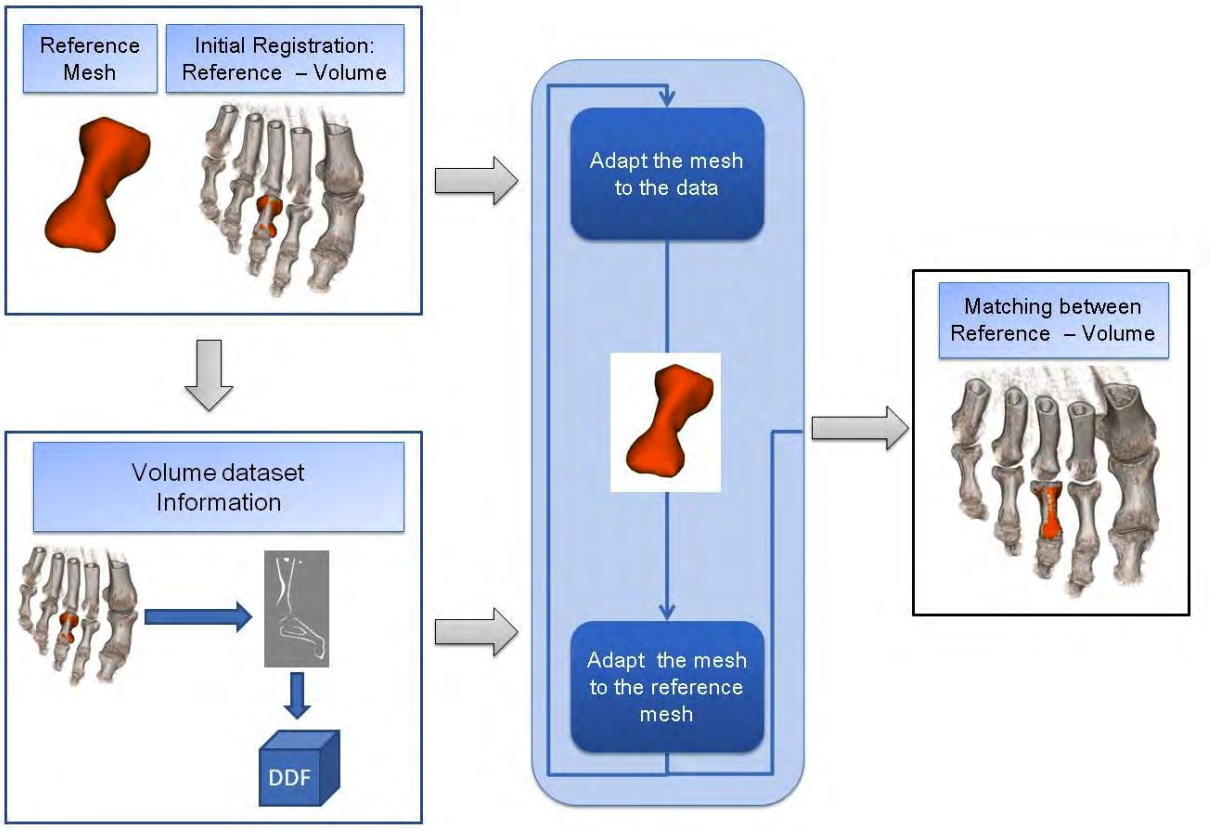


Figure 2.2: Scheme of the whole process: a reference mesh from a similar, segmented model is roughly aligned inside the volume to segment (upper left). Next, the volume data is processed to detect unambiguous boundaries of the structure to segment, and a driven distance field is computed (lower left). Then, an iterative process refines the rough alignment by using the volume boundaries when possible, and the reference model when boundary information is missing, yielding the segmentation result (right) without further user intervention.

The main components of the overall process are: the formulation of an energy minimization problem and the design of an adaptive minimization algorithm that tends to use the volume information in the areas that can be unambiguously segmented, while importing the example shape in the areas without relevant volume information. The local description of the shape of meshes is especially useful to capture and preserve details during deformations. To this end, we use the scheme proposed by Sorkine [SCOL+04], in which the local description of the shape is based on encoding each vertex with respect to the centroid of its topological neighbours through Laplacian coordinates (see Section 2.3.2).

2.2.1 Formal problem statement

Let V be a volume model. Let $M = (P, T)$ be a simple connected triangle mesh (the *reference mesh*) consisting of the triangles $T_{j\in\{1,\dots,U\}} \in T$ with vertices at points $P_{i\in\{1,\dots,N\}} \in P$. The mesh M represents the segmentation of a relevant portion (an anatomical structure, for instance) of a volume model $\hat{V}$ analogous to V. That is, all voxels in $\hat{V}$ that belong to the chosen anatomical structure are inside M, and all those that do not belong to that structure are outside of M.

We want to find a mesh M' contained in the bounding volume of V that is an adequate segmentation of the same structure given in the model $\hat{V}$. The mesh M' is a deformation of the input mesh M, and thus has the same topology; the vertices of both meshes are in a bijective correspondence $P_i \leftrightarrow P'_i$, and the triangle with vertices P_i, P_j, P_k belongs to M if and only if the triangle with vertices P'_i, P'_j, P'_k belongs to M'. In other words, we intend to set new positions of the vertices of the reference mesh, leaving its connectivity untouched, and so that, the new mesh represents a *plausible* segmentation of the same structure in the input volume model V.

By plausible segmentation we mean that the mesh M' follows the boundary of the chosen structure when that boundary is discernible in the input volume model V, and adopts a realistic shape (given by M) in the areas where it is not, perhaps because the structure in question is in contact with other parts of the model of very similar density, or because of other shortcomings in the acquisition process, like artifacts resulting from shadows of metallic implants.

The algorithm is not intended to solve the problem completely by itself, since it is very easy for a human to solve the problem approximately. We will thus assume that the given mesh M has been located close to the desired result, in a rough way. Let the user-defined, transformed version of M be $\mathscr{M} = (\{v_i\}, T_{\mathscr{M}})$. The algorithm does the second, more tedious step, of incrementally adjusting the vertices of $\mathscr{M}$ by minimizing an energy function until they are deemed a reasonable segmentation in V. The result of this process is the mesh M'. The result of the segmentation will be the voxels of V which are inside of M'.

The formulation of the energy function is composed of two complementary adjustment criteria.

The first criterion tries to measure the cumulative error of $\mathscr{M}$ with respect to the volume features in V, we call it $\mathscr{E}_\partial$. In order to carry out this calculation, we propose to compute the volume features (B_V) of V inside a region of interest (see Section 2.3.1), and store the distances from each voxel to B_V, which will be repeatedly required by the optimization algorithm (we dubbed this information *Driving Distance Field (DDF)*).

We define $\mathscr{E}_\partial$ as:

$$\mathscr{E}_\partial = \int_{\mathscr{M}} \text{distance}(x, B_V)^2 \, dS(x),$$

which, given the discrete nature of $\mathscr{M}$, can be computed as

$$\mathscr{E}_\partial = \sum_{v\in\mathscr{M}} \mathscr{E}_{v,\partial} = \sum_{v\in\mathscr{M}} \text{distance}(v, B_V)^2 A(v) \tag{2.1}$$

where $A(v)$ is the influence area of the vertex v.

The second measure tries to express the difference in shape between $\mathscr{M}$ and the example mesh M, we call it $\mathscr{E}_S$. To this end, we initially compute the Laplacian coordinates λ_{p_i} of each vertex in M (see Section 2.3.2), and then we define the shape error as:

$$\mathscr{E}_S = \sum_{v \in \mathscr{M}} \mathscr{E}_{v,S} = \sum_{v \in \mathscr{M}} \text{distance}\left(v, \sum_{v_i \in \text{ring}(v)} \lambda_{p_i} v_i\right)^2 \tag{2.2}$$

We should minimize each energy ($\mathscr{E}_\partial$, $\mathscr{E}_S$) where it is relevant. Notice that when $\mathscr{E}_\partial$ is zero (or very small), it should take precedence over $\mathscr{E}_S$. When $\mathscr{E}_\partial$ is large (implying no reliable boundary is available nearby), we should strive to minimize $\mathscr{E}_S$ (i.e. follow the example shape if the volume data does not shed any light on the boundary). The problem of computing M' can thus be seen as a minimization of the total energy given by the local geometric mean of these two energies:

$$\mathscr{E} = \sum_{v \in \mathscr{M}} \sqrt{\mathscr{E}_{v,\partial} \cdot \mathscr{E}_{v,S}} \tag{2.3}$$

Section 2.3.3 details the implementation of the adaptive minimization algorithm.

2.3 Implementation details

As exposed in Section 2.2, the algorithm requires two pre-processing steps before entering the minimization process. The first one (shown in the upper-left block of Figure 2.2) is a coarse registration between the reference mesh M and the volume dataset V. Given the anatomical knowledge of medical doctors, it is simple for them to identify 4 pairs of corresponding points between the reference mesh M and the given volume V; then M is transformed by the unique affinity that satisfies those four constraints. This is used as a starting approximation $\mathscr{M}$ of M', so the program may freely modify these four points just like any other in the ensuing optimization. This coarse alignment is the only step requiring user intervention. After that, the Laplacian coordinates of $\mathscr{M}$ are computed (see Section 2.3.2).

A second pre-processing step (lower left block in Figure 2.2) computes the volume features, B_V, of a region of interest around $\mathscr{M}$, and stores the distances (DDF) from each voxel to them (see Section 2.3.1).

Once the driving distance field DDF and the Laplacian coordinates have been computed, we proceed to minimize the energy $\mathscr{E}$ (Eq. 2.3). This is done in a greedy way, by applying two steps, which minimize the distance to B_V (volume features) and try to preserve the shape of the reference mesh, respectively (see Section 2.3.3).

2.3.1 Generation of the driving distance field

The driving distance field (DDF) is used to attract the mesh $\mathscr{M}$ to the unambiguous boundary features of the structure to segment. It is represented as a voxel volume dataset (same resolution than V) where for each voxel its chamfer distance to the nearest boundary voxel in B_V is stored.

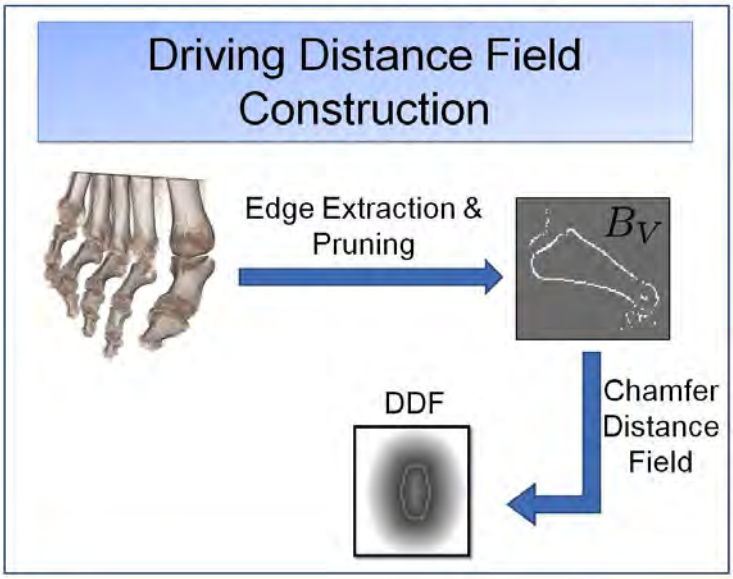


Figure 2.3: Steps for the driving distance field computation.

Figure 2.3 shows the overall process, which consists of three main steps:

1. Boundary detection of the structures captured in the volume dataset V.
2. Extraction of the unambiguous boundaries (B_V) of the structure to segment. This step consists of:
   a) Determining the range of density values of the volume dataset V, which defines the structure of interest.
   b) Erasing the boundaries which have a high likelihood of not belonging to the structure to segment.
3. Computing the volume distance field (DDF) to the remaining boundaries.

For the implementation of the first step, a 3D edge detection algorithm is applied to V in order to find which of its voxels correspond to boundaries between structures of distinct densities. There are many edge-detection algorithms that could be used; we have chosen the implementation by Monga *et al.* [MDMC90] because of its efficiency. The result is a new volume dataset of scalar values that indicate the strength of each edge (boundary). This volume dataset is binarized using a threshold (as seen in Figure 2.4(b)). In our experiments, we have considered a threshold of 0.5 after a normalization of the edge volume dataset.

However, some of the voxels may be erroneously classified as edges due to noise, while others match edges from other objects than the one we want to segment. So, the resulting binary volume model must be cleaned before proceeding. To do this, for each vertex of the roughly-aligned mesh $\mathscr{M}$,

we obtain the Hounsfield [Hou80] density value of the closest voxel from the volume model V. This will result in a range R of Hounsfield values corresponding to a range of density values of the structure of interest. Then, any voxel detected as an edge that has a Hounsfield value outside this range R will be erased (see Figure 2.4(d)). In this way, the edges belonging to objects of different tissue are eliminated after applying this process.

Nevertheless, after applying this initial cleaning, wrong edges may still remain. In order to get rid of them, edges with a high likelihood of not belonging to the object of interest are erased. First, edge voxels are classified into 2D connected components (Figure 2.4(e)). The connected components which are smaller than a certain size are erased, thus reducing the noise associated to the thresholding operation. Subsequently, the remaining components are checked to determine if the gradient vector volume at their location is coarsely aligned with the initial mesh. This is done by computing the gradient vector volume at each voxel of the connected component, and comparing it to the normal to the nearest vertex of $\mathscr{M}$. Any component with more than 50% of its voxels differing in angle more than $\frac{\pi}{3}$ with the corresponding normal of $\mathscr{M}$ are also discarded. These constant values were established empirically.

The result of the edge detection plus the cleaning process is the *volume feature set* B_V (see Figure 2.4(f)). Notice that the whole process is strict: in case of doubt we rather do without a possible boundary. We strive to keep only those voxels with the highest likelihood of being on the desired boundary as volume features. Other less clear portions of the boundary will be retrieved from the shape of the reference mesh (see Section 2.3.3).

From the volume features B_V a distance field is computed which stores for each voxel the distance to the nearest voxel classified during the previous steps as part of the boundary. We apply a Chamfer distance transform [Bai04] to do this efficiently. This distance field DDF is stored, since it will be repeatedly used in the optimization algorithm to get the mesh closer to the detected volume features (boundaries).



(a) Original volume data.



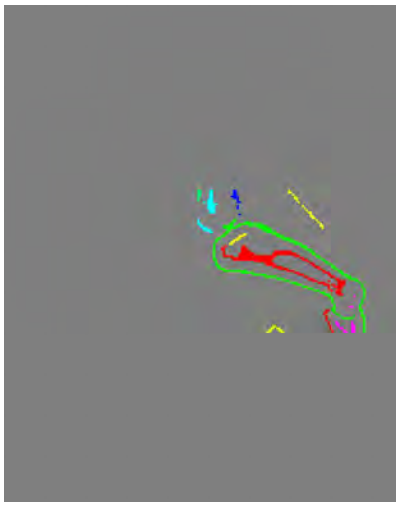
(b) Result of edge-detection filter.



(c) Same as 2.4(b), but showing the rougly-aligned example mesh.



(d) Edges within the bounding box of the mesh, and within the range R of density values.



(e) Edges are classified into connected components. Each component is coloured with the same colour.



(f) Result of the cleaning process.

Figure 2.4: Edges of the structure to be segmented are extracted using thresholding and connectivity for each slice. Here we show the steps of the process on a single slice of the volume model.

2.3.2 Generation of the Laplacian coordinates

The local description of the shape M is encoded using the Laplacian coordinates. This ingredient will be used to preserve, as much as possible, the original shape of the reference mesh (M) on the mesh ($\mathscr{M}$) we are deforming.

Given a mesh $\mathbb{M}=(\mathbb{K},\mathbb{V})$, where $\mathbb{K}$ describes the connectivity and $\mathbb{V}$ describes the geometric positions of its vertices, the Laplacian coordinates ($\mathscr{L}$) of a vertex $v_i \in \mathbb{V}$ are defined as:

$$\mathscr{L}(v_i) = v_i - \frac{1}{d_i}\sum_{j\in\mathscr{N}_i} v_j \tag{2.4}$$

where $\mathscr{N}_i$ is the *neighborhood ring* of the vertex v_i and is defined as the set of adjacent vertices to the vertex v_i, $\mathscr{N}_i = \{j|(v_i,v_j)\in\mathbb{K}\}$ and d_i is the *degree* of this vertex v_i, that is, the number of elements in $\mathscr{N}_i$.

Sorkine *et al.* [SCOL+04] use the Laplacian coordinates as a measure of the similarity between two meshes and also to transfer some geometric detail from one mesh to the other. When Laplacian coordinates are used for these purposes, it is convenient to express the Laplacian coordinates in a local reference system (orthogonal frame) relative to the vertex. Let's see how this local system is defined.

Given a vertex v_i in $\mathbb{M}$, its frame $\{\vec{e_1},\vec{e_2},\vec{e_3}\}^{\mathbb{M}}_{v_i}$ is defined in the following way:

- $\vec{e_1}$ is defined as the normal n_i of v_i.
- $\vec{e_2}$ is defined as the normalized projection u_{ij} of an specific edge e_{ij_i} emanating from i onto the tangent plane defined by n_i. We choose e_{ij_i} to be the edge with the largest ptojection onto the tangent plane at v_i.
- $\vec{e_3}$ is determined as $\vec{e_1}\times\vec{e_2}$.

Given a vertex v_i in $\mathbb{M}$ and its Laplacian coordinates $\mathscr{L}(v_i)$ expressed using the Equation 2.4, these Laplacian coordinates can be expressed in the corresponding local frame $\{n_i,u_{ij},n_i\times u_{ij}\}^{\mathbb{M}}_{v_i}$ as:

$$\mathscr{L}_{\mathbb{M}}(v_i) = (\alpha,\beta,\gamma)^T = \left[n_i,u_{ij},n_i\times u_{ij}\right]^T\cdot\mathscr{L}(v_i)$$

$$\mathscr{L}_{\mathbb{M}}(v_i) = (\alpha,\beta,\gamma)^T = [e_1,e_2,e_3]^T\cdot\mathscr{L}(v_i) \tag{2.5}$$

As we will see in Section 2.3.3, these definitions will be used in order to transfer the information of the reference mesh (M) onto the mesh $\mathscr{M}$. Before starting the optimization algorithm, for each vertex v_i in M, its Laplacian coordinates in its local frame, $\mathscr{L}_M(v_i)$ are computed and stored. Moreover, for each vertex v_i, the specific edge, e_{ij_i}, used in the definition of $\vec{e_2}$ is also stored.

2.3.3 Example-guided segmentation algorithm

The segmentation algorithm is based on an optimization process to minimize the energy $\mathscr{E}$ (see Equation 2.3) which measures the distance of the mesh $\mathscr{M}$ (which bounds the desired structure to segment) to both B_V (the detected boundaries features) and the reference mesh M. The optimization algorithm works in a greedy way, by applying two steps: Step_1 minimizes the distance of $\mathscr{M}$ to B_V, and Step_2 tries to preserve the shape of the reference mesh M. The minimization of the proposed energy $\mathscr{E}$ is used to prioritize Step_1 or Step_2 in different zones of the mesh. This adaptive scheme has proved to be a successful suboptimal optimization scheme for this problem, adapting to volume features and importing the example shape in the zones with poor volume information.

Step_1: Adapt $\mathscr{M}$ to the volume data (B_V)

The first step tries to reduce $\mathscr{E}_\partial$ (see Equation 2.2.1) by moving the mesh vertices towards the volume features B_V. Let DDF represent the driving distance field and v be a vertex of $\mathscr{M}$ inside the volume V. If the distance of v to B_v ($DDF(v)$) is larger than a certain tolerance (ε), Step_1 moves the vertex position in the direction opposite to the gradient (see Figure 2.5-left). In our implementation ε is the size of the main diagonal of a voxel. More precisely, the vertex v is translated by the vector

$$-min(DDF(v), \lambda \cdot \varepsilon) \cdot \nabla DDF(v) \tag{2.6}$$

where λ is a coefficient that scales the maximum amount of allowed movement.

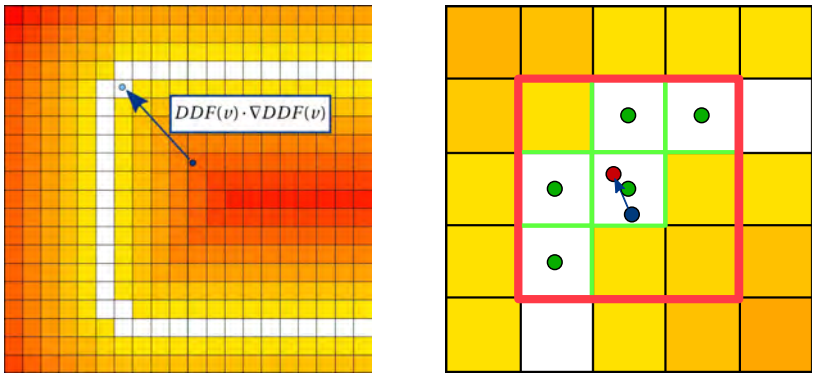


Figure 2.5: Left: Moving along the gradient brings the vertices closer to the detected edges. Right: Once a vertex (in blue) is close enough to the detected edges, the vertex is moved towards the centroid (in red) of the edge voxels (in green) inside a window (in light red) around the vertex.

However, distance values obtained from DDF are unstable for points closer than ε to the volume features. In this case we move the vertex towards the centroid of the positions of nearby voxels in B_V (see Figure 2.5-right).

At the end of Step_1, we apply a Laplacian tangential smoothing; that is, we compute a Laplacian smoothing [Tau95], and project the correction vector for each vertex v onto the tangent plane of the

mesh at that vertex. This smoothing process improves the triangle shape and also the stability of the shape-preserving part of the algorithm.

Step$_2$: Adapt $\mathscr{M}$ to the example mesh M

In order to preserve the shape of the reference mesh during the deformation caused by Step$_1$, a Laplacian reshaping [SCOL+04] is applied in Step$_2$ to decrease $\mathscr{E}_S$ (see Equation 2.2). This process consists of moving each vertex of the deformed mesh $\mathscr{M}$ to a new position which reduces the difference in shape between $\mathscr{M}$ and the example mesh M.

For each vertex v'_i of the deformed mesh $\mathscr{M}$, the corresponding vertex v_i of the mesh M is obtained. As was stated in Section 2.3.2, being v_i a vertex of the mesh M, its corresponding Laplacian coordinates expressed in its local frame $\{n_i, u_{ij}, n_i \times u_{ij}\}^M_{v_i}$ are $\mathscr{L}_M(v_i)$ (see Equation 2.5). Following the same process, for a vertex v'_i of the deformed mesh $\mathscr{M}$, its local frame is $\{n'_i, u'_{ij}, n'_i \times u'_{ij}\}^{\mathscr{M}}_{v'_i}$. For this last calculation, we only need that u'_{ij} be defined using the same corresponding edge as for u_{ij} (as explained in Section 2.3.2 for each vertex v_i, we store which was the edge used to calculate $\mathscr{L}_M(v_i)$).

As stated, being $v'_i \in \mathscr{M}$ and $v_i \in M$ its corresponding vertex, the position of v'_i is updated in Step$_2$ as follows:

$$v'_i \longleftarrow \frac{1}{d'_i} \sum_{j \in \mathscr{N}'_i} v'_j + [e'_1, e'_2, e'_3] \cdot \mathscr{L}_M(v_i) \tag{2.7}$$

In this way, the vertex v'_i is updated taking into account the local shape defined in v_i in M. Notice that the position of vertex v'_i, before being updated by Step$_2$, is in charge of defining its corresponding local frame and also its neighborhood participate in it, so the work that Step$_1$ could do, is not erased by the new update performed by Step$_2$.

As it will be detailed in the optimization algorithm explanation (see below), this updated is only done if it decreases the energy $\mathscr{E}$ (equation (2.3)), so the work that Step$_1$ could do to move the mesh towards the volume features is not erased by Step$_2$ in the case that there is not a decreasing of the total energy.

Optimization algorithm

The optimization process is shown in Algorithm 1. It works in two phases. In the first one (lines 2–7), the algorithm simply iterates Step$_1$ and Step$_2$ to obtain an intermediate mesh $\mathscr{M}$ which is closer to the target. A number of 10 iterations ($IterPhase_1 = 10$) has proven sufficient in all our tested examples. The objective of this phase is to get the mesh $\mathscr{M}$ closer to the target without taking into account the minimization of the energy – we do not have to forget that initially $\mathscr{E}_S$ is equal to zero since the shape of $\mathscr{M}$ is equal to M.

The second phase (lines 9–25) is adaptive, using the proposed energy to prioritize Step$_1$ or Step$_2$ in different zones of the mesh in a greedy and dynamic way. At each iteration, the energy $\mathscr{E}$ (equation (2.3)) of the deforming mesh $\mathscr{M}$ is first computed. Then, Step$_1$ is computed for each vertex v, but, at this step, vertex positions are not assigned. A second mesh traversal queries, for each vertex

v, whether the intended movement proposed by Step_1 would result in a decrease of $\mathscr{E}$. Only if it is true, Step_1 is applied to that vertex v. The second part of the iteration loop is identical, but considering Step_2 instead of Step_1. Note that, in regions close to volume features in B_V, Step_1 will be active. In other regions with no volume information, Step_2 will be automatically active. Notice also that the evaluation of the function DecreaseEnergy involves only a local computation (when we move a vertex, most terms in the evaluation of Equation (2.3) stay fixed, so only the terms involving v and its 1-ring need to be computed). The algorithm finishes when the minimization of the energy is less than certain tolerance – a value of 5 worked well for all the tested models. The program stores volume models as three-dimensional arrays of values, and the meshes using a corner-table as in [RSS01], which provides the necessary topological information for computing the Laplacian coordinates efficiently.

Algorithm 1 Optimization algorithm.

```
▷ Phase 1: Start with fixed number of alternate steps
for iter = 1 to IterPhase_1 do
    M = ComputeStep_1(M, DDF)                          ▷ See Eq. (2.6)
    M = ComputeStep_2(M, M)                            ▷ See Eq. (2.7)
end for
                                                       ▷ Phase 2: Adaptive refinement
E = MeshEnergy(M, M, DDF)                              ▷ See Eq. (2.3)
repeat
    E_prev = E
    M_aux = ComputeStep_1(M, DDF)                      ▷ See Eq. (2.6)
    for each v ∈ M do
        if DecreaseEnergy(v, M_aux, M, DDF) then
            M.v = M_aux.v
        end if
    end for
    M_aux = ComputeStep_2(M, M)                        ▷ See Eq. (2.7)
    for each v ∈ M do
        if DecreaseEnergy(v, M_aux, M, DDF) then
            M.v = M_aux.v
        end if
    end for
    E = MeshEnergy(M, M, DDF)                          ▷ See Eq. (2.3)
until E_prev - E ≤ ε
```

2.4 Results

We have used the proposed algorithm to segment some bones of the the foot. Figure 2.1 shows the extent of imbrication of these bones, the specific difficulty in segmentation that our algorithm is designed to overcome. Concretely, we have segmented the *Phalanx*, the *Calcaneus* and the 1 and 5 *Metatarsal* bones. We have used data from the Visible Human Project [Nat] to obtain the reference meshes of

the bones to segment. Table 2.1 shows the bones chosen for these tests and the characteristics of the triangle meshes obtained from their segmentation as provided by the Visible Human Project. The characteristics of the two CT volume datasets used are summarized in Table 2.2.

Structure	**Example Mesh**	
	Vertices	**Faces**
Phalanx (P)	1254	2504
5–*Metatarsal* (5–M)	24678	49352
1–*Metatarsal* (1–M)	9162	18216
Calcaneus (C)	90586	181108

Table 2.1: Information of the chosen bones to segment, and characteristics of the reference meshes.

Volume Model			**Structure**
Name	**Voxel Size**	**Resolution**	
V_1	1.0mm^3	$256 \times 256 \times 256$	*Phalanx* (P)
V_2	$0.601\text{mm}^2 \times 0.625\text{mm}$	$256 \times 512 \times 272$	5–*Metatarsal* (5–M) 1–*Metatarsal* (1–M) *Calcaneus* (C)

Table 2.2: Information of the volume models to segment and the structure which will be segmented in each of them.

We designed a first test to have a fair measure of the accuracy of the algorithm. We carefully segmented the 1-*Metatarsal* bone by hand in the patient's foot volume dataset V_2 (see Table 2.2). From the collection of hand-picked voxels, we extracted a smooth mesh using the *pressing* algorithm [CWA+08]. We then defined the error at a point as the distance to this hand-built mesh. This measure of error is displayed for each vertex of each mesh in Figure 2.6 from three different vantage points. The top row of Figure 2.6 shows this error applied to the vertices of the roughly aligned example mesh before applying our optimization algorithm, and the bottom row shows the same measure on the vertices of the resulting mesh after running it. With this measure, the vertices at the starting position of the example mesh were at a distance of 7.33mm or less, with a mean distance of 1.75mm. The result mesh, in contrast, was at a maximum distance of 3.07mm, with a mean distance of 0.39mm. Notice that, as listed on Table 2.2, the sides of the voxels in the volume model measured nearly 0.6mm, so we deem these results fairly good; while there are a few outlier vertices, the vast majority of the mesh is closely wrapped to the target structure. Figure 2.7 shows the resulting segmentation in the context of the whole dataset from two different viewpoints, to display the extent of imbrication of these bones and the specific difficulty in segmentation that our algorithm is designed to overcome.

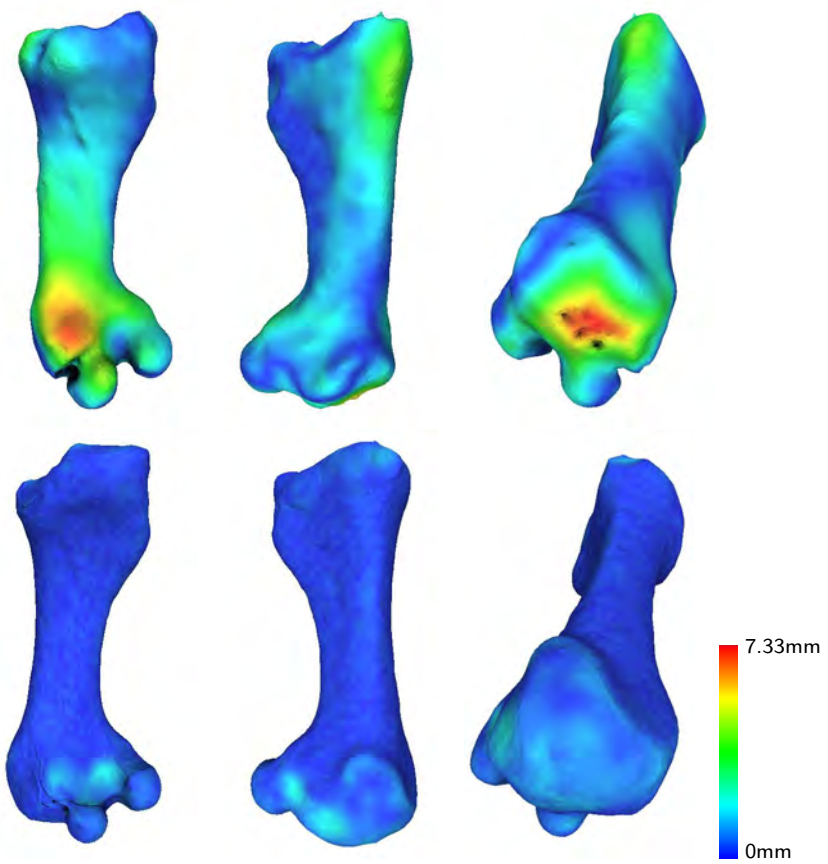


Figure 2.6: Error of the algorithm measured against a hand-made segmentation of the *Metatarsal* bone. The top row displays the error of the initial coarse alignment, while the bottom row shows the error for the mesh obtained by our approach.



Figure 2.7: The *Metatarsal* bone, shown in the context of the rest of the patient's foot, from a top (left), a lateral (middle), and a below (right) views. For each view, left image shows the initial mesh after the initial coarse registration and right image shows the resulting mesh.

In order to perform a more exhaustive testing without the need to segment each case by hand, we programmed an approximate measure of error by assigning to each point a distance equal to the distance to the closest bone voxel with an immediate neighbor which is not bone. All remaining examples in this section display this measure of error.

A first example using this measure is shown in Figure 2.8, which shows the segmentation of the *Calcaneus* bone in the patient's heel (volume dataset V_2 in Table 2.2). The left column shows three different views of the rough alignment given by the operator. The right column shows the result after 26 iterations, seen from the same viewpoints. The resulting average error is of the order of the scale of the voxels.

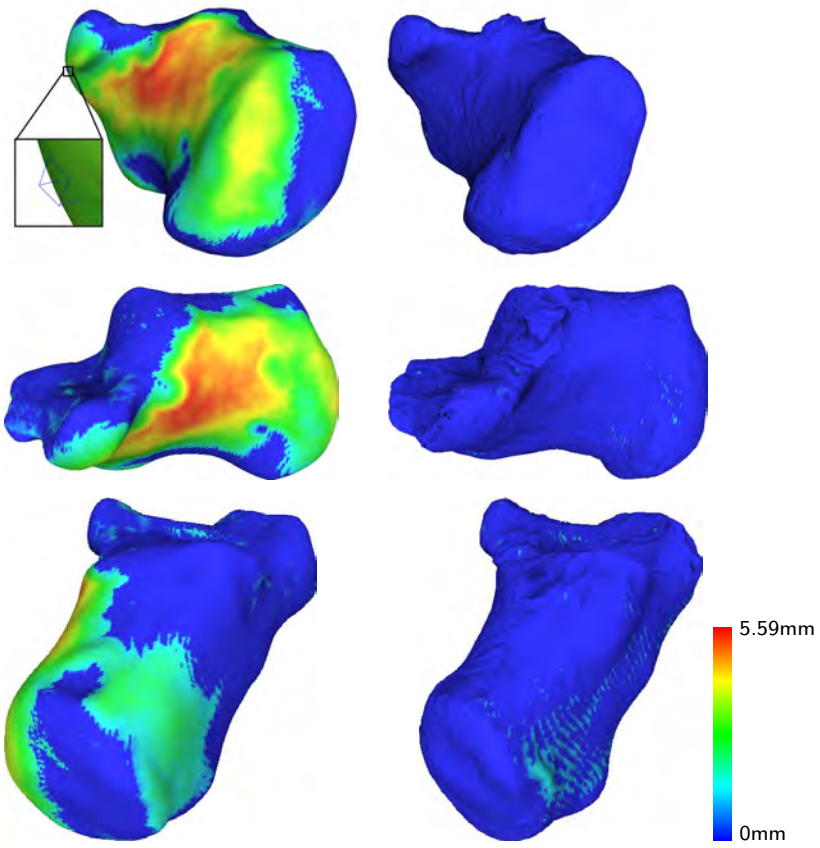


Figure 2.8: Three different views of the *Calcaneus* after the first, rough, alignment with the example (left column), and after running the algorithm (right column). The colors show distance from the mesh to the actual boundary in the scale presented. Notice the blow-up of the top-left figure, showing the scale of the voxels.

A second example using the same volume dataset is shown in Figure 2.9. Here, we show two different views of the *Phalanx* bone, displaying again the starting position in the left column, and on the right, the same views after 40 iterations, with the same color-coding of the errors. Figure 2.10 shows the resulting segmentation in the context of the whole dataset.

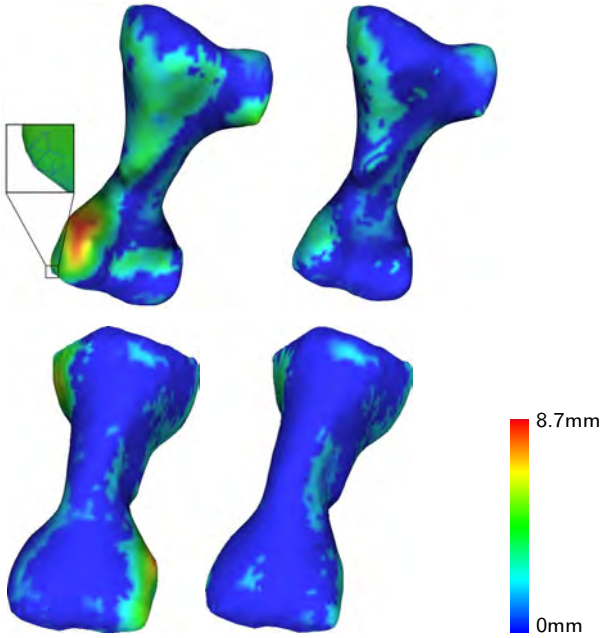


Figure 2.9: Two different views of a *Phalanx* bone after the first, rough aligment with the example mesh (left column), and after running the algorithm (right column). The colors show distance from the mesh to the actual boundary in the scale presented. Notice the blow-up of the top-left figure, showing the scale of the voxels. The resulting average error is 1.09mm, which is of the order of the scale of the voxels.

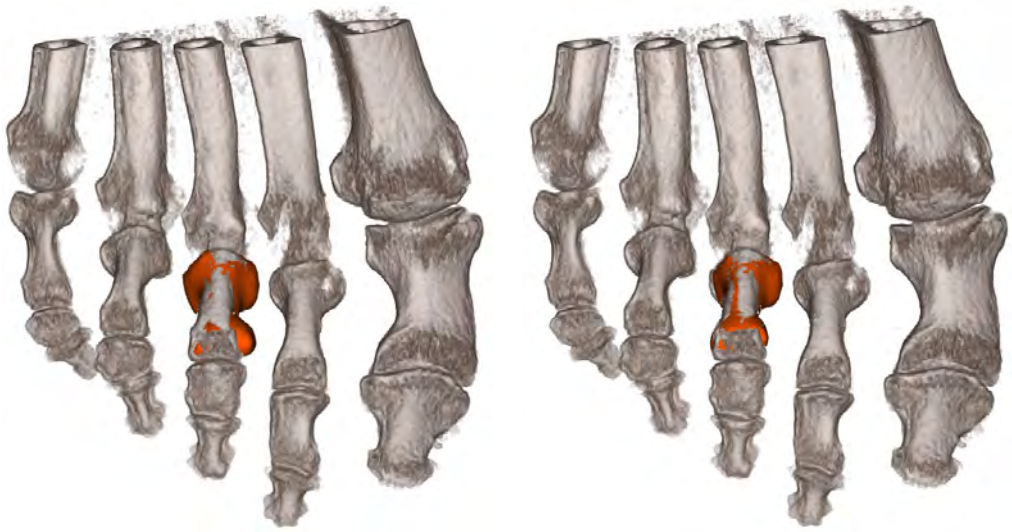


Figure 2.10: Resulting segmentation of the *Phalanx* bone shown in the context of the whole dataset. Left image shows the initial mesh after the initial coarse registration and right image shows the resulting mesh.

We finally display in Figures 2.11 and 2.12 the same kind of rendition of the results as in Figures 2.8

through 2.7, in this case for the 5-*Metatarsal* bone, using 26 iterations.

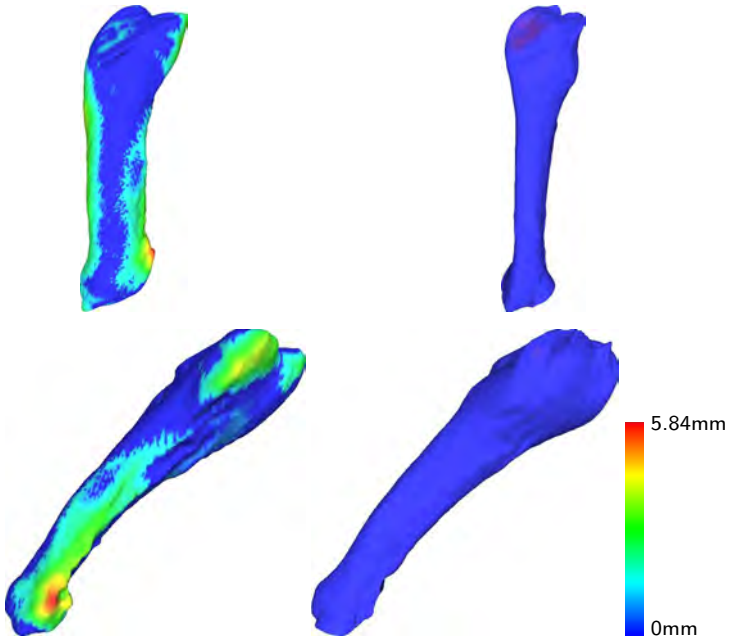


Figure 2.11: Two different views of a *5-Metatarsal* bone after the first, rough alignment with the example mesh (left column), and after running the algorithm (right column). The colors show distance from the mesh to the actual boundary in the scale presented. Notice the blow-up of the top-left figure, showing the scale of the voxels. The resulting average error is 1.77mm, which is of the order of the scale of the voxels.

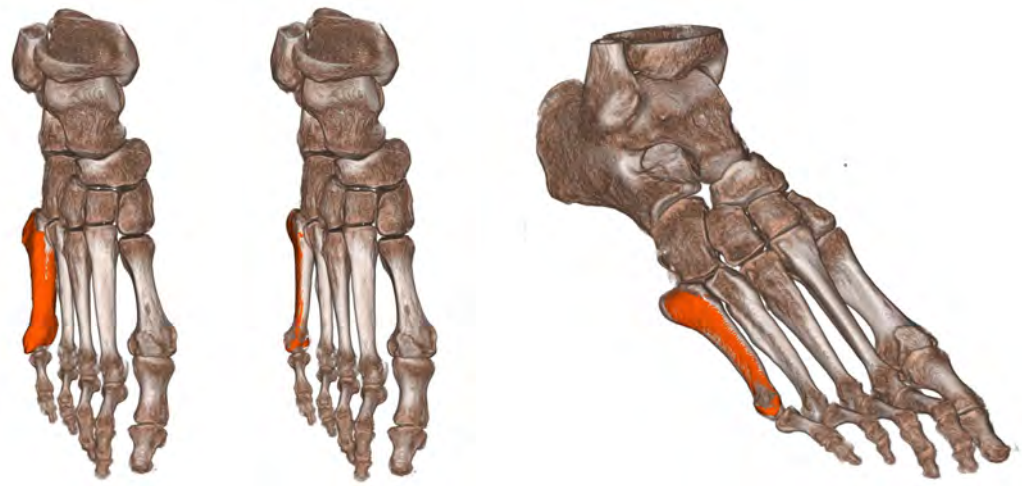


Figure 2.12: The 5-*Metatarsal* bone, shown in the context of the rest of the patient's foot. Left image shows the initial mesh after the initial coarse registration. Middle and right images show the resulting mesh.

Tables 2.3 and 2.4 summarize the performance and the accuracy of the algorithm. Table 2.3 shows

the time spent on the different steps of our technique for each of the examples. Table 2.4 shows the accuracy results of the adaptive algorithm for each of the examples. Notice that the example of the *Calcaneus* bone uses a lot more time, despite needing fewer iterations of the main loop. This is due to the larger complexity of the example mesh (see Table 2.2). Both the magnitude of errors and the timings of the algorithm compare favorably to those reported in the literature for similar applications (see for instance [LUS+08]), and our proposal requires far less operator intervention. We consider our work could be considered complementary to theirs to automate their initial manual segmentation of the structure of interest in one of the different volume models.

Bone	Laplacian Coord.	Edge Detector	Chamfer Distance	Iterations	Optimization Algorithm
P	0.049	6.48	2.06	40	14.51
1–M	0.068	20.9	4.42	30	18.48
5–M	0.207	69.12	6.14	26	47.83
C	0.588	375.16	10.88	26	163.24

Table 2.3: Execution times for the pre-processing sub-steps and for the optimization algorithm in seconds. We consider that the performance achieved is good enough for this kind of tasks.

Bone	Initial Max Distance	Final Max Distance	Mean Distance	Std. Dev	Initial Energy	Final Energy
P	8.7	4.77	1.90	0.99	16749.6	174.44
1–M	7.33	3.07	1.73	1.40	38216.7	29.75
5–M	5.84	1.58	1.77	1.21	114682	205.58
C	5.59	1.51	1.09	0.86	11122.5	38.83

Table 2.4: Quantitative results for the algorithm. The initial and final maximum errors are given in millimeters, as the mean and the standard deviation of the error are also given for all the vertices in the final registered mesh. Notice, that the accuracy of the algorithm is closer to the voxel size.

2.5 Conclusions

We have proposed a new algorithm for the segmentation of bone structures located at joints [CMB+12]. The described algorithm allows users to segment challenging cases, as foot bones, that are too close to each other to be distinguishable in CT datasets. The algorithm uses an example mesh of the same organ, usually from a different person, to drive the segmentation process. The final result is based both on the patient's captured volume information and on the geometric shape of the example mesh. The shape optimization is based on energy minimization and it works by deforming the initial example mesh while following the features of the volume data in a local and adaptive way. The results show a good convergence rate and reasonable clinically admissible residual errors.

We can not formally guarantee that the result mesh fulfills any property of quality. For instance, the result mesh could have auto-intersections, although it is highly unlikely to find auto-intersections, due to the application of the Laplacian smoothing after applying the $Step_1$ and, above all, the introduction of $Step_2$ which has the objective of preserving the shape of the reference mesh. Experimentally, the tested models do not suffer from this problem. Additionally, the result mesh is just a means to obtain the collection of voxels that belong to the structure of interest.

One of the main limitations of this technique is that the final mesh depends greatly in the effectiveness of the process of cleaning the volume features not belonging to the structure of interest, because the generation of the driving distance field comes from it. Therefore, it might be interesting to look for solutions in the cases when our heuristics can not achieve a good compromise between the preserved edges and those erased. According to the performed experiments, it is better to lose some correct edges than having spurious ones.

However, as it is common in this kind of algorithms, the convergence of the method can not be guaranteed, also the algorithm can fall into a local minimum due to the detection of edges not belonging to the real boundary of the structure of interest. As it has been mentioned before, a future line of work could be the investigation of new solutions for dealing with this kind of problems.

CHAPTER 3

AUTOMATIC GOOD VIEWPOINT SELECTION

Good viewpoint selection techniques seek to automatically obtain a viewing point or direction that allows the user to inspect the scene under certain conditions. For instance, it might be interesting to determine a view that is representative of an object or scene, or a viewpoint that provides better information on certain parts of the object, or simply a good initial point for an automatically guided exploration inside a complex indoor scene. Viewpoint selection has been an active research topic in many fields, such as object recognition, object modeling or cinematography, and also in *Computer Graphics*.

The visual exploration of volume models can be both a tedious and a time-consuming task. Consequently, assisting users in this task is a must. Saving time for physicians is always welcome. Therefore, the automatic determination of optimal viewpoints for starting the exploration may avoid the non-intuitive trial-and-error viewpoint search process. In addition, the task of looking for a specific volume dataset through a large collection of them can be highly improved by providing a representative view of each model. A well-known approach is the calculation of *Good Views* of a volume model, which gives users a quick first impression of it and helps them to understand the model and find representative views in an efficient way. In non-interactive situations, a minimal set of representative views can be used to improve the task of data understanding.

This chapter proposes a method, based on entropy measures, that improves the automation of the process of generating good viewpoints in order to offer an interesting set of views before starting the inspection task. The proposed technique allows users to obtain a quick previsualization of a volumetric dataset–not necessarily segmented, in a short time, and with the use of an automatic algorithm.

The remainder of the chapter is organized as follows. Section 3.1 summarizes previous works on best viewpoint selection techniques. Section 3.2 outlines the proposed approach. Section 3.3 introduces the used entropy measure and describes the developed algorithm for best view determination.

Section 3.4 introduces a similarity measure tailored to analyze the similarity between two views and describes the algorithm for the selection of a representative view set. Section 3.5 details the construction of the exploration path. Finally, conclusions are presented in Section 3.6.

3.1 Related work

As mentioned before, the computation of good viewpoints has been an active research topic in many fields, such as object recognition, object modeling or cinematography, and also in *Computer Graphics*.

Concerning *Computer Graphics*, initial research focused on surface-based models. In this domain, Vázquez *et al.* [VFSH01b] defined the *Viewpoint entropy*. It is based on Shannon Entropy [Sha48] and evaluates the amount of geometric information that arrives to a point taking into account the projected area of each of the scene's faces. This measure searches for a well-balanced distribution of the visible faces. The entropy value is maximized when all the faces project to an equal area on the screen. Unfortunately, this approach is not directly amenable to *Direct Volume Rendering*, because the rendering primitives are not a set of opaque polygons, and the usual rendering techniques produce images where, for each pixel, more than a single (iso)value may contribute to its final color. Thus, specific methods focused on viewpoint selection have been developed in the area of *Volume Graphics*. Most of them base their formulation on *Information Theory*.

Takahashi *et al.* [TFTN05] proposed a measure, called *surface area entropy*, based on the *Viewpoint entropy* defined for surface-based models in [VFSH01a]. In order to adapt the original formulation to volume datasets, they decompose the entire volume into a set of feature components (*interval volumes*), and find a globally optimal viewpoint using the local optimal viewpoints obtained for the geometric properties of each feature component using the surface-based entropy proposed in [VFSH01a]. The viewpoint entropy of the entire volume is defined as an average of the local viewpoint entropies. Using an specific *interval volume decomposer*, they provide a systematic decomposition of the entire volume reflecting its involved topological structures. Their feature analysis assumes that the given volume dataset contains some characteristic global structure. This is the reason why their method provides more reasonable results for simulated datasets rather than acquired volume datasets. Additionally, the opacity transfer function is used to assign different weights to the decomposed components, providing users the possibility to emphasize different features. In order to obtain the best and the worst viewpoint, they discretize the view space as an uniform triangular tessellation of the bounding sphere (162 samples) of the volume dataset and place viewpoints at the triangle centroids.

Bordoloi and Shen [BS05] proposed a new measure to identify a minimal set of representative views. This measure, called *voxel entropy*, is based on the opacity of a voxel as a indicator of its noteworthiness. To calculate the goodness of a view, they need to know the voxel visibilities and noteworthiness factors. Voxel visibilities can be queried through any standard volume rendering technique and the noteworthiness factor depends only on the transfer function involved. In order to find a representative views set, they proceed in the same way as Takahashi by a discretization of the view sphere into 128 viewpoint positions. For each viewpoint, they calculate the goodness of the view using their view

entropy measure. Starting from a number of representative views specified by the user, they partition the view space using a similarity function based on Jensen-Shannon divergence metric [Lin91] as partitioning criteria, to find the similarity between two views. Once the partition has been calculated, they select as a representative view of each cluster the view with the highest entropy value.

Ji and Shen [JS06] proposed an image-based metric for measuring the quality of a view based on the opacity, color and curvature images generated by a volume rendering package. They select the optimal viewpoint by calculating the image entropy for 256 sample views evenly distributed on the viewing sphere. Using this information and a dynamic programming approach, they construct an animation for a time-varying volume dataset under the constraints of smooth view change and near-constant speed.

Viola *et al.* [VFSG06] have also developed a preprocess method which takes into account not only a viewpoint quality metric, but also information on focus of attention. They introduce the mutual information between a set of viewpoints and a set of objects to calculate the representativeness of a viewpoint. This measure is used to automatically focus on features within a volumetric dataset. After the introduction of a feature-focus by the user, their system automatically determines the most expressive view on the feature of interest.

Chan *et al.* [CQWZ06] focus on viewpoint selection for angiographic volume data. Since their method is domain-specific, they can take profit of all the usable knowledge the physicians offer. To evaluate the quality of a view, they use different view descriptors that rely on the visibility, the self-occlusion and the coverage of the objects of interest captured from a certain viewpoint. Instead of sampling the view space and evaluating the views one by one, they construct a solution space to estimate the quality of the views by searching in it using the gradient descent-based algorithm.

Mühler *et al.* [MNTP07] focus on intervention planning. Their scenes consist of many pre-segmented anatomical objects of different importances. They preprocess a set of viewpoints placed at 4096 positions on a bounding sphere by recursively subdividing a double tetrahedron. At each point, they calculate a set of parameter maps that indicate the influence of the current quality parameter settings on the viewpoint. Some parameters are object-dependent (like the size of the visible surface or the portion of each object's surface which is occluded by other objects), while others are situation-dependent (like the distance to the current camera position or the stability of a viewpoint with respect the current camera position). These parameter maps, that are application dependent, are weighted and combined. The best viewpoint for the object of interest is given by the maximum of the weighted sum of all the parameters.

Tao *et al.* [TLB+09] present an algorithm that uses two structure-aware view descriptors to evaluate the viewpoint quality of global structures and local details. The evaluation of global structures is carried out by measuring the distribution of the relative angle between the view direction and the gradient direction around the viewing sphere. In order to extract the local details, they construct a new volume dataset by applying a bilateral filter, called *Shape volume*. Variances between the Shape volume and the original volume stand for the local detail information. Their detail descriptor maps local structure

variance to the emitted intensity by a voxel. The location of the optimal viewpoint implies the computation of the amount of information of the intensity image of both descriptors using the Shannon entropy. They integrate both descriptors into a viewpoint selection framework, where the user has the flexibility to emphasize global structures or local details depending on the specific situations.

Zheng *et al.* [ZAM11] use feature-clustering in a high-dimensional space as the criterion for viewpoint selection. They use the gradient variation as a metric to identify interesting local features. These are then clustered using the k-means paradigm in order to detect important salient features. This information is mapped into a 2D map and serves to guide the user in the visual exploration task.

Grau *et al.* [GPE+13] proposed an automatic method to obtain a representative complementary camera for a volume dataset. Given a camera of reference, they look for a complementary camera (view) that maximizes the complementary information on the structures visualized from the reference camera. The proposed measure combines the quantity of information that each view provides for each structure (using an entropy-based method as in [TFTN05]) and the similarity between the two views. The similarity between two views is based on the visual shape in terms of the silhouette and region properties. They complement the similarity measure with a correlation of the information captured from each feature between the two images. They assume that the volume dataset is semantically tagged in order to be able to apply the similarity definition. Given the reference camera, they use 768 sampled positions in order to find the position of the complementary camera.

Bramon *et al.* in [BRB+13] presented a new metric to evaluate the quality of a view. Their approach is based on the definition of an observation channel that relates the intensity values of the volume dataset with the observed pixel colors obtained by the rendering process. Once this relationship has been established, an analysis of the mutual information between the input and the output is used to evaluate the quality of different viewpoints.

An analysis of the published research revealed that, at the moment of our development [VMN08], there was still room for improvement. First, the view quality evaluation measures developed so far had not taken into account the final color the user is perceiving. Our approach focuses on the analysis of the information contained in the final view, since this information is what actually reaches the user. Also, a complex data preprocessing is necessary in some of the methods ([TLB+09, MNTP07]), while we would like to explore automatic (or almost automatic) algorithms that do not required any segmentation preprocess. Moreover, the published research needs non-negligible time for computing a good viewpoint since they perform an exhaustive computation around the viewing sphere so it is necessary to improve the amount of time in performing this task.

Furthermore, apart from developing a quality measure for single, best view determination, we also address the problem of the selection of a set of representative views and its efficient computation. From this initial set, we also provide an initial exploration path.

3.2 Process overview

We have focused our research on studying efficient algorithms to compute *Good Views* for non-segmented volume datasets. We propose a new algorithm for the automatic selection of representative views and the automatic generation of exploration paths for volume models based on entropy measures [Váz03]. The proposal works upon a model (a raw volume dataset) classified through the definition of a transfer function and, optionally, the specification of a region of interest. Starting from this minimal information, it generates both a set of representative views of the model and an exploration path that allows users to choose the most informative view. The proposed method only uses the images generated through a DVR algorithm. As it does not need any kind of preprocessing on the volume dataset, its use is feasible in the daily clinical practice.

We have developed a system (see Figure 3.1) that performs an integral analysis of a volume dataset providing solutions for the following goals:

1. Best view determination. An adaptive algorithm evaluates a set of views and determines the best one. In order to obtain a noteworthy view, a view quality measure has to be defined. We use an image-based entropy measure to address this issue (see Section 3.3).

2. Representative views selection. We tackle this with the use of the Normalized Compression Distance, a complexity-based metric (see Section 3.4). Once we are able to evaluate a single view, this method selects a representative set that maximizes information and reduces redundancy.

3. Exploration path construction. For the inspection of complex models, continuous and soft paths may be used to gather information around or inside the model. We have developed an algorithm that uses the previously defined metrics to construct a path around a model (see Section 3.5).

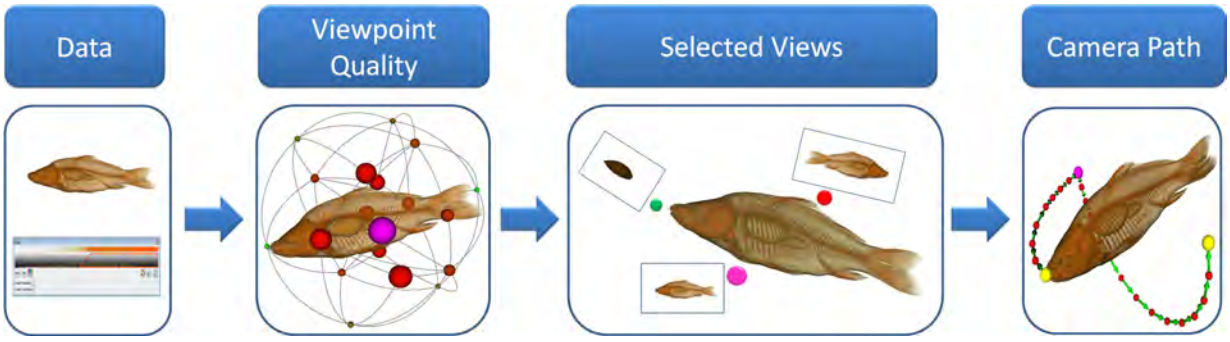


Figure 3.1: Workflow of the proposed system. Once the data is loaded and a transfer function is defined, an adaptive algorithm evaluates a set of views and determines the best view. With the analyzed images, a set of representative views is selected. This set is used for a final exploration path definition.

One of the main advantages of the proposed system is that it does not require any preprocess of the volume dataset. The required computations give the results in a time comparable to that of loading a

model. Thus, the user can immediately begin the inspection starting from a good view of the model. Moreover, the quality measure of a view can be changed without affecting the overall system, as the rest of the steps work independently of this measure.

3.3 Best view determination

The goodness of a view can be analyzed using different metrics (see Section 3.1). In [VS03, Váz07], Vázquez *et al.* introduced a measure based on Multiscale Entropy to analyze the amount of information present in an image that seeks to maximize the information revealed to the user given a certain illumination of the scene. We decided to adopt this measure to volume models, since it is based only on the analysis of images. Other proposed techniques do not measure the quality of a view directly on the color images generated by the rendering algorithm, and thus, different structures might be treated separately although they could produce a uniform colored region on screen. In contrast to these techniques, the measure that we use works on the generated image, with the objective of measuring only the information that will be effectively seen by the user.

3.3.1 Preliminaries

Most of the approaches developed in the area of *Computer Graphics* and *Volume Graphics* compute the amount of information contained in an image with the use of a measure based on the Shannon entropy from the field of Information Theory.

The information provided by an image can be measured by the Shannon entropy as:

$$H(X) = -\sum_{i=1}^{N} p_i \log p_i, \tag{3.1}$$

where $X = \{X_1, X_2, \cdots, X_M\}$ is an image containing M pixels (integer values), N is the number of different values that pixels can take, and p_i are the values obtained from the histogram of X, that is, the probabilities of each histogram entry:

$$p_i = \frac{\#\{X_j = i\}}{N} \tag{3.2}$$

The logarithms are taken in base 2 and $0 \log 0 = 0$ for continuity reasons. As $-\log p_i$ represents the *information* associated with the result X_i, the entropy gives the average information or the *uncertainty* of a random variable. Observe that the entropy $H(X)$ will be 0 if all the pixels have the same value and maximum when all pixels have a different value.

Unfortunately, this entropy definition leads to measures that are insensitive to pixel correlation or dependent on background. To overcome those problems, Starck *et al.* introduced the use of a multiresolution scheme that has been developed under the name of *Multiscale Entropy*. This metric measures the information of an image as the sum of the information at different resolution levels. The core idea

is to use the wavelet transform to generate a set of resolutions of the image and measure the entropy of the wavelet coefficients in all the resolutions using the Shannon entropy (see Equation 3.1).

Vázquez [Váz07] adapted the Multiscale Entropy measure (MSE) by Starck [SMPA98] for its use in RGB images, and used it to determine the amount of information in a rendered view. The amount of information contained in a RGB image is measured by analyzing the Shannon entropy of the wavelet coefficients of each color channel at each level:

$$MSE = H_W(X) = -\sum_{l=1}^{L}\sum_{k=0}^{N_l} h_{RGB}(w_{l,k}), \tag{3.3}$$

where $h_{RGB}(w_{l,k}) = -p(w_{(l,k)})\log p(w_{(l,k)})$, with $p(w_{(l,k)})$ being the relative number of coefficients of a color channel with value k in level l. Hence, h_{RGB} means that the entropy is measured separately for each RGB channel. In the initial implementation the Haar wavelet transform was used over RGB images encoded in 8 bits per color channel.

By measuring the information of a wavelet transform W of the image, the metric is less sensitive to noise and captures the correlation between pixels. If the number of levels is high enough, the remaining information can be considered background. As shown in [Váz07], four levels of the wavelet decomposition is usually enough.

3.3.2 Application of the Multiscale Entropy to Volume Graphics

In order to analyze the behavior of the Multiscale Entropy measure when applied to images obtained from a DVR of a volume dataset, some experiments were performed. Their objective was to assess its performance and its effectiveness. The first experiment consisted of evaluating the quality measure for a dense set of viewpoints around a volume model. The described measure (see Equation 3.3) gives good results, as shown in Figure 3.2 for different volume models. We can observe how it computes higher values (encoded in the figure as warmer and larger spheres, being the pink sphere the best viewpoint) where more details from the volumetric structures are provided.

In order to evaluate the effectiveness of the measure more accurately, we studied how the resolution of the image affects the obtained result. We were interested in achieving the best performance for the algorithm, such that we kept interactivity of the overall inspection task. We analyzed and compared the quality measure for different volume datasets (Table 3.1 shows their the resolution and the voxel dimension) for a dense set of viewpoints (642) on a bounding sphere in order to evaluate how the results depend on the resolution of the viewport. Table 3.2 shows the best and the worst views calculated using different viewport sizes.

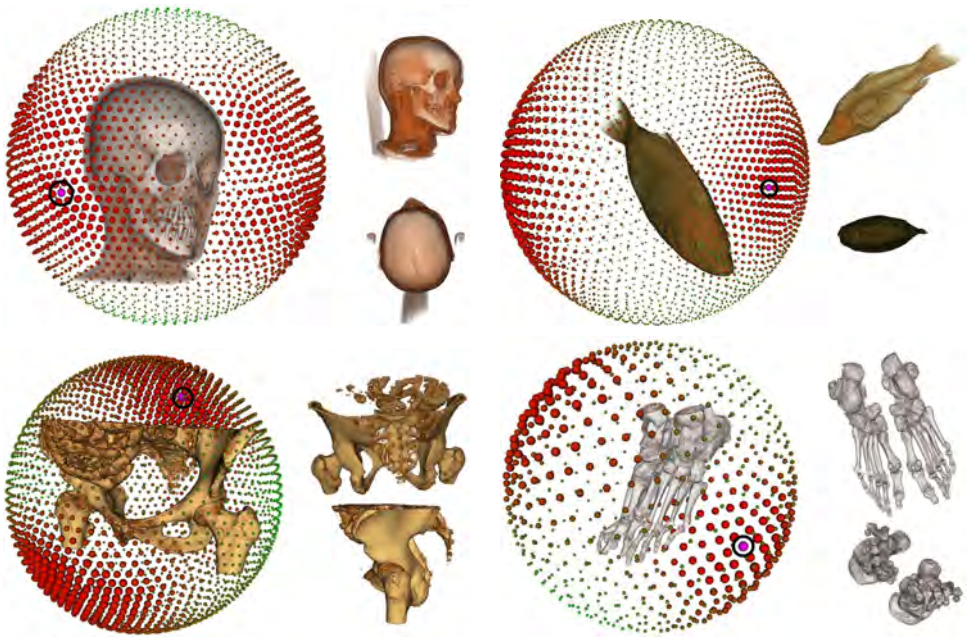


Figure 3.2: Examples of the entropy measure for a dense set of views (2562). Viewpoint quality is encoded in both the color and the node size (the higher the quality the warmer the color and larger the size is). The best view node is painted pink (highlighted with a black circle in the images). The images placed on the right of each view quality sphere show the best (top) and worst (bottom) views according to the proposed measure. Note how the selected views provide a lot of details on the models being inspected.

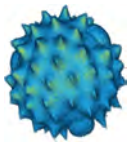
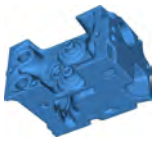
Model		**Resolution Voxel Dimension**	**Model**		**Resolution Voxel Dimension**
Hydrogen		128 × 128 × 128 1.0 × 1.0 × 1.0	Daisy Polen		192 × 180 × 168 1.0 × 1.0 × 1.0
Tooth		256 × 256 × 161 1.0 × 1.0 × 1.0	Engine		256 × 256 × 256 1.0 × 1.0 × 1.0
Head		512 × 512 × 485 0.51 × 0.51 × 0.5	Trunk		512 × 512 × 512 0.74 × 0.74 × 1.0

Table 3.1: Resolution and voxel dimension (in millimeters) of the tested models. The size of the models goes from quite small, to sizes that are almost the largest volumetric models the GPU could fit at the time of the experiments.

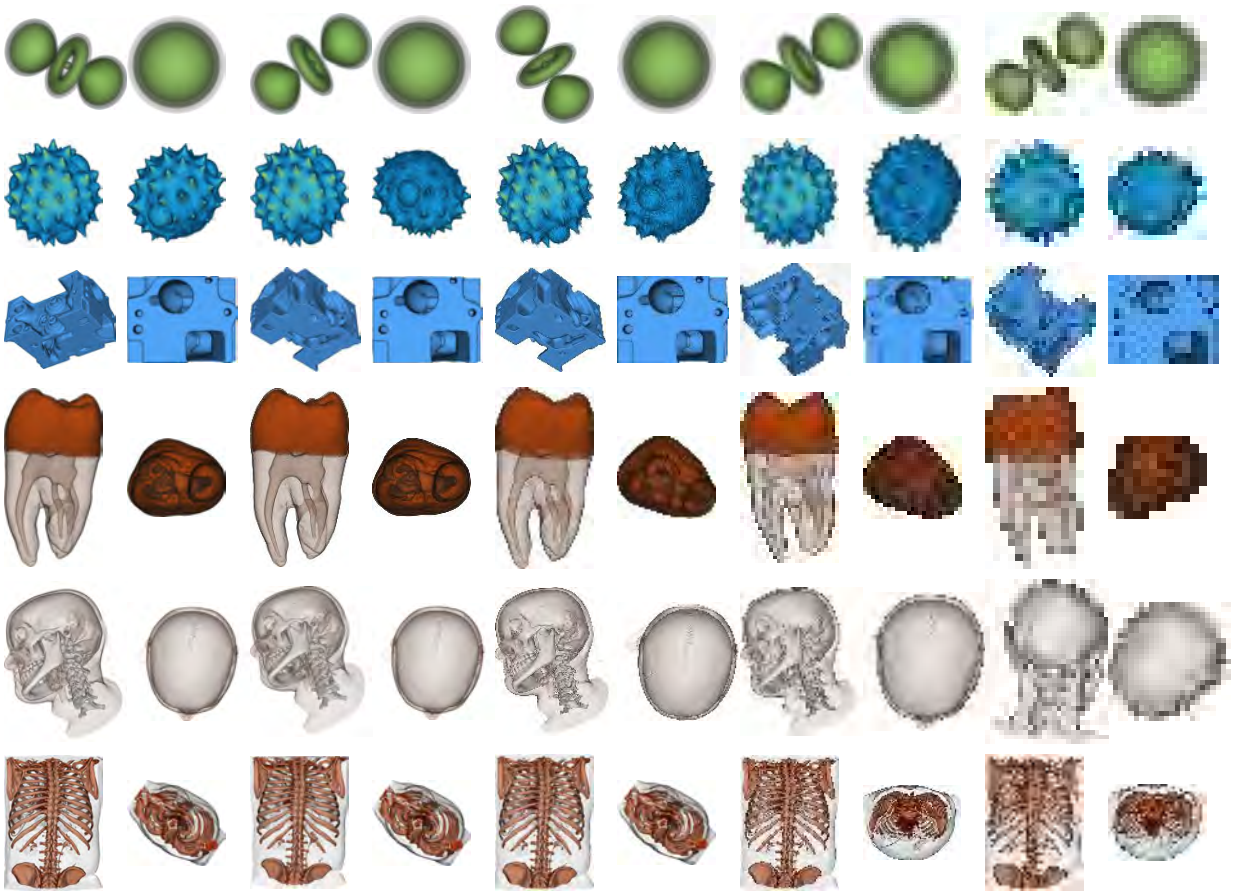


Table 3.2: The best and worst views according to the Multiscale Entropy measure for different viewports. Note that for almost all the tested models, the best and worst views are similar even if we reduce the viewport size down to 64×64 (this means that our measure is quite robust to viewport changes). Notice that the results are very good, since the best views are providing a lot of details on the analyzed models.

Table 3.3 shows the complete analysis we performed. For each model and for each viewport size, the table shows:

- The time needed for the calculation of the best and worst viewpoints. The time differences are due to differences on the performance of the DVR algorithm for each of the tested models. The cost of the DVR algorithm is a function of the resolution of the volume dataset, the used transfer function and the size of the viewport.
- The entropy values for the best and worst viewpoints.

		1024×1024	512×512	256×256	128×128	64×64
Hydrogen	Time (s)	283.73	60.72	16.50	4.85	1.58
	Best Entropy	2.81	3.47	4.65	6.55	9.23
	Worst Entropy	1.65	2.08	2.60	3.36	6.43
Daisy Polen	Time (s)	459.33	90.40	17.34	7.14	1.56
	Best Entropy	3.83	4.43	5.20	6.83	9.39
	Worst Entropy	3.11	3.56	4.28	5.43	7.63
Engine	Time (s)	401.18	68.64	18.30	5.3	1.93
	Best Entropy	4.93	6.08	7.37	9.16	12.13
	Worst Entropy	1.92	2.68	3.45	4.39	6.99
Tooth	Time (s)	254.2	57.63	17.61	4.77	2.07
	Best Entropy	3.0	3.49	4.28	5.88	8.56
	Worst Entropy	1.23	1.52	1.9	2.6	4.67
Head	Time (s)	530.9	122.9	27.5	9.75	3.91
	Best Entropy	6.89	7.68	8.79	10.64	13.41
	Worst Entropy	4.20	4.79	5.54	7.22	9.64
Trunk	Time (s)	680.26	186.35	57.90	21.26	8.90
	Best Entropy	15.14	16.42	17.94	20.66	23.56
	Worst Entropy	9.3	10.16	11.18	12.88	15.04

Table 3.3: The best and worst views information according to our measure for different viewports. For each viewport size, the computation time (in seconds) to generate the 642 images is shown. Also, the values of the Multiscale Entropy measure for the best and worst images are shown. Notice, that for the purpose of fast model previsualization, the time needed for the analysis of a sufficient set of views may become non-interactive.

A qualitative analysis of the results allows us to conclude that Multiscale Entropy measure is an effective measure for computing good views in the area of *Volume Graphics*. Note how the selected views in Figure 3.2 and in Table 3.2 provide a lot of details on the analyzed models. Our analysis showed that resolutions of 256×256 are good enough for the best view computation for all the models we tested, as best views were identical to the ones selected with 512×512 images. Smaller images ($128 \times$

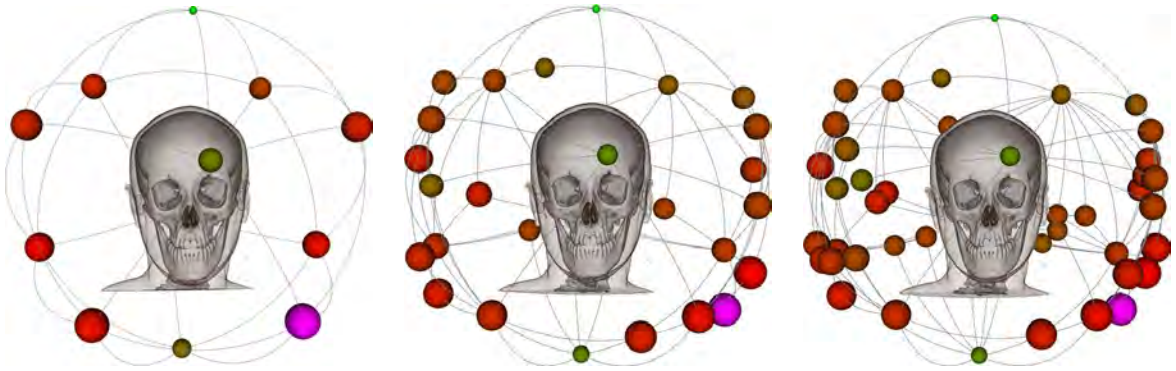
128), generally produce good views (very similar if not equal to the optimal) for most models and thus can be used if a quick response is mandatory.

Unfortunately, for the purpose of fast model previsualization, a brute-force approach is not practical, because the analysis of a sufficient set of views takes non-negligible time. In order to achieve better performance, the best view determination algorithm should use a more clever scheme than a brute-force one (see Section 3.3.3).

3.3.3 Efficient adaptive algorithm

In order to achieve better performance for the best view determination algorithm, we propose an adaptive approach similar to the method used by Gumhold in [Gum02] to compute an optimal light source placement for a given camera. Gumhold considers the fact that it makes sense to assume that the entropy function is Lipschitz continuous [Lip], and then avoid the exploration of unnecessary regions during the search for the maximum entropy.

The algorithm starts from a coarse sampling of the bounding sphere and adaptively subdividing it according to an estimator of the entropy. Initially, the spherical viewpoint domain is sampled on the vertices of a triangle mesh of a icosahedron (see Figure 3.3.a). For each vertex i, the Multiscale Entropy (MSE) function, MSE_i, is evaluated and stored. Also, the minimum MSE_{min} and the maximum MSE_{max} are determined.



(a) (b) (c)

Figure 3.3: Mesh subdivision scheme. Left image shows the initial mesh subdivision. Middle image shows the subdivision after one iteration of the algorithm. Right image shows the final subdivision. Viewpoint quality is encoded in both the color and the node size (the higher the quality the warmer the color and larger the size is). The best view node is painted pink.

Like Gumhold, we assume that the Multiscale Entropy function is Lipschitz-continuous in a neighborhood of the analyzed points. In this sense, the Multiscale Entropy function fulfills the Lipschitz condition:

$$\forall p_1, p_2 \in S^2 : \|p_1 - p_2\| \leq \delta \Rightarrow \|MSE(p_1) - MSE(p_2)\| \leq \delta * L \tag{3.4}$$

where L is the Lipschitz constant, that will be estimated from the Multiscale Entropy function. If we choose two points, $p_1, p_2 \in S^2$, being S^2 the bounding sphere, such that $\|p_1 - p_2\| = \delta$, Equation 3.4 tell us that the absolute value of the gradient of the entropy is bounded by L.

To estimate the Lipschitz constant L, the maximum norm of the entropy gradient over the whole domain has to be computed. The Lipschitz constant L is equal to L_{app} multiplied by a safety factor K_{safety}. The addition of a safety constant K_{safety} is necessary in order to be conservative and avoid missing any maximum – a value of 2 for the safety constant worked well for the tests we performed. This value, L, gives an estimation of the maximum quality variation around a viewpoint.

After the determination of L, the algorithm proceeds to refine the initial approximation of the viewpoints domain (see Figure 3.3.a) in order to adaptively search for the maximum entropy value up to a minimum distance criterion. Our algorithm is slightly different than that of Gumhold. He calculates if a triangle needs to be subdivided and then, it is subdivided in a regular fashion, so 3 new vertices are added, and the entropy function is evaluated at these new vertices. Our approach estimates the point of maximum entropy and only one vertex is added per subdivision, so less new evaluations are required.

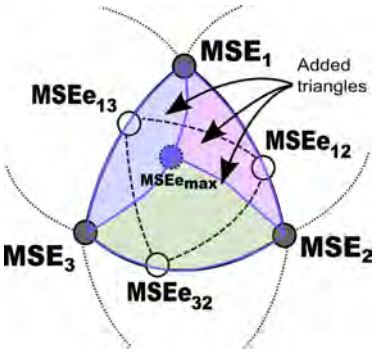


Figure 3.4: Scheme of the adaptive subdivision algorithm. Next subdivision point is selected by estimating the maximum reachable entropy inside the triangle.

This phase consists of two steps. First, for each triangle of the current triangle mesh, the maximum reachable entropy point at each edge ($MSE_{e_1} \cdots MSE_{e_3}$) is estimated using L (see Figure 3.4), $MSE_{e_i} = MSE_i + ||e_i|| * L * MSE_{i_2}$. Then, the estimation of the maximum value inside the triangle $MSE_{e_{max}}$ is estimated by averaging the computed values (MSE_{e_1} to MSE_{e_3}). If $MSE_{e_{max}}$ is higher than MSE_{max}, the actual value is measured by rendering a new viewpoint from this position and adding the subdivided triangle (see Figure 3.3.b). The algorithm stops when none of the estimated entropy values is higher than MSE_{max}, or when those views are too close (i. e. 5 degrees) to existing analyzed positions (see Figure 3.3.c). For illustrative purposes we include the pseudocode of the algorithm in Algorithm 2.

This method obtains similar maximum values than the ones obtained with the brute-force method (see Table 3.5), but at a fraction of the time (see Table 3.4).

Algorithm 2 Adaptative subdivision algorithm.

Initialization

Evaluate the entropy function MSE for each vertex of the initial triangular mesh
Determine the minimum MSE_{min} and maximum MSE_{max} of the sampled entropy values

Step 1: Estimate the Lipschitz-constant

$L_{app} = 0.$ ▷ Maximum entropy gradient computation over the whole domain.
for each vertex v **do**
  $\nabla_v = (0.,0.,0.)$ ▷ Entropy gradient computation at v (∇_v).
  for each edge e incident to v **do**
    $\nabla_v = \nabla_v + \frac{|MSE_{e.v_2} - MSE_{e.v_1}|}{\|\vec{v_1 v_2}\|} \cdot \frac{\vec{v_1 v_2}}{\|\vec{v_1 v_2}\|}$ ▷ v_1 and v_2 are the vertices of the edge e.
  end for
  $\nabla_v = \frac{1}{NumberOfEdges} \cdot \nabla_v$
  $L_{app} = max(L_{app}, \|\nabla_v\|)$
end for
$L = K_{safety} \cdot L_{app}$

Step 2: Determine which triangles have to be split

for each triangle t **do** ▷ Maximum entropy $MSE_{est,t}$ estimation, reachable on each of the edges.
  for each edge $e_{\{1..3\}}$ **do**
    MSE_{i_1} = evaluate the entropy function at vertex e_{i_1}
    MSE_{i_2} = evaluate the entropy function at vertex e_{i_2}
    $MSE_{e_i} = MSE_{i_1} + ||e_i|| * L * MSE_{i_2}$
  end for
  $MSE_{est,t} = \frac{1}{3} * (MSE_{e_1} + MSE_{e_2} + MSE_{e_3})$
  if $MSE_{est,t} \geq MSE_{max}$ **then**
    Calculate the point $p_{MSE_{e_{max}}}$ inside the triangle which produce $MSE_{e_{max}}$
    if $p_{MSE_{e_{max}}}$ differs ϵ degrees from the set of calculated viewpoints **then**
      Entropy MSE_t calculation at the point of maximum estimation
    end if
  end if
  if $MSE_t \geq MSE_{max}$ **then** ▷ the triangle t has to be split in 4 triangles.
    Mark triangle t to be split in point $p_{MSE_{e_{max}}}$
  end if
end for

Step 3:Evaluate the end condition

Determine the new maximum entropy value MSE_{new} inside all the triangles to be split
if $\frac{MSE_{new} - MSE_{max}}{MSE_{max} - MSE_{min}} \leq \epsilon_{\%}$ **then**
  Finish the adaptive subdivision
else
  Actualize the estimation of L, MSE_{min}, and MSE_{max} with the information of the new vertices added
  Go back to Step 2
end if

3.3.4 Results

In order to evaluate the performance of the adaptive algorithm, it has been tested with the volume models of Table 3.1. Timings were taken in a computer equipped with an Intel Core 2 Duo PC running at 3.00 GHz with 8GB of RAM and a NVidia GeForce 280 GTX with 1GB of RAM. Table 3.5 shows the best and worst views using a brute force algorithm versus the adaptive solution using a viewport size of 256×256. As it can be observed, the best views selected by the adaptive approach are very similar to the brute-force approach. Table 3.4 gives the obtained performance compared to the load time. The computation time of the adaptive approach is bounded by a number of little seconds, which is a waiting time acceptable in this kind of applications.

Model	**Resolution**	**Load time (s)**	**Best View**		
			Brute-force time (s)	**Adaptive #views**	**Adaptive time (s)**
Hydrogen	$128 \times 128 \times 128$	0.17	16.50	43	0.98
Daisy	$192 \times 180 \times 168$	0.38	17.34	41	0.95
Engine	$256 \times 256 \times 256$	0.64	18.30	52	1.47
Tooth	$256 \times 256 \times 161$	1.62	17.61	46	1.07
Head	$512 \times 512 \times 485$	6.22	27.5	32	1.70
Trunk	$512 \times 512 \times 512$	29.25	57.9	29	2.62

Table 3.4: Best view computation time for the tested models compared to their loading time. The fourth column shows the best brute force algorithm using a set of 642 positions. The sixth column shows the time needed to obtain the best view using the adaptive method with 256^2 resolution. The number of viewpoints calculated by the adaptive method is shown in the fifth column.

It is true that depending on the volume model, some smaller structures, which can be visible only from a little subset of viewpoints, may be lost when using the adaptive approach. However, in order to focus the inspection analysis in only a portion of the volume model, we have developed the possibility of restricting the analysis to a region of interest defined by a bounded region. In this way, if a region of interest is provided, the algorithm can make a more specific search of the best view for it. Figure 3.9 shows an example of a volume dataset of a whole chest, where the analysis of the best view determination was done only for a small region of the volume dataset: the left kidney.

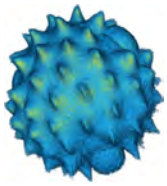
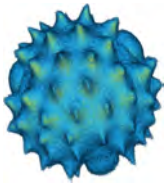
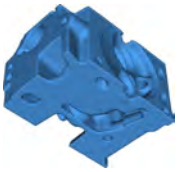
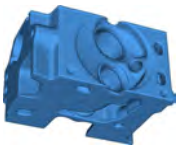
Model	Best View	
	Brute Force	Adaptive
Hydrogen		
Daisy		
Engine		
Tooth		
Head		
Trunk		

Table 3.5: Comparison of the results obtained using the best brute force and the adaptive approaches for a viewport size of 256 × 256. For all the models, the results obtained by the adaptive approach are very close to the brute-force one.

Comparison with other methods in literature

We have also compared the best and worst views generated by the proposed Multiscale Entropy measure for some typical models that have been analyzed by other authors dealing with best view descriptors. The results are shown in Figure 3.5. We have tried to use a transfer function as similar as possible to the one used in the other proposals. The best views we select are roughly the same than the ones selected by other authors. For instance, the tooth model is also analyzed in [BS05, TLB+09, JS06] and they obtain results for best and worst views very similar to ours ([TLB+09] only shows the best view) . The same happens for the daisy pollen grain, where the best view selected by our technique is similar to the one obtained by [TLB+09], as well as for the hydrogen molecule. We consider our methodology more suitable for the medical domain because we do not require any parameter tweaking, such as the method of Ji and Shen does [JS06], and also our proposal is much more efficient in terms of performance due to our adaptive solution. All of the proposals analysed in Section 3.1, perform an exhaustive computation around the viewing sphere using from 128 to 768 viewpoints, so the computation time can be very highly dependent on the volume dataset used.

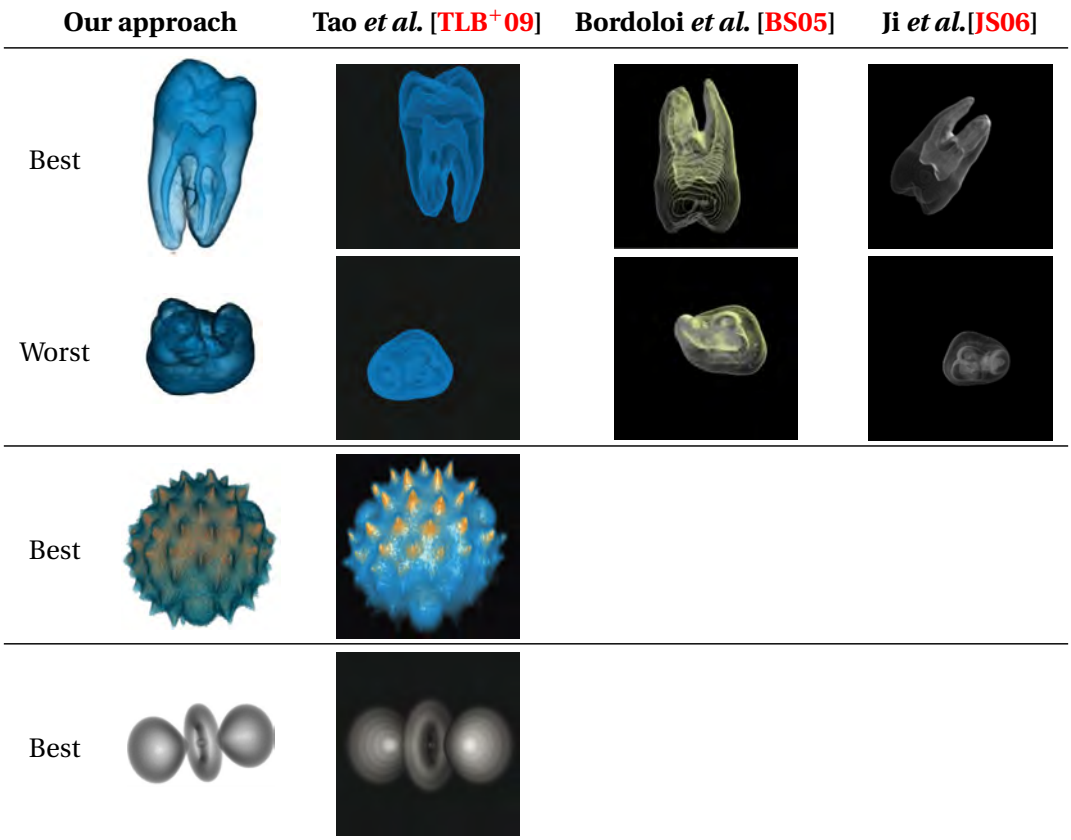


Figure 3.5: Comparison of the obtained views with other methods from literature ([BS05, TLB+09, JS06]). Our views almost coincide exactly with the ones selected by other methods. We have tried to use a transfer function as similar as possible to the one used in the other proposals. Images are taken from [BS05, TLB+09, JS06].

3.4 Representative views selection

Once we are able to evaluate a single view, we need a method to select a representative set that maximizes information and reduces redundancy. We address this goal with the use of the Normalized Compression Distance, a complexity-based metric.

3.4.1 Preliminaries

Normalized Compression Distance is a universal metric of distance between sequences. It has its roots in *Kolmogorov complexity* (also known as *algorithmic complexity*). We will briefly detail here some concepts of algorithmic complexity. The interested reader can refer to Li and Vitányi's book [LV93] for a deeper and more theoretical introduction.

The **Kolmogorov complexity** $K(x)$ of a string x is the length of the shortest binary program to compute x on a universal computer (such as a universal Turing Machine). Thus, $K(x)$ denotes the number of bits of information from which x can be computationally retrieved. As a consequence, strings presenting recurring patterns have low complexity, while random strings have a complexity that almost equals their own length. Hence, $K(x)$ is the lower-bound of what a real-world compressor can possibly achieve. The **conditional Kolmogorov complexity** $K(x|y)$ of x relative to y is the length of a shortest program to compute x if y is provided as an auxiliary input. Both Kolmogorov complexity and conditional Kolmogorov complexity are machine independent up to an additive constant.

Bennet *et al.* [BGL+98] define the **information distance** between two binary strings, not necessarily of the same length, as the length of the shortest program that can transform either string into the other one, both ways. The information distance is a metric because it satisfies the metric inequalities. Li *et al.* [LCL+04] present a normalized version of information distance called *similarity metric*, defined as:

$$d(x,y) = \frac{max\{K(y|x), K(x|y)\}}{max\{K(x), K(y)\}} \tag{3.5}$$

The authors also prove that this metric is universal (two files of whatever type which are similar with respect to a certain metric are also similar with respect to the similarity metric). Being Kolmogorov complexity not computable, it may be approximated with the use of a real-world compressor, leading to the **Normalized Compression Distance (NCD)**. Given two files x and y, the Normalized Compression Distance between them can be computed, up to an ignorable precision, as:

$$NCD(x,y) = \frac{C(xy) - min\{C(x), C(y)\}}{max\{C(x), C(y)\}} \tag{3.6}$$

where function $C(f)$ is the size of the compression of a certain file f, and xy is the concatenation of files x and y. Although the similarity metric has values in [0..1], NCD values are usually in the range of [0..1.1], due to compressor imperfections. NCD has been successfully used for applications such as language classification and handwriting recognition [CV05]. Cilibrasi and Vitányi [CV05] also analyzed the conditions that compressors must fulfill in order to be used for computing the Normalized

Compression Distance. The data compressors with these properties are dubbed *normal compressors*. Most real-world compressors do fulfill those properties, at least to a point where they are usable for NCD computation. Some of the candidates are: stream-based (*zlib*), block-based (*bzip*), and statistical (*PPMZ*) compressors. As studied by Cebrián *et al.* [CAO07], in the case of *bzip2*, the *best* option works properly for files up to 900KB before being compressed. Larger sizes make the comparison processes less effective.

Vázquez and Marco in [VM12] analyzed the performance of NCD applied to color images. They provided a comprehensive comparison on the efficiency of the different compressors and which are the most suitable image format for image comparison using NCD. They found some interesting results respecting the robustness of NCD to rotation, translation and scaling depending on the compressor and the image format. They concluded that NCD worked well with *bzip2* compressor using *ppm* images, although translation behaved better than rotation.

3.4.2 Representative views selection algorithm

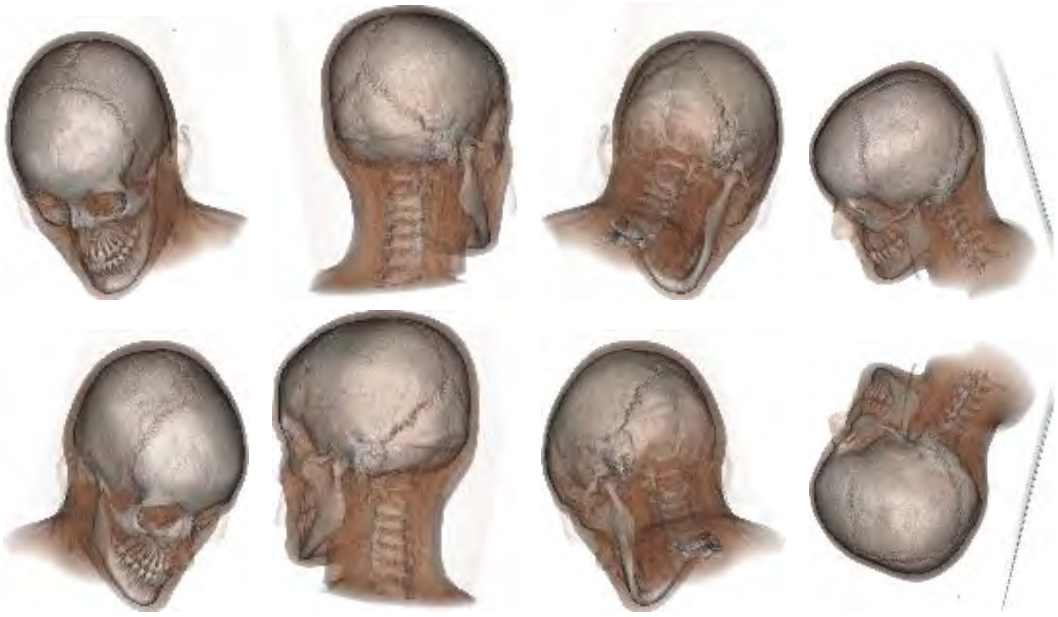
The set of representative views of a volume dataset may be determined by a set of visually different views of a model. This set can be used to illustrate model libraries or serve as key points for automatic exploration path construction.

We have developed an approach that uses a greedy scheme to determine the view set that adequately represents a model, starting from the images calculated to determine the best view of the same model (see Section 3.3). At high-level, the algorithm starts with the following steps:

1. Select the best view B_0 obtained by the adaptive algorithm.
2. Measure the distances from B_0 to the remaining views obtained by the adaptive algorithm. This process involves the evaluation of the similarity between two images.
3. Next representative view B_1 is the one at the highest distance from B_0.

The algorithm cost is linear with the number of views, being the comparison process the most costly one. In order to evaluate the similarity between two views, we use the *Normalized Compression Distance (NCD)* (see Equation 3.6). The NCD between two views X and Y is evaluated using the images of the model taken from the viewpoints X and Y. These images are stored as files. Then, we take each image file pair and concatenate the files and store the concatenated pair in a new file, and then, all these files are compressed. Afterwards, distance can be computed using the compressed original files and the concatenated compressed one. The algorithm uses the ppm file format in conjunction with the bzip2 compressor.

Once we have the two initial representative views, if we want to gather the missed information by these two, we can proceed the same way: compute the distances from the remaining views against B_1 and choose as new view B_2 the one that maximizes the geometric average of the distances to B_0 and B_1. This process can be repeated several times, but three or four are usually enough for most models.



(a) (b) (c) (d)

Figure 3.6: Top row shows four random views of the head model, and bottom line shows the view whose distance is minimum to the one on the top. Note how our similarity measure is robust with respect to rotation and symmetry. Images are measured as are, that is, without any rotation to align them.

3.4.3 Results

Figure 3.6 shows, from a set of views around the head model, the four corresponding images whose NCD distance is the smallest to the given one. Note how symmetric views are correctly ranked as being very similar.

The first three columns of Table 3.8 show the three representative views for the tested models. Observe that these views show substantially different visual information.

Representative views computation is quite fast (see Table 3.6). Concatenation and compression are the most costly parts. Note that, as the representative view selection algorithm works with the images calculated by the best view determination algorithm, the algorithm cost is proportional to the number of the obtained images. Fourth column shows the number of images calculated by the best view determination algorithm.

Because we measure *NCD* over rendered views, our representative view selection method yields views that *look different*. For most models, this will be a key issue, as selecting views according only to the visible voxels does not guarantee that the final rendered information will enhance our knowledge of the model, since it might yield a symmetric view that would not add substantial new information.

Model	Slices	load time (s)	# images	Representative views time (s)
Hydrogen	$128 \times 128 \times 128$	0.17	43	2.36
Daisy	$192 \times 180 \times 168$	0.38	41	2.62
Engine	$256 \times 256 \times 256$	0.64	52	4.46
Tooth	$256 \times 256 \times 161$	1.62	46	2.3
Head	$512 \times 512 \times 485$	6.22	32	1.89
Trunk	$512 \times 512 \times 512$	29.25	29	2.41

Table 3.6: Performance of the representative views selection algorithm for different models compared to their loading time. The resolution of the images is 256×256. Fourth column shows the number of images obtained by the adaptive method. All times are in seconds.

Optimization

There is a simplification we can do to further accelerate the representative view selection process. The most important cost is incurred in the compression of the images. We already reduced their size by reducing the viewport, while maintaining the quality of results. What we will do now is to reduce the amount of information per pixel. So that, instead of storing an RGB value for the representative view selection, we will work with grey scale images. The results will slightly different, but in all the tested models, the results were perfectly acceptable. In Figure 3.8, we can see a comparison for the tested model. Table 3.7 shows the time required for the representative view selection task using both approaches.

Model	# images	Representative views time (s)	
		RGB	grey scale
Hydrogen	43	2.36	0.77
Daisy	41	2.62	0.70
Engine	52	4.46	1.33
Tooth	46	2.3	0.66
Head	32	1.89	0.62
Trunk	29	2.41	0.84

Table 3.7: Performance of the representative views selection algorithm for the tested models. All times are in seconds. Third and fourth columns show the time required when using the RGB and grey scale images respectively.

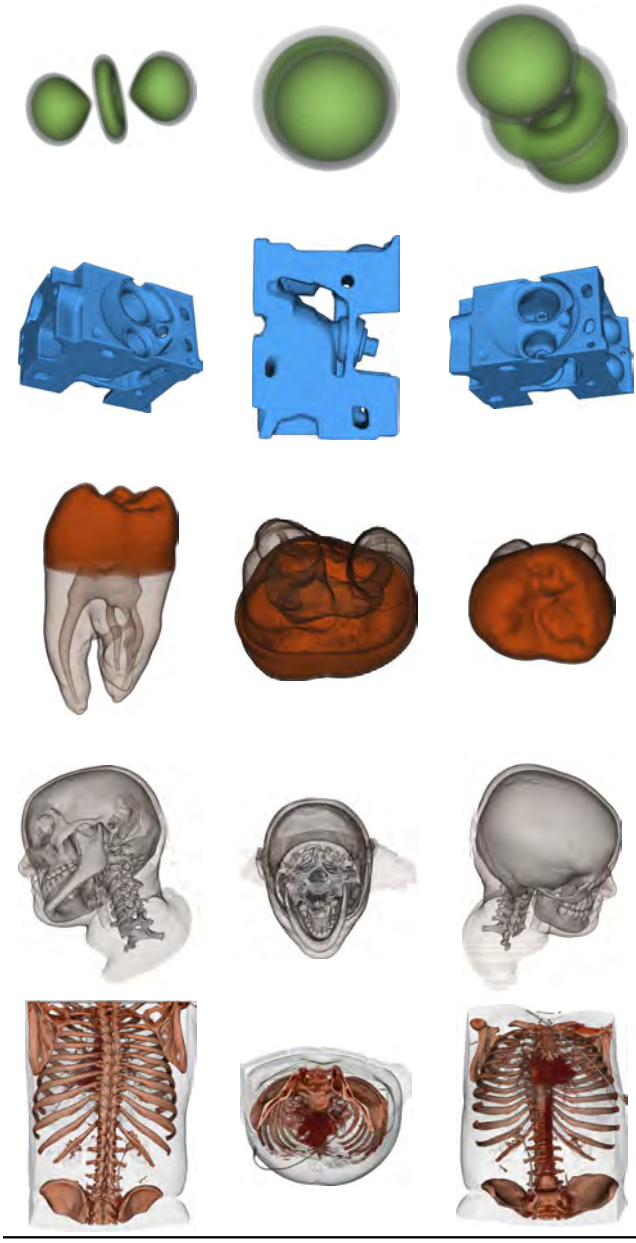
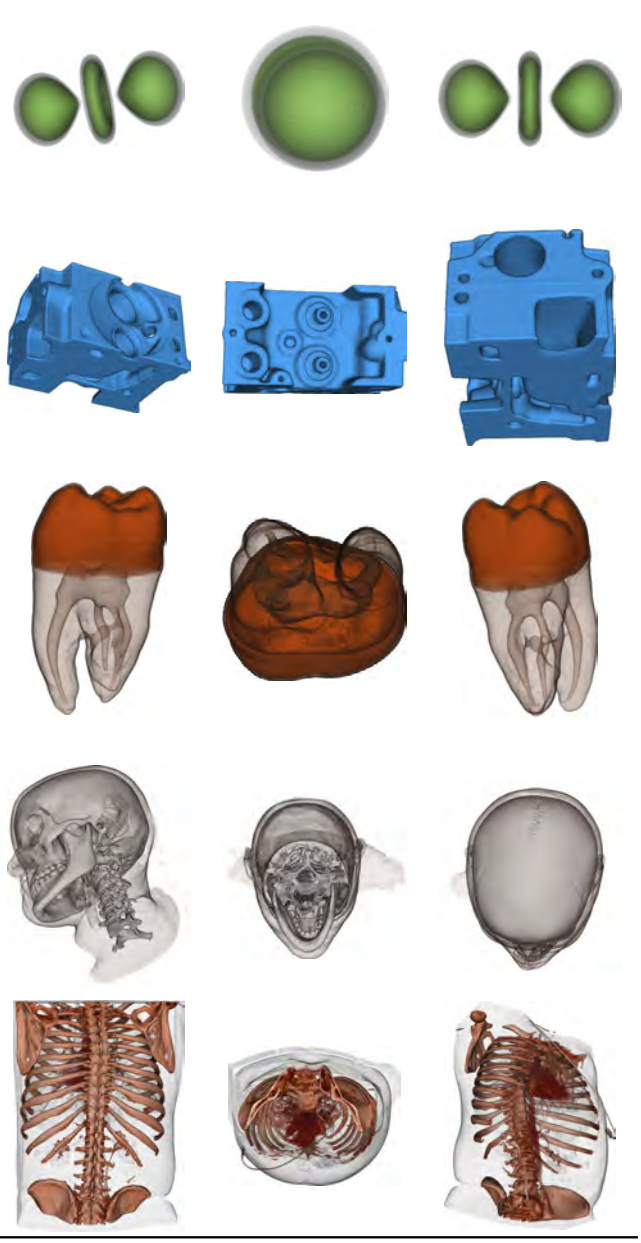
RGB Calculation	Greyscale Calculation
	

Table 3.8: Comparison of the representative views selection with the RGB and the grey scale strategies. For some models, some views might change a bit, but they are still acceptable.

3.5 Exploration path construction

The last step of our system is the construction of an exploration path that allows users to inspect the volume model, obtaining in this way an initial comprehension of its overall structures. Taking the representative views set as input, we generate a camera path which consists of a set of positions located on a bounding sphere of the volume model or a region of interest previously selected. We propose to use as key points of the path the representative viewpoints selected in the previous section and ensure visiting all of them. The viewpoint with the highest entropy is visited in second place. This allows the camera to visit its surroundings that, intuitively, should be more informative than the rest. Additionally, the speed of the camera is reduced when visiting points nearby the best one because, intuitively, they will show a higher amount of valuable information.

3.5.1 Exploration path construction algorithm

The path construction algorithm determines the minimal length path that passes through all the representative views. As we usually have only three to four views, an exhaustive search of the shortest path that ensures we pass through the best view in second position is computed instantaneously. We call this approach: the *simple exploration algorithm*. The algorithm proceeds as follows: once the order of visiting each representative viewpoint has been established, the algorithm has to compute a set of viewpoints (keyframes) in order to compute the final pre-visualization. From one representative viewpoint to the other, equally displaced keyframes are computed at a constant rate. This step size is measured in degrees (this size is named *Step* in Figure 3.7) . Furthermore, the speed of the camera is reduced when visiting points nearby the best view because, intuitively, they will show a higher amount of valuable information.

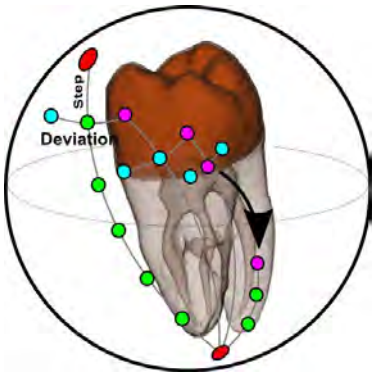


Figure 3.7: Scheme followed to build the two exploration paths: the simple path is painted in green and the improved path scheme in cyan. In the improved path, for each keyframe the camera may deviate a certain distance (*Deviation*), and for three candidate points, the entropy is analyzed and the one with the highest entropy is chosen as the next keyframe. Once the distance between the current keyframe and the target (a representative viewpoint) is less than a certain value, the scheme for computing the shortest path is activated.

In order to maximize the information gathered through the exploration, we introduce an improvement: the camera may deviate from the short path up to a certain angular distance at each step (this angular distance is named *Deviation* in Figure 3.7). For each step, the entropy of three candidate points, which are placed toward the destination point and equally separated from the shortest path, is analyzed. The one with the highest entropy is chosen. The total allowed deviation is limited and reduced as long as we get closer to the next representative viewpoint. This ensures we are visiting all the representative viewpoints.

3.5.2 Results

Figure 3.8 shows the simple path and the improved path computed by our algorithm for the exploration of some of the tested volume models. The improved approach gathers a higher amount of details than the simple method. The simple path calculation time is negligible, but the improved path requires the evaluation of entropy at each step, and therefore, the time will depend on the number of steps we want to produce. For tests we set the step size to 8°. The deviation angle was set to 30°. The time required ranges from 0.88 seconds for the smallest model (hydrogen model) to 4.46 seconds for the biggest one (trunk model).

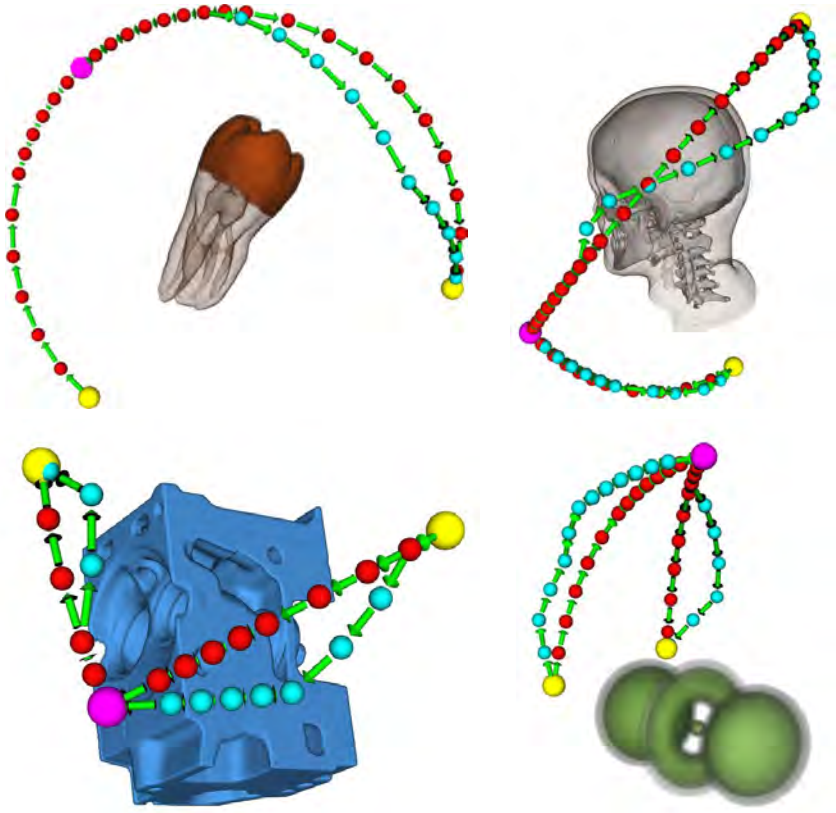


Figure 3.8: Exploration paths around some of the tested models. The simple exploration path is coloured in red and the improved path is coloured in cyan. The best view is coloured in pink.

An example of an exploration path built around a selected region of interest of a thorax model is shown in Figure 3.9. The region selection tool allows to determine a spherical region around a kidney, and the analysis is performed only using this region of the volume.

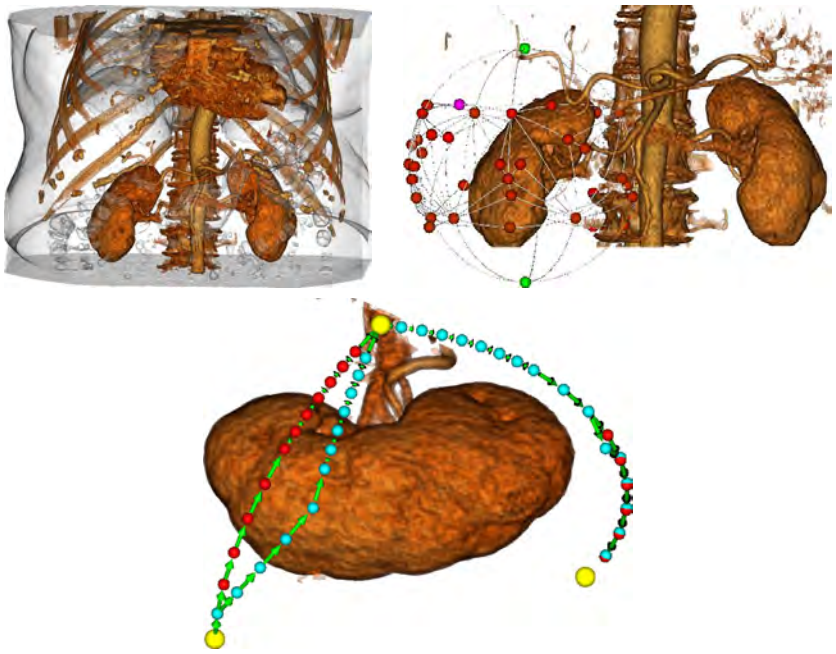


Figure 3.9: Analysis of a region of interest around the kidney. Top-left image shows the overall model. Top-right image shows a close-up to the model, where the region of interest has been defined around the left kidney. This image also shows the adaptive subdivision computed to find the best view. The bottom image shows the two versions of the exploration path construction (the simple path is coloured in red and the improved path in cyan).

3.6 Conclusions

In this chapter we have presented an algorithm based on the Multiscale Entropy measure [Váz07], for the automatic selection of the best and the representative views of a volume model and the automatic generation of a exploration path for it [VMN08]. The described techniques allow users to obtain a set of representative views in a short time and permits the generation of inspection paths at almost no extra cost. In most cases the required total time (adaptive best view selection + representative view selection) is comparable to the loading time. Moreover, we may perform the whole process in roughly one second for most of the models tested if we restrict ourselves to offscreen-viewports of 128×128. Our system works with a raw volume dataset in which only has been defined a transfer function for its visualization. This is not a strong requirement since, usually, the standard format used to save the medical images (DICOM) has information about the captured anatomical structures and, therefore, an standard transfer function could be used.

There are a lot of situations in the daily routine of physicians where a quick first inspection of a medical model can be of great utility. For instance, radiologists sometimes need to scan the patients database in oder to look up a particular patient study. Although physicians know what they are looking for and where it is located, a help in the first task of understanding how the model is positioned or oriented is welcome in order to quickly know how to reach the area of interest. For all these situations, we consider that our approach facilitates the overall task of inspecting medical models. Moreover, to have a better feeling of if medical doctors consider useful our system, we asked two medical doctors: a maxillofacial surgeon and a cardiologist. Both of them agreed that the resulting views were useful for the preparation of medical education materials. Moreover, the cardiologist suggested that having the videos prepared and delivered to her iPhone would facilitate to perform a fast diagnosis. Concretely, she believes that aneurysms at the aorta arch can be easily detected, and, in some cases, other aneurysms may be initially diagnosed. This conforms to previous research on video creation for diagnosis [IbHT+02], although our video generation algorithm is more general, in the sense that can be applied to any models, and cardiological diagnostic may be an useful result, but was not our goal. Our method is adequate for fast previsualization of models, and can be used to select initial inspection views for limited capacity devices such as iPhones or low-end PCs.

Comparing our method with the literature analyzed in Section 3.1, the relevant differences are:

View quality measure There is a wide range of different options. They can be classified into object-oriented approaches [BS05, TLB+09, WZZ+07] (which take into account some characteristics of the overall volume), viewpoint-oriented approaches [JS06] (which compute the measure from the image visualized from the current viewpoint) and hybrid approaches [TFTN05] (which merge both criteria). We are interested in measuring the information effectively seen by the user with almost no-extra cost and above all without any user intervention. Viewpoint-based measures are the most suitable, since they are focused on the final visual information. Therefore, they are easy to compute and do not require a costly preprocess as most of the object-oriented approaches. In this sense, our measure only needs to calculate the final image a user would visualize (using a DVR algorithm). In terms of results, Table 3.5 shows the best and the worst view for some of the models used in the literature.

Adaptative solution All the methods analyzed in Section 3.1 perform an exhaustive computation around the viewing sphere using from 128 to 768 viewpoints. As it has been shown in Section 3.3.2, the time needed for performing these computations overtakes the minimum performance for being considered an interactive approach. As far as we know, our method is the first in the area of Volume Rendering that does not perform an exhaustive computation around the viewing sphere. As explained in Section 3.3.3, our approach performs an adaptive search to find the best viewpoint, saving a considerable amount of time in performing this task.

Representative views Only few works in the literature determine a set of representative viewpoints. Bordoloi *et al.* [BS05] proposed a method which considers that two views are similar if they have

a similar entropy value, without worrying about the contents itself. In this sense, we consider that our approach has a more appropriate definition on the similarity of views since we use the information contents in the view. More recently, Grau *et al.* [GPE+13] have also studied the similarity between two images. They base part of their formulation on the semantic information of the visualized structures of the volume dataset, which implies a preprocess in order to obtain it.

CHAPTER 4

VIRTUAL MAGIC LANTERN: AN INTERACTION METAPHOR FOR ENHANCED MEDICAL DATA INSPECTION

Virtual Reality (VR) technology offers several advantages for scientific visualization, among them, the ability to perceive 3D data structures in a natural way. The recent advances in medical imaging, graphics hardware, and virtual reality technologies at affordable prices have empowered the development of Virtual Reality medical applications. However, in order to facilitate its integration in the clinical practice, these applications have to be effective in terms of performance and accuracy, easy to learn and use with a friendly and intuitive user interface, and minimizing the data preprocessing required.

Direct Volume Rendering (DVR) provides a means for spatial interpretation of medical images. But as volumetric structures may occlude each other, the analysis of the 3D relationships among different anatomical structures may be difficult. Several methods have been proposed to address this issue in the past. However, there is still wide room for improving the interaction techniques tailored to facilitate the inspection process in VR environments.

This chapter presents the *Virtual Magic Lantern* (VML), a new technique for helping users in the task of medical data inspection. It addresses the occlusion management problem and facilitates the inspection of inner structures without the total elimination of the exterior structures, offering in this way, a *focus+context-based* visualization of the overall structures.

The rest of this chapter is organized as follows. Section 4.1 summarizes the relevant works in the area of volume exploration. Section 4.2 introduces the VML approach and delineates the interaction process. Section 4.3 details the implementation of this technique. Results and a user study are presented in Section 4.4. Finally, Section 4.5 presents the conclusions.

4.1 Related Work

Different techniques and strategies have been proposed with the objective of facilitating the identification and exploration of features or regions of interest in volume datasets. In literature, these are usually classified in the following categories: *cutaway views*, *focus+context* visualization, *volume deformation* methods and *lens and distortion* approaches.

Cutaway views is a common paradigm that eliminates part of the volume from the rendering. This can be achieved by simply defining a cutting plane, or using more complex cutting geometries as in Weiskopf *et al.* [WEE03]. Another example of cutaway views can be found in the anatomical atlases by Höhne *et al.* [HBR+92], where the inner parts of anatomical datasets are visualized by using cutting planes defined by the user through the specification of three points on the visible object. McInerney *et al.* [MB06] used 3D slice plane widgets as a 3D interaction model for exploring volume image data. The virtual resection technique by Konrad-Verse *et al.* [KVPL04] also performs cuts on anatomical models. This method generates a deformable clipping plane from user-defined resection lines on the surface of the organ to cut. Then, the deformable plane can be manipulated using the mouse in order to perform the desired resection. Li *et al.* [LRA+07] let the user to modify the appearance of the elements in the clipped region by means of a rigging system that defines how the cutaway affects each structure. Most cutaway techniques do not preserve the context surrounding the structure of interest.

Focus+context visualizations add cues for the user to know which information is being hidden. These techniques may be implemented by modifying the transfer function of the structures placed between the user and the region or feature of interest like in Bruckner *et al.* [BGKG06]. For those systems to work, it is necessary to develop new tools that give users an easy definition of regions of interest (commonly called *ROI*) and importance information [BHW+07], and eventually, the focus of attention [VFSG06]. Svakhine *et al.* [SES05] describe a framework to create anatomical illustrations by highlighting the focus structure while deemphasizing less important regions. They use different shapes for establishing the focal region and combine different rendering techniques in order to improve the amount of information shown to the user. Since their objective is the production of good volume illustrations, they do not provide any interaction mechanism to manipulate the region of interest. Most of the *focus+context* techniques assume that a segmented volume dataset, a set of isovalues, or a set of focus layers have been previously defined by the user [KSW06].

Volume deformation techniques take a totally different approach. These methods are based on the interactive manipulation of volume models to create feature-based cutaway visual effects inspired by surgical metaphors. The main differences among them rely on the deformation technique used to achieve the cutaway, and the kind of interaction defined by the user. For instance, in [MRH08] a technique for interactively create deformations similar to those commonly presented in anatomical textbooks is exposed. With the use of deformations, the shape of an object is modified (not its position as in exploded view) to reveal an obscured view onto the deformed object itself or other objects. In [CSC06] a new approach is presented which allows complex deformation of volumetric models for the creation of illustrative visualizations through interactive manipulation of volumetric models. Chen *et al.* [CSS08]

propose a set of interactive manipulation tools for drilling, lasering, peeling, cutting and pasting different layers of volume data sets. Birkeland *et al.* [BV09] generate automatic peel-away visualizations of segmented features of interest by deforming and translating a certain region of the model through the use of a vector field that contains the inverse transformation of the peeled structure. McGuffin *et al.* [MTB03] propose to separate the different parts of the volume as if the object was formed by different pieces that can be moved independently. They dubbed their method *Exploded views*, which is a new term used to categorize the methods based on it. Bruckner and Gröller [BG06] automated this process by splitting the object into a set of different parts that are separated through the use of repulsive forces. Notwithstanding, deformation-based methods have two major disadvantages. First, they require accurate selection of the region of interest, which implies previous knowledge of the model. Second, data preprocessing (such as the segmentation of the structures of the model) is necessary. Segmentation is also common to previous approaches [BV09, MRH08, BGKG06], and has the disadvantage of usually being quite costly.

In ***Lens and distortion*** approaches, a virtual lens is placed in front of the volume, between the user and the region of interest. As a consequence, the information is amplified and distorted [BSP+93], so that the user may see the region of interest with a higher detail. Zhou *et al.* [ZHT02] use a sphere as a focal region which will be rendered using a DVR algorithm, the rest of the volume will be rendered using a simple technique that informs about the overall shape of it. Wang *et al.* [WZMK05] allow the user to determine a focus point and modify lens parameters for the focus area; this information is used on a GPU-based ray casting implementation [HKRs+06]. Brown and Hua [BH06] propose a platform for augmented virtual reality that displays the focus view in a separate display that acts as a window in the virtual or real world.

Since we are interested in enhancing the user experience in the *exploration* of a volume dataset using DVR, we are especially concerned on real-time inspection techniques that do not require preprocess of the volume data (such as the segmentation) and may be used in VR systems. Most of the previous approaches do not fulfill those constraints because they were designed for a desktop-based application, and thus may not be easily ported to VR environments. This kind of difficulty can be either due to the performance or because of the used interaction technique. Moreover, in several of the previous approaches related to *focus+context*, data preprocessing is necessary and it is usually quite costly (such as the segmentation of the structures of the model). We consider that it is very important to develop easy techniques which help the user in the exploration task without the need to worry about tweaking parameters or complex setups. Note that users are not used to VR environments, so we have to focus on offering them an intuitive interface and a natural working flow.

Simultaneously or beyond the publication of our proposal, other authors presented other *focus+context* visualizations based on similar principles than us. Luo *et al.* [LIGGM09] proposed a new illustrative technique for focusing on a user-driven region of interest while preserving context information. The region of interest is defined using different shapes based on the superquadrics family functions modeled as a distance function which controls the opacity of the voxels within the probe. This proposal defines two different rendering styles for being used inside the probe and outside it. They are

based on silhouette enhancement and using non-photorealistic shading techniques to improve shape depiction.

Kirmizibayrak *et al.* [KWBH10] presented a visualization metaphor inspired in a volumetric Magic Lens-paradigm (ML). The content of the ML region consisted of the rendering of one of the available data sources (CT images and CFD simulations). Their approach was presented as a useful technique to guide surgeons during a laryngoplasty procedure. In a posterior publication [KRW+11], they used mutimodality data previously co-registered and they proposed to interactively edit the way the volume was visualized. In this sense, by using the cylindrical lens (ML) as a volumetric brush, they allow to handle the visualization of arbitrary shaped regions by moving the lens and, optionally, changed the data source visualized in the interior of the ML (additional details can be found in Section 4.4.2).

4.2 The Virtual Magic Lantern metaphor

In order to improve the exploration of complex medical datasets in a VR environment, we concentrate our efforts on the visualization of interior and exterior structures in a *focus+context-based* paradigm. In order to accomplish this, we have developed a new interaction metaphor called *Virtual Magic Lantern* (VML). It is inspired by the *Magic Lantern*, a device intended to project images onto a wall through the use of sunlight or candle light and a convex lens as an objective to focus the images [Mag]. The Magic Lantern is the precursor of modern projectors and its invention is not clear (see Figure 4.1, taken from [Mag]).

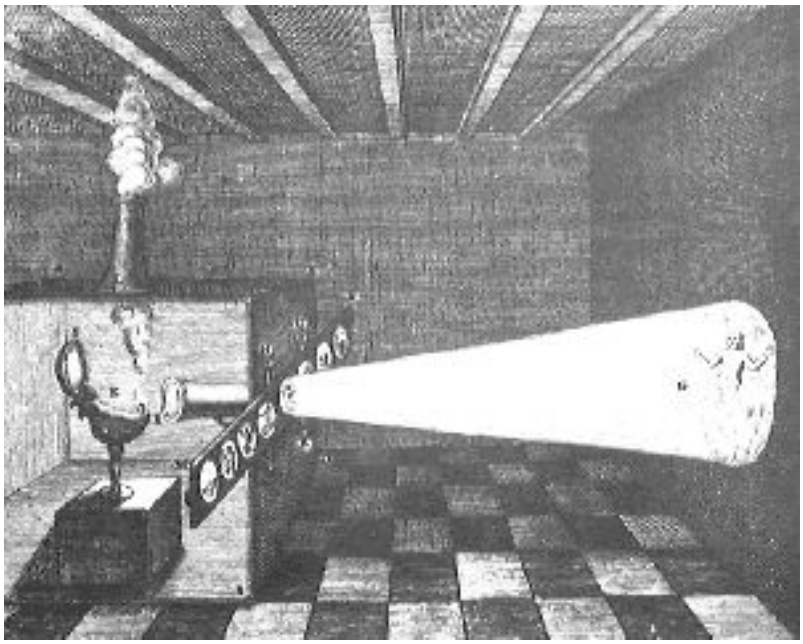


Figure 4.1: Image from the 1671 edition of *Ars Magna Lucis et Umbrae* where the Magic Lantern is depicted. © The Magic Lantern Society 2007. All rights reserved.

The VML metaphor is illustrated in Figure 4.2. We envision the following scenario: a user is exploring a volume dataset using a specific transfer function (TF) which allows her to see the exterior of an anatomical region. When the user wants to inspect the interior of a region of interest, she focuses a virtual lantern (guided with a 3D pointer) to this region. The pointer casts a cone that imitates the light that would be casted by a lantern. In this cone, we use a second TF which allows the interior to stand out (by making transparent the structures the user wants to remove). As it can be observed in Figure 4.3, the boundary of the region of interest is enhanced with the visualization of the original medical images (the raw volume dataset) using the classical grey-level representation.

Our method gets the inspiration from the Magic Lantern [Mag] in the definition of a user-driven region of interest where the virtual lantern *projects* a different kind of visualization from the rest of the volume dataset. Its main use is the inspection of different anatomical structures that cannot be rendered simultaneously with the same transfer function as we can see in the Figure 4.2.

The use of the VML metaphor has several advantages, the most important is that most of the people have used a lantern many times in order to inspect a low light environments, and therefore its usage is totally familiar to us.

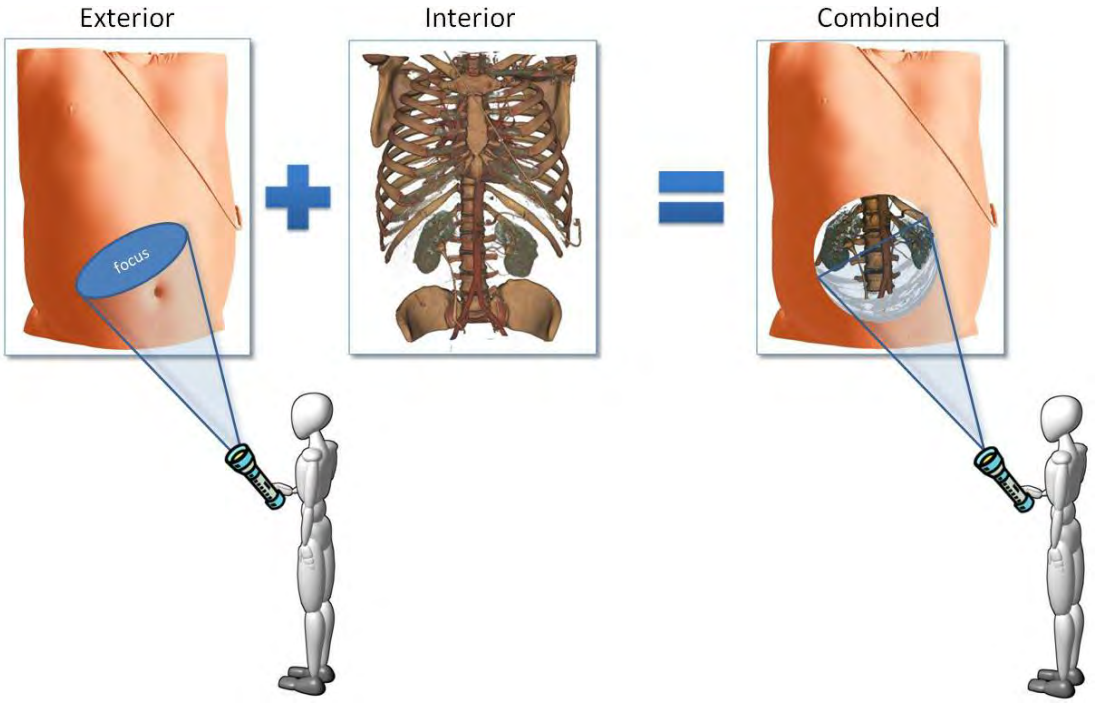


Figure 4.2: The Virtual Magic Lantern metaphor. Our method uses the simile of a virtual lantern to define a user-driven region of interest guided with a 3D pointer. This region is rendered using a second transfer function.

We have also proposed a second interaction metaphor, named Virtual Magic Window (VMW), that can be seen as a particular case of the VML metaphor. VMW allows the user to locate a virtual window with the help of a 3D pointer (see Figure 4.3). The region of interest becomes the part of the volume

that can be seen through it. VMW does not provide as much contextual information as VML, especially on the boundary of the region of interest. VMW produces a similar effect than other established Magic Lens approaches ([BSP+93] [WZMK05]), although our method is intended for VR environments. Furthermore, it provides higher flexibility in shading style, or the shape of the analyzed region.



VML Zoom-in of the VML VMW Zoom-in of the VMW

Figure 4.3: Advanced inspection of a medical dataset using the VML and the VMW metaphors. VMW does not provide as much contextual information as VML, especially on the boundary of the region of interest. In the VML metaphor, the boundary of the region of interest is enhanced with the visualization of the original medical images using the classical grey-level representation.

In the development of both metaphors, two design aspects had been taken into account:

- The definition of the shape projected by the lantern, named the VML shape (see Section 4.2.1).
- The guidance of the lantern by the user (see Section 4.2.2).

4.2.1 Definition of the region of interest with the lantern

In order to give the user more freedom of choice, we allow them to change the size of the region of interest by using a joystick provided by the 3D pointer device. In this way, changing the aperture angle of the cone, users can increase or reduce the region of interest pointed by the virtual lantern.

Moreover, the initial design of the metaphor simulated to move a cone which started from the 3D pointer. As a consequence, if the user moved towards the projection screen, the conical region of interest reduced its size. Although the user was able to modify the cone angle at will, this might produce a somewhat disturbing effect. In order to solve this, the shape of the VML metaphor was substituted by a cylinder (see the rendering result in the left image of Figure 4.4). This solves this problem, while maintaining an intuitive intersection shape. The visual effect is not very different from the cone unless the user moves forward and backward, where the constant radius makes it more comfortable. Users can modify the size of the region of interest by modifying the radius of the cylinder using the joystick.

Additionally, when analysing possible applications with physicians, we were suggested to use a prism with square basis as the lantern's shape because of its similarity to the classical views: axial, sagittal and coronal (see the right image of Figure 4.4). We were also suggested to add a final cap to the

region of interest, that determines the maximum reachable depth by the rays that are traced inside the VML region. This cap is initially placed at a fixed distance, but the user can modify its position, thus enlarging or shortening the exploration volume. Like in the case of the size of the region of interest, the distance of the cap is controlled using the second axis of the joystick. We may see the effect of adding and removing the final cap in Figure 4.5.

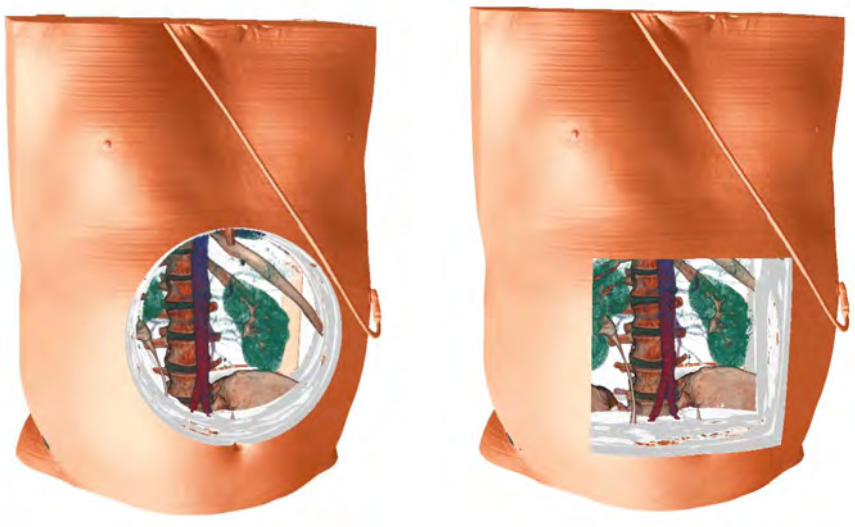


Figure 4.4: Cylindrical and prism shapes of the interaction tool.

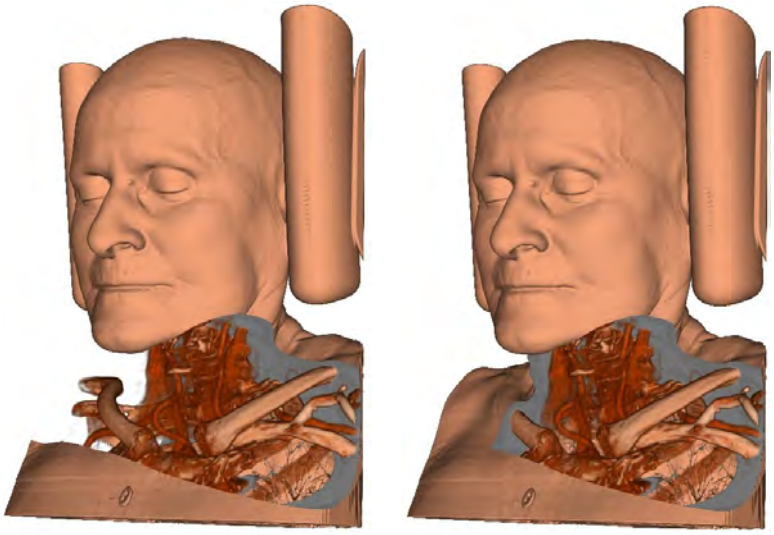


Figure 4.5: Left image shows how the VML looks like when the region of interest has no limit. As it can be seen, the virtual lantern completely pierces the volume dataset. The right image shows the result when an ending cap is added to the VML shape.

4.2.2 Lantern guidance

Regarding the lantern guidance, two methods have been designed depending on the kind of navigation the user is doing: exploring the volume from the exterior or inside it). If the inspection is performed from the exterior, the natural guidance comes from the user's hand. But, if the user is located inside the volume, this may not be the best choice, since it can be difficult to do the lantern points exactly where the user wants. Therefore, an opaque object might occlude the virtual lantern without this being noticeable to the user (see Figure 4.6-left) and also the view frustum may not contain the region of interest defined by the VML. In this situation, a better solution is to change the pointer origin to the observer's position. This is a similar approach to the one taken for ray-based object selection by Andujar *et al.* [AA09]. As a result, the lantern guidance remains intuitive, and the occlusion problem disappears (see Figure 4.6-right).

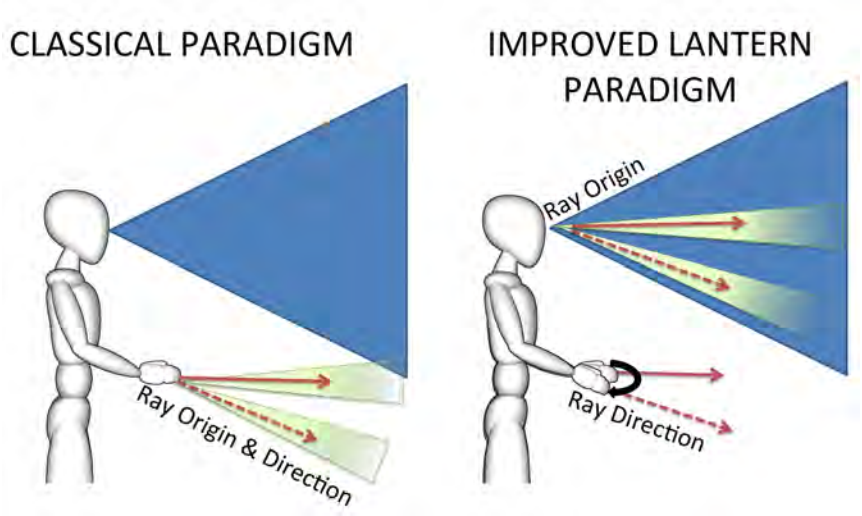
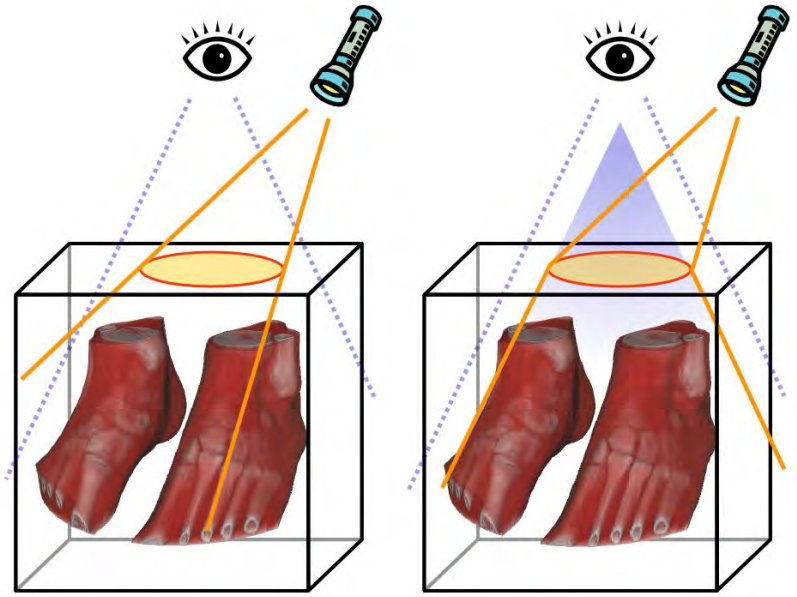


Figure 4.6: Left image shows a user using the original lantern guidance method. In order to avoid that some structure occludes the VML shape or that the view frustum does not contain the lantern frustum, we can make the origin of the VML shape lay in front of the observer, while the hand still guides the direction of the tool. This solves the occlusion problem while maintaining an intuitive pointer manipulation like in [AA09].

4.3 Implementation details

The *Virtual Magic Lantern* and *Virtual Magic Window* approaches are implemented in a similar way. The volume is considered as implicitly divided into two subvolumes, each of which has to be rendered using a different transfer function. Actually, the main difference between both approaches is the way to compute the region of interest. In the VML metaphor, the region of interest is the cone (or other VML shape) created from the virtual lantern that intersects the volume in the direction indicated by the 3D pointer device. In the VMW metaphor, the 3D pointer device creates a virtual cone that intersects

the bounding box of the volume, and the region of interest is defined by all rays that are cast from the observer and traverse the intersection of the virtual cone and the bounding box, in other words, that traverse the located window. These two interaction metaphors are depicted in Figure 4.7. The following sections detail their implementation.



(a) Virtual Magic Lantern (b) Virtual Magic Window

Figure 4.7: The two cone-based focal inspection regions. Left image shows the *Virtual Magic Lantern* metaphor, that generates a virtual cone with origin at the 3D pointer device that determines the region to be inspected with a secondary TF. Right image shows the *Virtual Magic Window* metaphor, that generates a window whose shape is the intersection of the virtual cone with the bounding box of the volume.

4.3.1 VML implementation

In order to obtain a lantern-based inspection, we simply modify the GPU ray casting Algorithm [HKRs+06]. In a very summarized description, our basic ray casting implementation performs three steps:

1. Render the back faces of the bounding box of the model and code the outgoing points of the rays in its color. Store the result as a texture ($Texture_{out}$).
2. Render the front faces of the bounding box and color-code the incoming points of the rays. Store the result as a texture ($Texture_{in}$).
3. Execute a GPU-based volume ray casting using the *in* and *out* points indicated by the previously computed textures.

The implementation of the VML metaphor only requires modifying the third step of the algorithm. We have added to the fragment shader the required code to determine if a point is inside the lantern's shape and also a second transfer function which will be used inside the cone. The application passes to the GPU both the transfer functions and the geometric parameters that defines the cone: apex, axis, and aperture angle. Figure 4.8 shows how the tracing of rays has to be changed in order to take into account the lantern's shape.

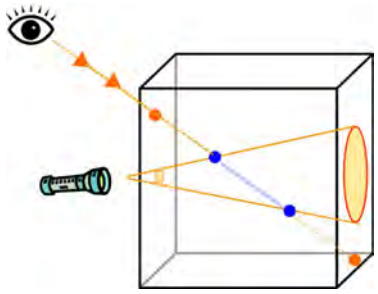


Figure 4.8: VML ray casting scheme. The shading of the samples belonging to the orange segment will be calculated using the main transfer function. For the samples belonging to the blue segment, the second transfer function will be involved.

We have to take special care on the shading of the samples located at the boundary of the interest region. In the classical Phong shading, the gradient influences the final color. As the boundary is a region where the gradient computation could be not robust (see Figure 4.9 left image), we have opted to render the samples at the cone boundary directly with a grey color proportional to the value of the volume dataset at sample's position, without Phong illumination. This approach avoids noisy images and, moreover, shades the boundary regions in a way similar to the gray-level visualization of the medical images, giving an extra cue to the user to easily identify this region, as shown in right image of Figure 4.9.

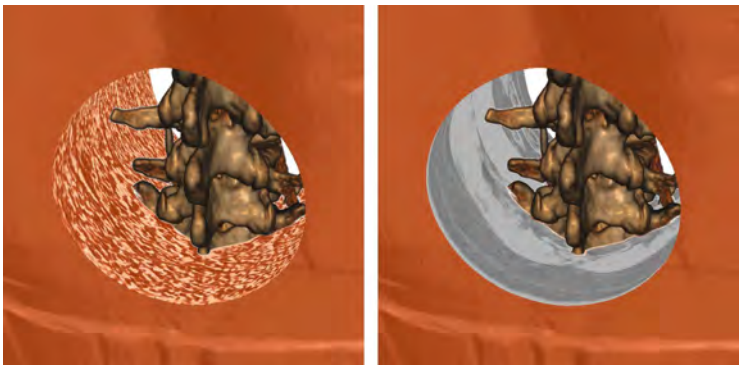


Figure 4.9: The boundary of the VML region is rendered without illumination, in this case, using the gray-level representation of the original medical images to avoid rendering noise.

For illustrative purposes, we sketch the pseudocode of the GPU-based ray-casting implementation in Algorithm 3.

Algorithm 3 Fragment shader pseudocode of the VML metaphor.

```
vec3 position = calculateRayOrigin(gl_FragCoord, Texture_in, Texture_out);
while (! endRaycasting(...)) do
    float v = texture3D(VolumeTexure, position).a;    ▷ Volume texture codified in the alpha channel
    whereis = isInsideVML(position);
    if (whereis == IN) then
        material = getMaterial(interiorTF, v);
        color = calculatePhongShading(material, position);
    else if (whereis == ON) then
        color.rgb = vec3(v,v,v);
        color.a = getMaterial(exteriorTF, v).a;
    else
        material = getMaterial(exteriorTF, v);
        color = calculatePhongShading(material, position);
    end if
    compose colors
    update position
end while
```

4.3.2 VMW implementation

VMW defines the region of interest as the intersection of the front-faces of the bounding box with the lantern's shape (see Figure 4.7). Figure 4.10 summarizes the algorithm developed for the VMW metaphor. Its implementation is straightforward. It has been carried out in two different GPU-based volume ray casting passes. Each of these passes implies modifying the second step of the basic GPU-based ray casting algorithm (see Section 4.3.1). Hence, the algorithm performs the following steps:

1. Render the back faces of the bounding box of the model and encode the outgoing points of the rays in its color. Store the result as a texture ($Back_{VMW}$ texture shown in Figure 4.10).

2. Render the front faces of the bounding box discarding fragments belonging outside the region of interest and color-code the incoming points of the rays ($Front_{InVMW}$ texture shown in Figure 4.10).

3. Execute a GPU-based volume ray casting using the *in* and *out* points indicated by the previously computed textures and the interior transfer function. This generates the IN_{VMW} image shown in Figure 4.10, which corresponds to the rendering of the interior of the region of interest (rays inside the lantern's shape).

4. Render the front faces of the bounding box. Now the discarded fragments are the ones belonging inside the region of interest and color-code the incoming directions of the rays ($Front_{OutVMW}$ texture shown in Figure 4.10).

5. Execute a GPU-based volume ray casting with the new textures that will render the remainder of the volume with the exterior transfer function. This generates the OUT_{VMW} image shown in

Figure 4.10), which corresponds to the rendering of the exterior of the region of interest (rays lying outside it).

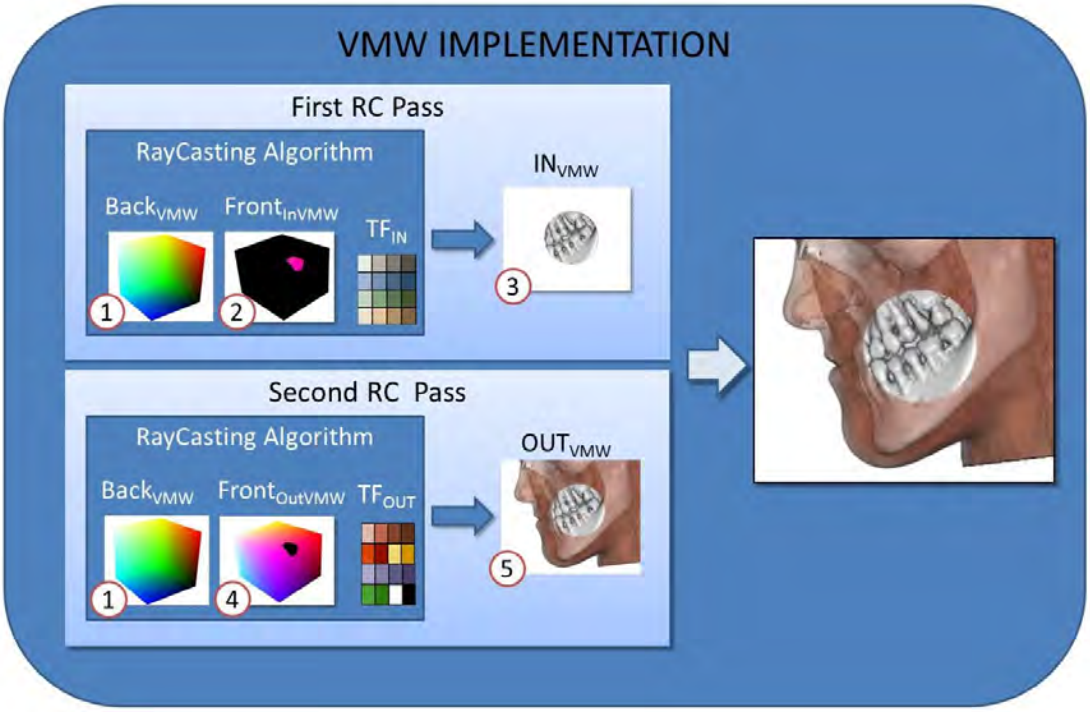


Figure 4.10: Implementation of the VMW metaphor. Two ray casting executions are used to render the inner and outer parts of the region of interest.

Although this implementation requires two GPU passes, it allows the easy adding (without any cost of developing time) of different rendering styles to be incorporated in the VMW metaphor. Note that with our design proposal, the different shaders which implement different rendering styles have not to be modified in order to adapt them to our metaphor. If we had not adopted this solution, for each new render style incorporated to our metaphor we would had to modified the correspondent shader program following the same design scheme we used for the VML metaphor.

Figure 4.11 compares different rendering motifs for visualizing the same region. The top-left image shows a Phong-based shading and the top-right image a Maximum Intensity Projection rendering. The bottom images show the visualization of the beetle model applying a Style Transfer Function (STF) [BG07] as a primary TF (bottom-left) and combined with a classical TF as the secondary one (bottom-right). In Figure 4.12, two rendering styles, Phong-based shading and an ambient occlusion [JPIF10], are compared. We believe our method may help in the transfer function definition process, because it is capable of using two different TFs simultaneously and therefore it provides a powerful yet intuitive way to compare two different transfer functions or advanced shading effects.

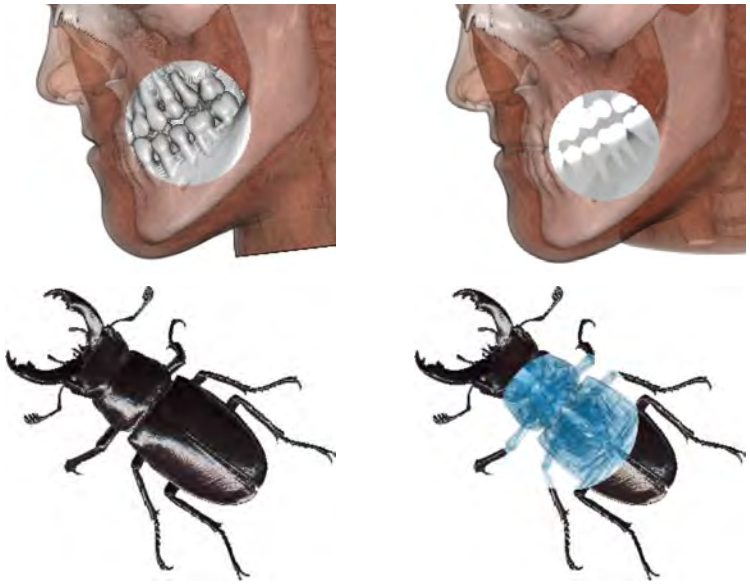


Figure 4.11: Examples of VMW metaphor. First row shows a Phong-based shading (left) and a Maximum Intensity Projection rendering. Second row shows different render styles on the beetle model. On the left, a Style Transfer Function (STF) [BG07] is applied as a primary TF. On the right, the STF has been combined with a classical TF on the region of interest.

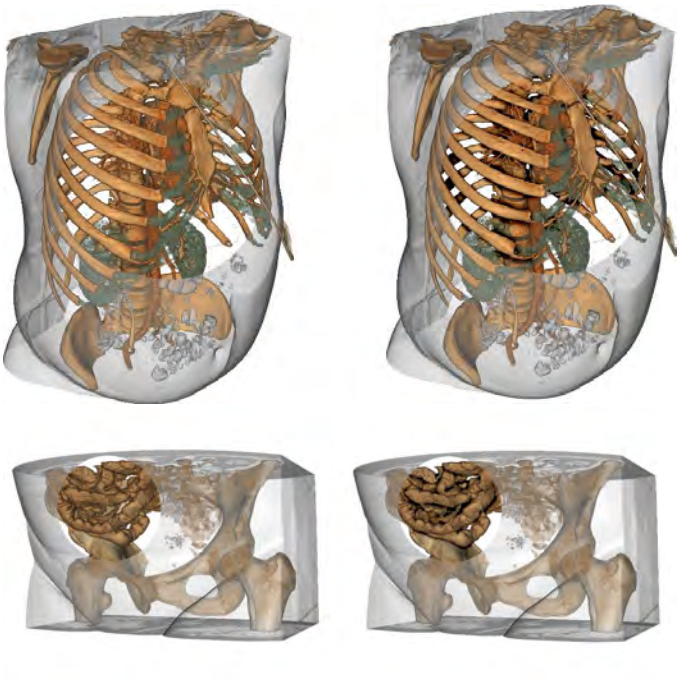


Figure 4.12: Two models rendered without and with ambient occlusion on the region of interest (left and right respectively).

4.4 Results

We have tested our technique with different CT volume models, whose size is up to 512 × 512 × 512 voxels. Timings were taken in a computer equipped with an Intel Core i7-3820 running at 3.60 GHz with 16GB of RAM memory and equipped with a GeForce 590GTX GPU with 1.5GB of RAM memory. The sampling rate was 1 sample per voxel and the size of the viewport was 768 × 768 pixels. Note that the frame-rates shown correspond to the stereoscopic visualization required in Virtual Reality setups.

Table 4.1 shows the complete information of the volume models used. It is also shown an image of the models rendered with the transfer functions used for the exterior (TF_{out}) and the interior region (TF_{in}) of the VML. In order to have a reference, columns $RC_{TF_{out}}$ and $RC_{TF_{in}}$ show the frame-rate obtained for render the model with a classical ray casting by using the TF_{out} and the TF_{in} transfer functions respectively. In this way, we can compare the performance of both techniques against the classical ray casting.

Model	**Resolution**	$\boldsymbol{TF_{out}}$	$\boldsymbol{TF_{in}}$	$\boldsymbol{RC}$	
				$\boldsymbol{RC_{TF_{out}}}$	$\boldsymbol{RC_{TF_{in}}}$
Head	$512^2 \times 485$			50	25
Ribs	512^3			38.95	20.1
Manix	$512^2 \times 460$			70	13

Table 4.1: Performance of the classical ray casting with different data sets using different transfer functions. The frame-rate shown is for the stereoscopic view. The performance is measured in frames per second.

Table 4.2 shows the performance of the VML and VMW techniques in a VR setup for the tested

models, when using the cylinder and the prism shapes (the cylinder and the cone shapes have the same performance). As it can be observed, the VML method shows a performance proportional to the classical ray casting when using TF_{out} and TF_{in}. The VMW technique shows a worse performance than VML, although the obtained results are enough good for being considered available for its use. The penalty in the framerate of VMW could be due to the cost of performing two ray casting passes instead of only one, following the same design scheme than VML. As stated in Section 4.3.2, we adopted this solution due to its simplicity at the time of incorporating different shading styles for being used in the VMW technique.

Model	**Classical RC**		**VML region**	***VMW***		***VML***	
	$RC_{TF_{out}}$	$RC_{TF_{in}}$		**Cylinder**	**Prism**	**Cylinder**	**Prism**
Head	50	25		31	30	45	48
Ribs	38.95	20.1		30	32	44	47
Manix	70	13		29	30	34	39

Table 4.2: Performance of the VMW and VML methods. The frame-rate shown is for the stereoscopic view measured in frames per second. The first column indicates the model used. The second and third columns show the frame-rate of the classical ray casting using the transfer functions specified in Table 4.1. The fourth column shows the VML region used. The rest of the columns show the frame-rate of VML and VMW using the cylinder and prism shapes. The performance is measured in frames per second. Note, how VML and VMW achieves a good performance for being used in a VR setup.

Although VML is better than VMW because it shows a higher amount of context, we still consider the VMW technique a good candidate for visualizing the same model with different rendering techniques.

4.4.1 VML in medicine

As exposed in the sections above, VML is specially useful to provide contextual information of the structures of interest. This is one of the reasons why we consider that the use of VML can be specially interesting in the medical area. To test its utility in the area of medicine, we developed an specific application. It can be seen in Figure 4.13. The application window has three main regions. The *rendering window*, placed in the middle, where the model is rendered. The *panel TFs*, placed on the left, that contains the set of transfer functions which can be used in the region of interest. The *tool box*, placed at the bottom, that contains the remainder of the interaction elements of the application (for instance, the palette icon allows to choose the transfer function used in the exterior of the VML region, and the lantern icon will activate the VML metaphor).

The application works on a 2.7 × 2 meters passive stereo powerwall, and we use an Intersense IS-900 Motion Tracking System device [Int] consisting on a Head Tracker and a MiniTrax Wanda with a joystick as a pointer. The Wanda device is used to track the position and orientation of the lantern, and its joystick is used to change the size of the region of interest and the distance of the cap to the user's hand. Moreover, one of its programmable buttons toggles between the two rendering modes: VML and VMW.

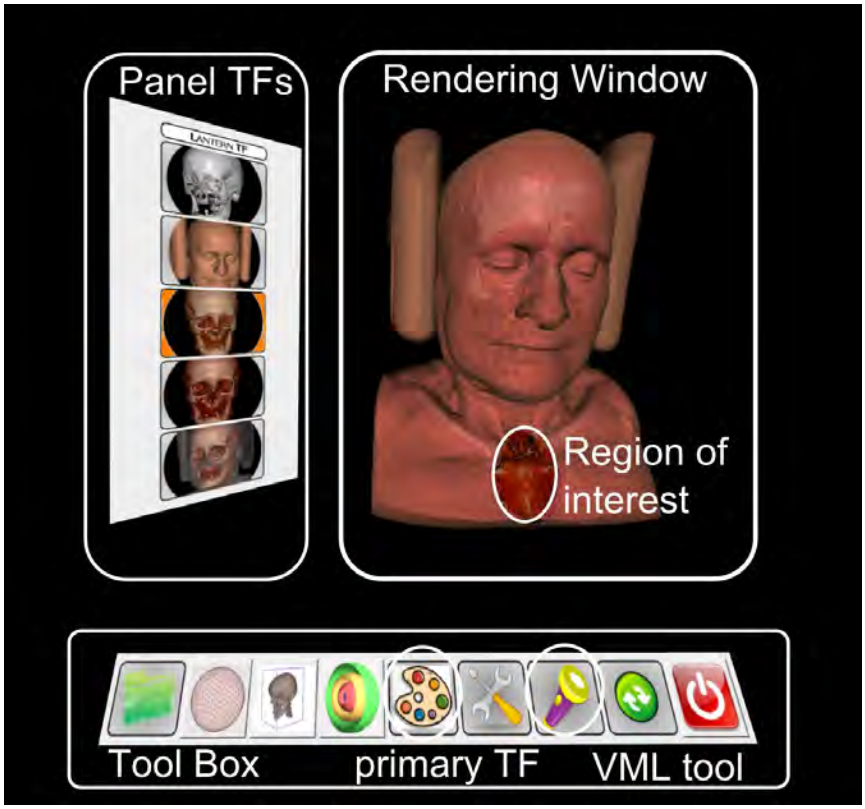


Figure 4.13: Application layout: the rendering window is placed in the center, while the panel that indicates the transfer function used in the VML region is shown on the left side. The bottom widget contains additional tools.

We demonstrated our technique in several interactive sessions with medical doctors. Concretely, in collaboration with two hospitals of Catalonia (Hospital Universitari Arnau de Vilanova and Hospital Universitari Vall d'Hebron) we prepared a showroom where several specialists could test the application (see Figure 4.14).

The main conclusions obtained from these demonstrations were that the technique is very intuitive, behaves in a natural way and it also improves the 3D understanding of medical models. Moreover, we received a number of suggestions for improving the interaction that has already been incorporated (see Section 4.2.1). In addition, some practical applications were pointed out. For example, in the field of surgical pre-operative planning, some medical doctors considered VML useful for surgical procedures that require a trajectory going from the exterior surface to an internal organ. In conventional or robotic laparoscopy surgery, it can be used to localize the most adequate access point for the intervention portals, in order to avoid noble structures and to identify the shortest trajectory. In the area of obstetrics, the inspection of the relative position of the fetus in the final stage of pregnancy was pointed out by the physicians as an example where the use of this metaphor could be very interesting because it can reveal the relative 3D position of the fetus with respect to the body of the pregnant woman.

These showrooms where very encouraging due to the quick understanding of our tool by the medical doctors. They were very excited about the usefulness of this kind of VR applications to the medical context. Moreover, we were very surprised about the swiftness of explaining potential specific medical applications using the VML technique.



Figure 4.14: Showroom presented at the Hospital Universitari Arnau de Vilanova (2010).

4.4.2 User studies

Almost at the same time we published the VML metaphor, another group, leaded by Kirmizibayrak, published a very similar metaphor, called *Magic Lens* [KWBH10], which follows the same objective than the VML metaphor. They performed a user study in order to evaluate the efficiency and usability of their technique. As the specification of the Magic Lens in their experiment was similar to the VML metaphor, we consider that it is reasonable to conclude that we would had obtained comparable results if we had carried out a similar user study. So, we concentrate our efforts to find out the preferences of people of the medical community. Next sections present the user study carried out by Kirmizibayrak *et al.* and our complementary user study.

Initial findings

The content of the Magic Lens (ML) region presented by Kirmizibayrak *et al.* [KWBH10] consisted of the rendering of one of the available data sources (CT images, CFD simulations and real-time video). Their approach was presented as a useful technique to guide surgeons during a laryngoplasty procedure. In a posterior publication [KRW+11], they used mutimodality data previously co-registered and they proposed to interactively edit the way the volume was visualized. In this sense, by using the cylindrical lens (ML) as a volumetric brush, they allow to handle the visualization of arbitrary shaped regions by moving the lens and, optionally, changed the data source visualized in the interior of the ML.

We can see that both methods, ML and VML, are very similar in terms of the pursued objective and also in terms of the proposed solution, although ML achieves more sophisticated results due to their possibility of using it as a volumetric brush.

They performed a user study in order to evaluate the efficiency and usability of their technique. The results of the user study were exposed in [KRW+11, KWY+13]. The study group consisted of 15 people all of them college-educated adults (see [KRW+11] for a deep explanation of the study group characteristics). None of the subjects belonged to the medical domain.

The experiment consisted of localizing artificially created targets inside a volume. Users were asked to explore the datasets to locate these targets as quickly as possible using, for the sake of comparability, both ML and a traditional 2D slice-based interface (see [KRW+11] for a complete description of the test performed and the results obtained). The Magic Lens allowed to explore the inside of the volume datasets (by changing the transfer function used in its interior). After the experiment was performed, participants filled out some questionnaires to evaluate the perceived effectiveness of the approaches ([KWY+13] describes the results). As a summary of their conclusions, most users (11 out of 15) considered their system very easy to use and 12 indicated that this visualization improved their understanding of a volume dataset when compared to 2D slice-based visualizations. With respect to the analysis of interior structures from the outside, 10 out of 15 users found the task very easy or easy to perform.

Findings in the medical domain

As described in 4.4.1, the use of the VML metaphor can be very interesting in the development of new VR applications focused on the medical area. To evaluate the usability and the physicians preferences of the VML metaphor, we have conducted an informal user study. It consisted on performing an exploration of a specific medical dataset (see Figure 4.15) with the objective of analyzing its interior. 7 subjects participated in the evaluation. All of them belong to different medical areas of expertise: radiology, maxillofacial, obstetrics and digestive diseases. Users had to test all the possibilities of our approaches:

- Shape of the lantern: cylinder and prism.
- Finite and infinite version of the lantern shape.
- Lantern guidance: when the user is outside the model, uses the hand to manage it completely and when the user is inside the model, uses the hand to orient it and the head to position it.

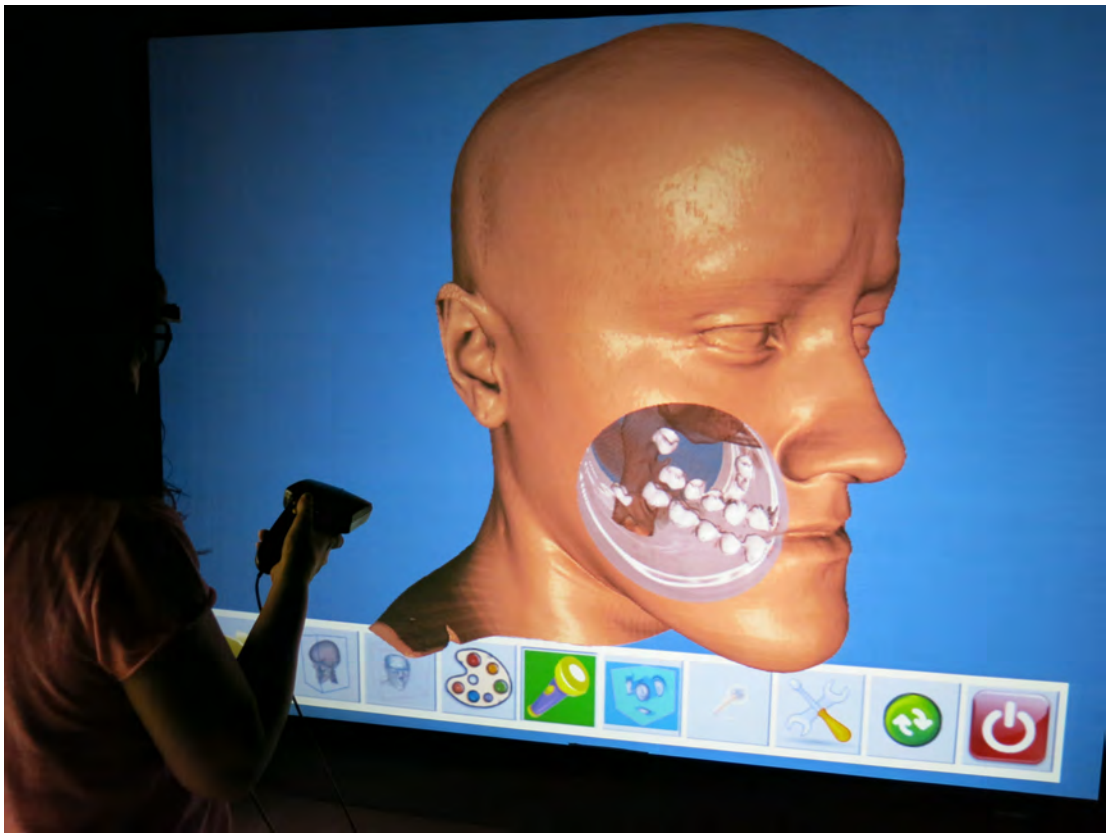


Figure 4.15: Inspection of the model used in the user study.

After testing all the variables, participants filled out a questionnaire (see Table 4.3) and indicated their level of agreement or disagreement with each statement using a 7-point Likert scale, where 1 meant the worst value and 7 was the best value.

Questionnaire	
Q_1	Was the use of the Virtual Magic Lantern easy to learn?
Q_2	Was the use of the Virtual Magic Lantern easy to control?
Q_3	Was the use of the Virtual Magic Lantern comfortable?
Q_4	The hand & head guidance control used when you were inside a volume dataset, was it easy to learn?
Q_5	The hand & head guidance control used when you were inside a volume dataset, was it useful?
Q_6	Do you think these tools would improve your understanding of 3D medical datasets?
Q_7	Did you like the VML metaphor overall?
	Personal Preference and why?
	cylinder | prism
	finite | infinite

Table 4.3: Questionnaire to be filled out by the participants. All the responses were measured on a Likert scale of 1-7, where 1 meant the worst value and 7 was the best value.

The results are shown in Figures 4.16, 4.17 and 4.18 . The answers seem to indicate that VML is very easy to learn and to use. Moreover, all of the participants considered that VML helps understand 3D medical datasets, specially the spatial relationships between different anatomical structures. For instance, the obstetrician stated that the use of VML would be very interesting to show the future obstetricians the spatial relation between the fetus and the coccyx. Moreover, he considered the VML very useful in the area of surgery planning, where the selection of the best path to reach the injury has to take into account noble anatomical structures (nerves, vessels, organs, etc.). The maxillofacial specialist also noted that the VML metaphor would be very useful to have a deeper comprehension of the different entrance choices when planning an endoscopy. Some medical doctors proposed the possibility of freezing the placement of the VML and continue the exploration task with the fixed VML until they pressed a specific button.

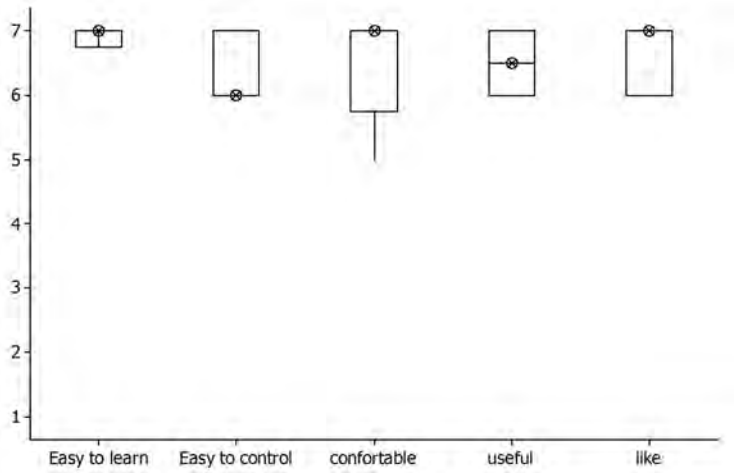


Figure 4.16: Post-questionnaire results from questions Q_1, Q_2, Q_3, Q_6 and Q_7. The boxes show the interquartile range with the median as the horizontal bar. The whiskers extend to the minimum and maximum of the data. These results show that the users' perceptions are quite positive respect the VML technique.

The solution proposed for inspecting volume datasets using the VML when the user is inside the volume dataset was also positively judged. All the participants agreed that using only the hand to control the lantern was not useful enough for the inspection – all of them lost the understanding of where the lantern was pointing to. Questions 4 and 5 revealed whether the participants considered our proposal for inspecting volume datasets from its interior useful. Results are shown in Figure 4.17.

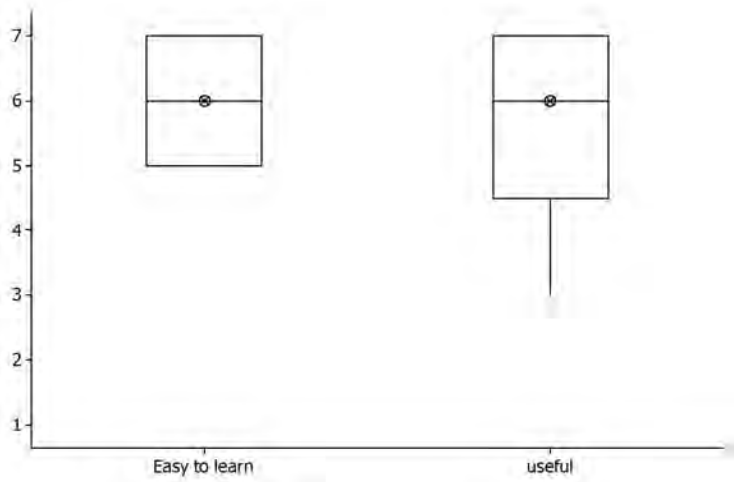


Figure 4.17: Post-questionnaire results from questions Q_5 and Q_6. The boxes show the interquartile range with the median as the horizontal bar. The whiskers extend to the minimum and maximum of the data. These results show that participants didn't have any problem in understanding the hand & head guidance control. Moreover, they considered it useful for the inspection of volume datasets from the interior of them.

With respect to the personal preference about the shape of the VML, the answers revealed that in general the participants preferred the cylinder shape to the prism one. The preference between the use or not of the final cap (finite) was very clear: all of the participants considered the final cap useful.

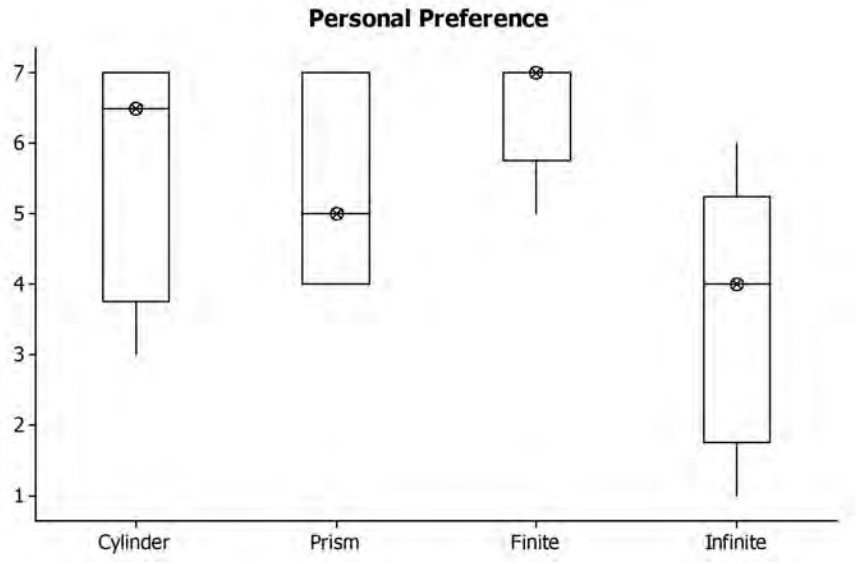


Figure 4.18: Results obtained from a personal preference evaluation questionnaire. The boxes show the interquartile range with the median as the horizontal bar. The whiskers extend to the minimum and maximum of the data. These results show that participants prefer the cylinder shape to the prism one. The preference between the use of the final cap (finite) against not using it is very clear: all of the participants consider more useful the use of the final cap.

4.5 Conclusions

In this chapter we have presented the Virtual Magic Lantern metaphor [MDNV09]. It is a tool tailored to facilitate volumetric data inspection. It behaves like a lantern whose illumination cone determines the region of interest. The lantern is guided by a 3D pointer device that provides the axis direction and the apex position of the VML shape. The region of interest can be rendered using different shading styles that provide a feature rich volume inspection experience. The VML is particularly useful in virtual reality setups with large screens because the interaction becomes very natural and significantly widens the user inspection possibilities. We have shown that the integration of this metaphor into a classic GPU ray casting algorithm can be done seamlessly and runs in real time. We have also shown multiple examples that illustrate the benefits of the VML for improving *focus+context* visualizations.

As a collateral implementation of the VML, we have proposed the Virtual Magic Window metaphor. When using VMW, the user uses the 3D pointer to locate a virtual window to see the model through it. This window is automatically computed as the intersection of the lantern shape with the bounding box of the volume. The VMW also provides an intuitive interaction and its implementation is very simple.

Unfortunately, the VMW does not provide as much contextual information as the VML, especially in the boundary of the region of interest.

The showrooms have demonstrated a very good acceptance of these techniques from the medical community and its potential use in concrete areas of the medical practice. The user study demonstrated that our technique is easy to use and effective. As an outcome, we established the most preferred options by default and left the rest of its configuration available through options.

Depending on the complexity of the volume dataset with respect its size and the transfer function used, its visualization can be costly in terms of frame-rate. The majority of applications in VR setups have a strong component of interaction, so it is mandatory to guarantee at least a frame-rate around of 15 in order to achieve a good application response. Thus, depending on the size of the volume dataset and the transfer function in use, some optimization techniques should be applied in the way of handling with large volume datasets in order to achieve the required frame-rate.

CHAPTER 5

DAAPMED: A DATA-AWARE PICKING TECHNIQUE FOR MEDICAL MODELS

There is a large number of problems where the analysis of medical datasets requires the selection of points in 3D space, such as the measurement of anatomical structures (i.e. lengths of bones) or pathological structures (i.e. tumors). Depending on the area of applicability, it is possible to develop semi- or fully automatic measurement tools in order to assist the physicians (for instance, the computation of the minimal distance between two anatomical structures). But this automation always requires an explicit segmentation preprocess which can be sometimes impossible, for instance, due to physicians' time restrictions. In parallel, with the success and popularity of Virtual Reality (VR) environments, researchers are more and more interested in the development of interaction metaphors that may take advantage of the 3D environment.

Although many researchers have investigated 3D object selection techniques for general -non medical- VR applications, less research have been devoted to the specific area of medicine. Our main motivation has been to improve the selection of 3D points in non-segmented volume datasets rendered with a *Direct Volume Rendering* (DVR) algorithm. In this specific context, the visualization may produce images in which the structures can be visualized with semi-transparencies, providing a means to increase the amount of information visible to the users, and facilitating the establishment of spatial relationships between them. However, determining what the user wants to pick or select may be ambiguous depending on the kind of structures she is working with. Nowadays, helping the user in the task of selecting or picking objects in VR-setups is still an open problem.

In this chapter we propose a new selection metaphor supported with some visual cues for the efficient, accurate anchor point selection in non necessarily segmented volume datasets rendered using DVR in VR. It is important to note that we are not interested in selecting a concrete structure, but a

point on it, without any previous surface extraction nor segmentation process.

The remainder of the chapter is organized as follows. Section 5.1 summarizes the related work on the inspection of volume datasets and on picking structures or points on them in VR setups. Section 5.2 outlines the proposed technique. Section 5.3 describes the implementation of its main components in depth. Section 5.4 details the user study we carried out for its evaluation. Finally, conclusions are presented in Section 5.5.

5.1 Related work

In this section we summarize the most relevant works related to the problem addressed in this chapter: the selection of anchor points in volume datasets in VR setups.

In a pioneering work, Hinckley *et al.* proposed a 3D user interface for pre-operative neurosurgical planning based on the physical manipulation of familiar real-world objects (head model, cutting-plane and stylus-shaped props) to access and manipulate a virtual model [HPG94, GHP+95]. This approach offers the possibility to select anchor points in a brain model consisting of a polygonal mesh. They used a clipping plane to access occluded or interior points in the brain and then select anchor points on it as the intersection of the linear trajectory defined by the stylus and the cutting-plane. If the clipping plane was not activated, the user was able to select points lying on the brain model as the intersection of the ray cast from the stylus and the brain model's surface. Following the research carried out on the design of 3D tangible user interfaces, Qi and Martens [QM05] presented three different designs of a clipping-plane interface with 2D and/or 3D interaction devices for a small size VR system (based on a 14" display). Their informal evaluation showed that most users perceive the tangible interface as being much easier to use than a traditional 2D interface, although their system did not provide any selection mechanism. More recently, Song *et al.* [SGF+11] proposed the use of a touch mobile for manipulating (positioning and orienting) a slicing plane. Since their objective was only the exploration, they did not address the problem of anchor point selection.

While trying to automate the process of taking measurements between anatomical structures, Preim *et al.* [PTSP02] introduced a set of applicable tools for the computation of distances, angles, and volumes in 3D visualizations. These tools are 3D virtual objects, such as a distance line, a ruler, and angular measurements that are manipulated using the mouse in a desktop-platform. They allow to position these tools on the surface of the pre-segmented anatomical structure. Following the same line, Rossling *et al.* [RCD+10] proposed a method for the automatic determination of different distance−based measures (shortest distance, diameters and wall thickness) also on segmented anatomic structures. The necessity of this kind of tool is justified by the fact that manual distance calculation is tedious and imprecise in single 2D slices, and although it is possible to achieve an accurate result in 3D, it would also be tiresome. However, completely automatic measurements are difficult to generalize due to the great variety of problems and anatomical structures. Notice that both previous approaches [PTSP02, RCD+10] work on triangle mesh representations, so a surface extraction process is needed before using them. Moreover, they always select the nearest visible point on the surface and

they do not deal with semi-transparent models. Both systems work on a standard desktop workstation using the mouse.

Following the same line but in the context of VR, Reitinger *et al.* [RSBB06] presented a 3D measurement toolkit developed for liver surgery. Their measurements include distance between points and structures, volumes, and angles. Their evaluation indicated that VR-based measurement tools have a sufficient benefit compared to 2D desktop-based systems in terms of task completion time. In terms of accuracy, slightly better results in most of the tasks were achieved. The anatomical structures models (liver, vessels,...) are computed through segmentation from CT scans and they are represented by opaque triangle meshes where the user may select points by using a virtual pencil. Hagerdorn *et al.* [HJDP07] proposed a set of tools for performing measurements in a VR visualization environment. A 3D Rubberbanding line for selecting free points in the scene is proposed. They use clipping planes for accessing interior parts of the volume dataset. Their scene is also composed by triangle meshes.

Segmentation and surface extraction are time consuming operations. To overcome this problem, Hastreiter *et al.* [HTEE98] suggested the use of DVR of the entire data volume, giving insights to interior and facilitating the establishment of spatial relationships between the different elements. In order to inspect interior structures, independent clipping planes provide an intuitive way to virtually cut off parts of the volume data set. Then, anchor points can be interactively placed on the clipping planes.

Gobbetti *et al.* [GPZT98] introduced in the area of *Volume Graphics* a raycasting-based selection technique. It consisted of searching along the cast ray using the usual color front-to-back compositing scheme in DVR. The ray stopped when the accumulated opacity exceeded a user defined threshold. This technique assumes that surfaces are at locations where opacity exceeds a given threshold. For very transparent surfaces the assumption may not hold because some very transparent structures could be missed and, on the other hand, large "foggy" areas could be picked if the threshold is reached. Although the defined threshold can be changed by the user, it may be difficult to fine tune it, since the reached opacity may be view dependent. Gallo *et al.* [GDPM08] present a VR system for the exploration of volume datasets using a *Wiimote*. Apart from the basic interaction techniques for navigating, they propose a mechanism for point selection based on the classical ray-casting technique adding the mechanism of *fishing reel* in which the users can move the cursor closer or farther away by using two buttons in order to accurately locate a mark. Unfortunately, point positions are not aware of the isosurfaces and no visual cue is used to reveal the cursor when it is moved into an occluded region.

In a recent work, a new approach has been proposed with the objective of overcoming the problem of using opacity as the only term involved in the equation of getting a 3D point from the 2D mouse position of the user. Wiebel *et al.* [WVFH12] presented a new volume picking technique called *WYSIWYP* ("What You See Is What You Pick") based on the volume dataset and the transfer function used on their volume rendering algorithm. Their technique focus on selecting the sample which more contributes to the final pixel. It is based on the assumption that high opacity is usually assigned to important features. Depending of the purpose of the visualization, some relevant structures, such as the skin, can be highly transparent in order to see the other structures inside it (the skull, for example). This technique is not

suitable for the selection of anchor points on the highly transparent structures, for example. Moreover, as shown in a posterior work [WPVH13], the selection may be unstable, and therefore it makes it difficult to select less visible points.

As mentioned before, 3D object selection in VR environments has been mainly addressed for general polygonal scenarios [BKLP04]. *Ray-based* techniques [Min95] have shown a better performance than *point-based* techniques [PBWI96, VGC07]. The former approaches are usually based on a cone or a ray. Since our interest is on accurate anchor point selection, we only consider ray-based tools. In order to solve the inherent problem of multiple intersection candidates, several disambiguation techniques have been proposed. Olwal *et al.* [OF03] proposed the *flexible pointer*, a ray cursor technique that allows users to point around objects with a curved arrow, to select fully or partially obscured objects. It is important to note that most of these VR selection metaphors are focused on selection and manipulation of objects (not points) in populated scenarios, and thus they are not specially concerned about accuracy in single point selection.

Grossman *et al.* [GB06] explored 3D selection techniques for volumetric displays and proposed four new ray cursor techniques which provide disambiguation mechanisms for multiple intersected targets. The *Depth Ray* tool augments the ray cursor with a depth marker. The position of this marker is changed dynamically moving the hand forward and backward. As the hand also controls the position and orientation of the ray cursor, the two phases could potentially interfere with each other. To solve this problem, they propose the *Lock Ray*, a similar technique, where selection and disambiguation phases are carried out sequentially, in a two-step process. First, the user selects the ray. Once it is locked, the depth marker appears. Then, forward and backward hand movements fix the depth marker, and the intersected target closest to it is highlighted in red indicating that it can be selected by releasing the button.

After the analysis of the related work, we can conclude that there is still a lack of tools for measurement support for medical models in VR environments. The majority of the techniques that try to improve the user experience in the selection task take advantage of some kind of semantic information. In order to obtain this semantic information some, usually costly, segmentation preprocess has to be applied to the raw medical datasets. It is necessary to continue developing new techniques to overcome the limitations of these methods: limitations due to the appearance of the involved anatomical structures (the use of transparency) or due to a complex data preprocessing (the necessity of an isosurface extraction process of the involved structures).

5.2 A data-aware anchor point selection for medical models

As stated above, our main motivation is to help users in the task of 3D point selection in volume datasets rendered using DVR. We propose a new VR-based interaction technique, named DAAPMed:*Data-Aware Anchor Points for Medical models*. This technique is specially focused on the fast and accurate selection of 3D points on implicitly defined surfaces of anatomical structures present in volume datasets rendered using methods which allows semi-transparency in a virtual environment.

The DAAPMed selection technique has three main components shown in Figure 5.1.c :

- **Ray cursor tool**: It casts a pointing ray through the volume. The ray path visualization is enriched with the candidate selection points and two supporting planes, which provide a better insight of its position and orientation. The candidate points to be selected are automatically calculated by detecting isosurface intersections with the ray.
- **Helper Views**: We provide two views which help the user to understand the position of the ray inside the volume. This extra-visualization is inspired by the Magic Mirrors View [KDGB99], but, instead of showing the whole model, each view shows the model clipped by one of the supporting planes that enables the possibility to show the ray trajectory without any occlusion.
- **Disambiguation mechanism**: Once the ray is locked, it allows the user to select among the different intersections of the ray with the isosurfaces in the model. We adopt the same solution as Hinckley in [HPGK94], cycling from one target to the next.

The DAAPMed metaphor works as follows. Initially, the user may explore the volume dataset interactively to set the best view for the next selection task (see Figure 5.1.a). When the user presses the corresponding button of the input device, the *selection task* starts (see Figure 5.1.b) and the ray is painted with a gradient color from red to yellow (in this way we provide users with a visual cue of the depth of the ray). Throughout this process (while the user is pressing the button), the system continuously computes and visualizes the proper set of candidate points. This set is composed by all the intersections of the ray with the implicitly defined isosurfaces. As the 3D ray is painted over the volume, it is sometimes difficult to interpret how the volume is traversed. In order to give the user a second cue on the intersection of the ray with the volume, we provide the *Helper Views* (see Figure 5.1.c). This visualization has a main advantage: it shows all the candidate points that lie inside the volume. In this way, this visual feedback facilitates the ray selection without any previous manipulation of the volume (i. e. clipping) and disoccluding inner intersection points. The visualization of the volume model is augmented with a wireframe representation of the cutting planes (also named supporting planes) used in the Helper Views in order to provide the users with a visual feedback of the placement of such planes.

Upon button release, the last ray shown is locked, meaning that the selection phase has finished and the disambiguation task begins (see Figure 5.1.d). The nearest candidate point is marked in orange (default selection) and the rest of the points are in white. The joystick provided by the input device allows the user to cycle among all the candidate points. This is convenient because it reduces movements. Pressing a specific button, the point marked in orange will be selected as the anchor point used in the current measurement task the user was involved. Figure 5.2 shows a user interacting with a head model.

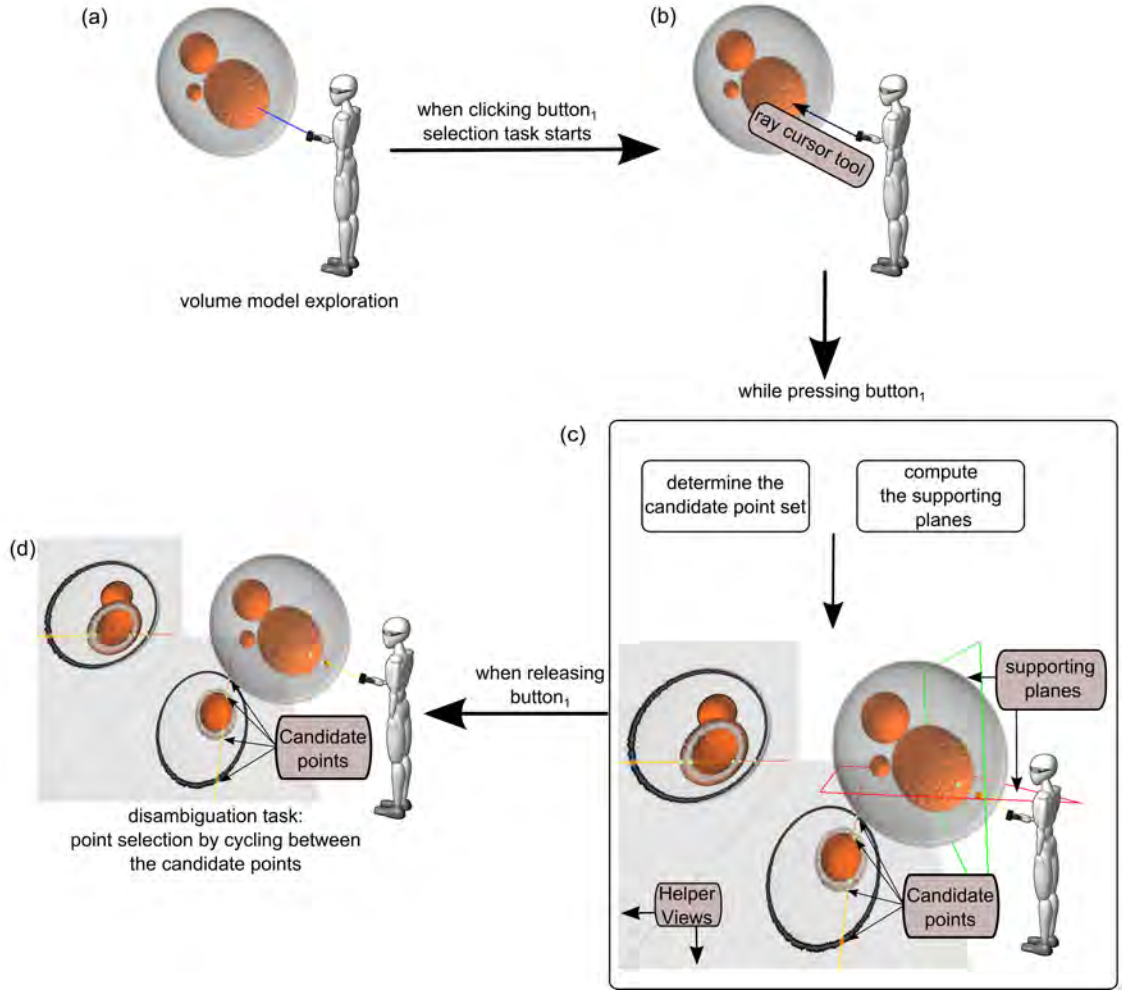


Figure 5.1: Block diagram illustrating the workflow of the DAAPMed technique. When the user clicks button$_1$ of the input device, the selection task starts. While the user is pressing the button, the system calculates the proper set of candidate points and visualize all the components of the DAAPMed technique (Helper Views, the supporting planes, and the candidate points). When user releases the button, the ray is frozen and the user can select the point she is interested in by cycling among them with the help of a joystick provided by the input device.

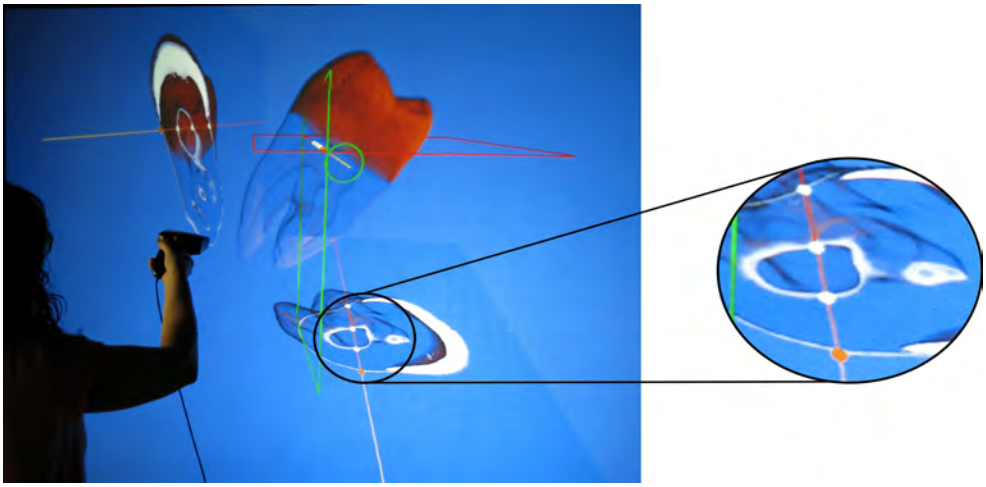


Figure 5.2: User interacting with a model using DAAPMed. The blow-up in the figure magnifies the visualization of the candidate selection points in one of the Helper Views.

5.2.1 Heisenberg effect

It is well known that, when working with a tracked device, a discrete input (e.g. button press) will often disturb the position of the tracker. This phenomenon, called Heisenberg effect of spatial interaction, has to be taken into account when developing new interaction techniques [BWC+02]. Otherwise, the accuracy of the selection may be affected due to changes in the holding forces done by the user when pressing or releasing a button. To overcome this problem, we enhanced the visualization of the ray with a freezing timer. Figure 5.3 illustrates the use of this add-in.

The mechanism works as follows. When the user presses the button in the selection task, a circle centered around the ray begins to be drawn and the current ray is saved to be used when user releases the button. While the movement done by the user's hand is smaller than a certain tolerance, the circle continues being drawn. If the movement exceeds the tolerance, the part of the circle that has been drawn is erased and the process starts again. When the circle is completely drawn, the user can release the button without worrying about losing the ray she is seeing. The establishment of the tolerance also prevents any involuntary movement (little shaking) caused by the holding of the input device for a while. The completion time for drawing the complete circle is around 2 seconds.

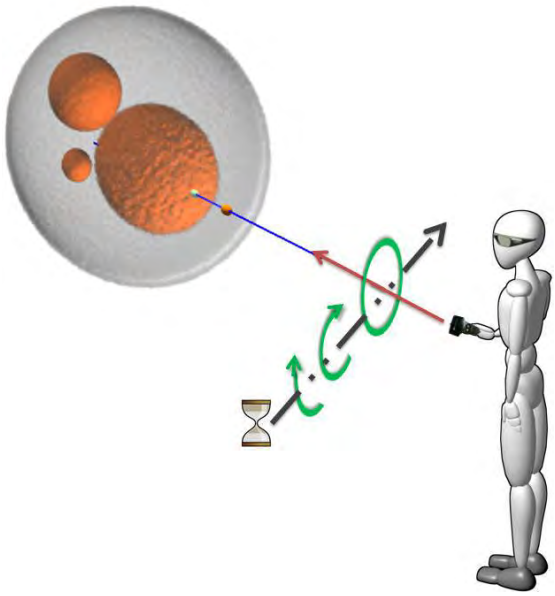


Figure 5.3: Illustration of the add-in incorporated to prevent Heisenberg effect of spatial interaction. The visualization of the ray is enhanced with a freezing timer.

5.3 Implementation details

In this section we detail the on-the-fly automatic detection of the isosurfaces along the pointing ray as well as on how the Helper Views are created. Since we want to work with a non-segmented model, these isosurfaces must be determined in real-time, as they depend on the transfer function. Throughout all the process we use a DVR method based on GPU ray casting.

5.3.1 Automatic detection of the ray-isosurface intersection

A critical aspect of our interaction metaphor is the detection of the isosurfaces traversed by the ray in order to set the candidate points. We accomplished it in the following way.

Volumetric models can be seen as a 3D scalar function $f : V \subseteq \Re^3 \rightarrow \Re$ (e.g. density value of a material). Let $TF : \Re \rightarrow \Re^4$ be the transfer function that assigns color and opacity to a scalar property. First of all, we have to define the conditions that a point p sampled inside the volume dataset V must fulfill to be considered a boundary-surface candidate point. These conditions are:

1. p must belong to a visible (non-transparent) material. It can be expressed as

$$opacity(TF(f(p))) > 0.0$$

2. p must belong to the boundary of a well-defined isosurface. While ideal boundaries have a sudden change in the 3D scalar function, boundaries in medical images are smoothed due to the image acquisition process. In most cases, the boundary can be identified [KD98] by analysing

mathematically the 3D scalar function f. The conditions a point p belonging to the boundary of an isosurface satisfies are: f' reaches a local maximum at p and f'' is zero at p. We can express these conditions less formally as:

a) The gradient at point p, $\nabla f(p)$, has to be well defined. This means that $\|\nabla f(p)\|$ is larger than a certain threshold. Accomplishing this condition is a weak calculation of f' without taking into account the neighborhood.

b) There exists a sudden change in the gradient around a neighborhood of p. This property expresses the fact that the boundary passes through p.

Since the detection of the *2.b* condition may not guarantee interactive times in a VR environment, it was decided to precompute the information necessary to test it. This is carried out by applying a 3D edge detection process [MDMC90] to the volume V and storing the result in a 3D dataset which consists of a value per voxel that indicates the possibility of being crossed by the boundary of a surface. As the used 3D edge filter returns scalar information (normalized to the range 0.0 to 1.0) about the magnitude of the edges, we only use the high values (values superior to 0.5), in order to get rid of false edge points which are falsely detected due to the presence of noise in the volume dataset. Then, the condition *2.b* is tested by checking whether p belongs to a boundary voxel (isAnEdge(p)).

Therefore, the algorithm to compute automatically the points located at the boundary of the isosurfaces traversed by the ray consists of evaluating the expression:

$$opacity(TF(f(p))) > 0.0 \,\&\, isAnEdge(p)$$

Since our technique takes into account the information of the 3D edge detection filter in conjunction with the opacity assigned by the transfer function, we can avoid the detection of a isosurface when the ray passes across foggy regions while we can detect very transparent well-defined isosurfaces.

Figure 5.4 shows the candidate points detected along a ray cast through a volume dataset using our solution and also the different conditions exposed above evaluated individually (the opacity, the gradient magnitude and the edge detection filter). As it is shown, our approach detects all the visible well-defined isosurfaces traversed by the ray. Note that if we would analyze the conditions individually, we would mislead the detection of the well-defined isosurfaces contained in the volume dataset depending on the threshold established in the evaluation of the condition.

The implemented algorithm guarantees testing at least a point for each voxel intersected by the ray, thus, the accuracy of our approach is related to voxel's size. As shown in Section 5.4.3, we obtain an accuracy comparable to that of both a clipping plane selection approach and to a desktop application that works with a triangle mesh model (not a volume model) for the anatomical structures. This is due to the fact that surface extraction methods also have an accuracy proportional to the voxel size. The computation of the 3D edge detection filter is comparable to the model loading time. So, the preprocess performed is acceptable in terms of time. Notice that the edge detection filter works with

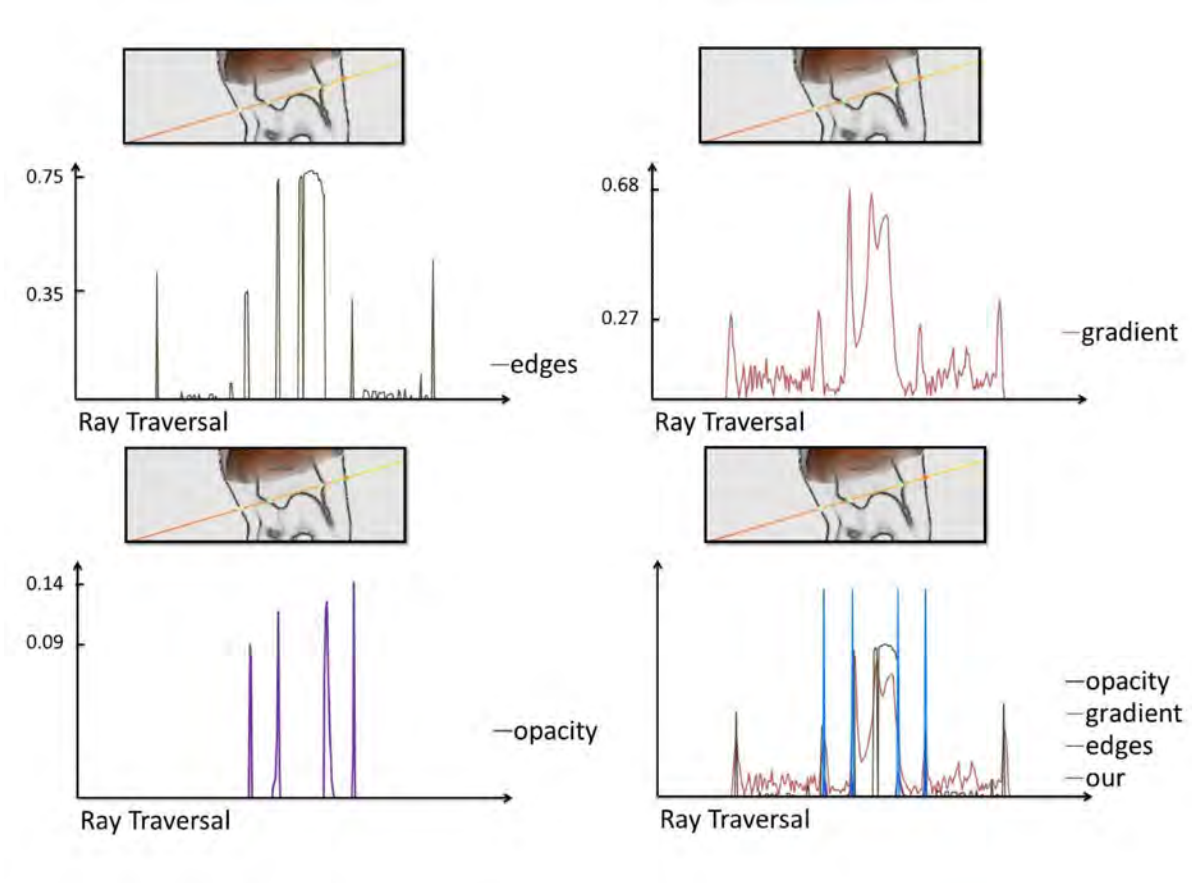


Figure 5.4: Plots showing all the components involved in the computation of the candidate points along a ray cast through a volume dataset. Notice that the points detected correspond to peaks in our isosurface detection measure.

the raw volume data, so it has to be computed only once, even if the user changes the current transfer function.

5.3.2 Helper Views: Visual feedback framework

The goal of Helper Views is to provide additional information on the exact position of the ray inside the volume. These views are drawn on two planes, located at fixed positions with respect to the reference coordinates system of the virtual world (planes YZ and XZ) (as shown in Figure 5.5).

Images displayed on each of these planes are generated with the same DVR algorithm used for rendering the volume model, but in this case, the volume is clipped by the plane that contains the ray and is the most parallel to the image planes YZ and XZ, respectively.

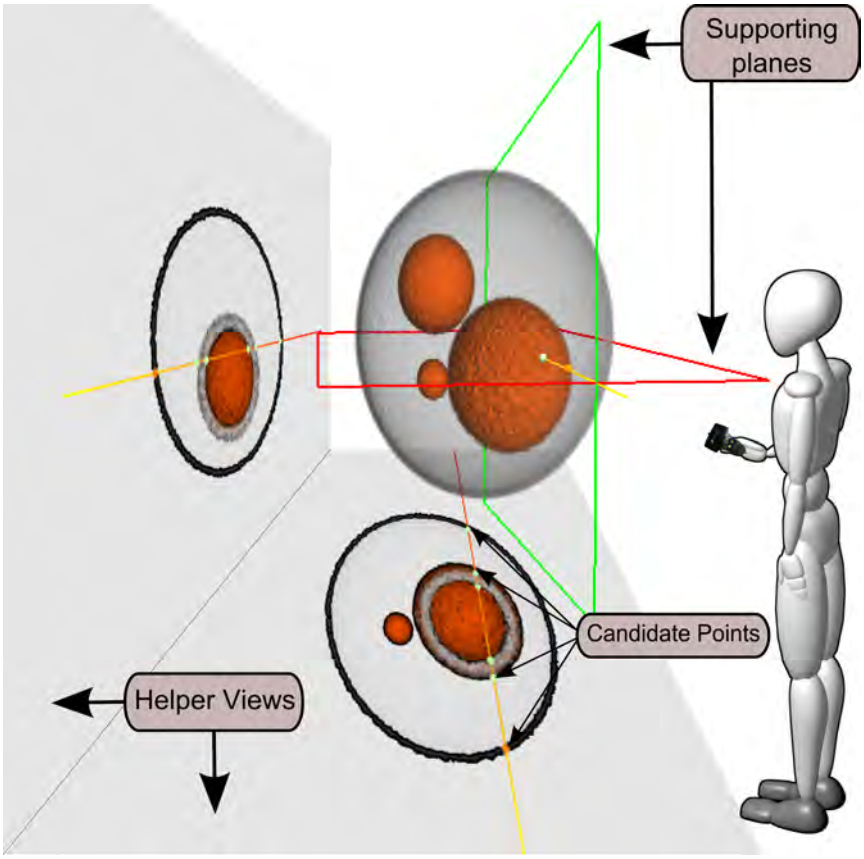


Figure 5.5: Helper Views consist of two planes which help to understand the position of the ray inside the volume. These views show the model clipped by a plane that aids disoccluding interior candidate points. The volume dataset consists of four spheres of different materials. Notice that the bottom view allows as to see that the large orange sphere is hollow.

Being R the ray defined by the user, each clipping plane, $\pi_{clipping}$, satisfies that:

1. R is contained in the clipping plane $\pi_{clipping}$.

2. $N_{\pi_{clipping}} = \vec{v} \times \{\vec{X} | \vec{Y}\}$, where $\vec{v}$ is the support vector of the ray and $\vec{X}$ and $\vec{Y}$ are the normal vector of the corresponding YZ or XZ plane, respectively.

This visualization has a main advantage: it shows all the candidate points that lie inside the volume. In this way, this visualization facilitates the ray selection without previous manipulation of the volume (i. e. clipping) and disoccluding inner intersection points.

The implementation of the Helper Views is simple. It consists in rendering a polygon located at YZ or XZ, which is textured with the result of the visualization of the volume dataset clipped by the corresponding clipping plane. The resolution of the texture affects both the quality of the result and the performance. We have to find a compromise between these two factors in order to guarantee that

DAAPMed technique is feasible in VR setups. Next section details the performance of the overall technique).

5.3.3 Performance

The proposed method was tested in an immersive virtual reality setup composed of a 2.7 × 2 meters passive stereo PowerWall (see Figure 5.6). All timings were computed in a window size of 768 × 768 pixels. The rendering hardware is a 3.60 GHz Intel Core i7-3820 with 16GB of RAM memory and equipped with a GeForce 590GTX graphics card with 1.5GB of RAM memory.

Table 5.1 shows the performance of the DAAPMed method. First column shows the name used to reference the model, its resolution, and the voxel dimensions (in mm). Second column shows a capture of the rendering of the volume model. Third column (*DVR*) shows the frame-rate achieved by the DVR algorithm when the user is exploring it. Fourth column (*DVR plus Points*) shows the frame-rate achieved with the candidate points detection is activated. The rest of the columns show the total time including the visualization of the Helper Views using different viewport sizes.

Analyzing the performance, we can see that DAAPMed takes 3 times more than the isolated visualization of the volume dataset. Note that the computation of the candidate points set is not significant in terms of performance. The overall performance strongly depends on the visualization of the volume dataset, and this depends on the size of the volume dataset and also on the design of the transfer function. It has to be taken into account that the volume dataset has to be rendered three times for each frame: the visualization of the volume dataset plus the visualization of the Helper Views. In addition, the visualization of the volume dataset plus the visualization of the Helper Views have to be rendered twice: once for each viewpoint of the stereoscopic system. So, in total, we have to render the volume dataset six times each frame (although the viewport used for each volume rendering has a different size). So, depending on the size of the volume dataset and the transfer function in use, it could be necessary to develop some optimizations in order to guarantee the minimum frames per second required in a VR environment. For the models tested, we use viewports of 512 × 512 for the Helper Views since it is enough for having a good quality image on them.

The time needed for the calculation of the edge detector goes from 0.52s for the smallest model to 54.2s for the biggest model, which is an acceptable time. Moreover, due to the fact that its calculation only depends on the raw data of the volume model, it can be precomputed for each volume dataset and be loaded together when the volume dataset is loaded. The loading time goes from 0.18s for the smallest model to 23.37s for the largest one.

Model	Transfer Function	DVR	DVR plus Points	HelperView Viewport Size		
				256×256	512×512	768×768
Spheres $128 \times 128 \times 128$ $1.0 \times 1.0 \times 1.0$mm.		44	44	37	30	24
Tooth $256 \times 256 \times 256$ $1.0 \times 1.0 \times 1.0$mm.		38	38	26	19	12
Head $512 \times 512 \times 485$ $0.51 \times 0.51 \times 0.5$mm.		25	25	16	11	8
Skeleton $512 \times 512 \times 369$ $0.92 \times 0.92 \times 1.6$mm.		23	23	17	13	9

Table 5.1: Performance of the DAAPMed technique. The frame-rate shown is for the stereoscopic view. The performance is measured in frames per second with different viewport sizes for the Helper Views. The first column shows the information of the volume dataset. The second column shows a capture of the DVR rendering. The third column shows the frame-rate of the DVR algorithm. Fourth column shows the frame-rate achieved once the computation of the candidate point set has been added. The rest of the columns show the achieved frame-rate once the visualization of the Helper Views has been added using a texture of the resolution indicated in the column. For the models tested, a viewport of 512×512 for the Helper Views show good quality and they do not penalize framerate highly enough, so it is the resolution we used for these elements.

5.4 User study

We have conducted a formal user study to evaluate the accuracy, efficiency and ease of use of our approach. We take as a reference an implementation of the Clipping Plane (CP) selection method, since it is a widely used technique in medical applications (see Section 5.4.1). This technique consists on first positioning a clipping plane inside the volume dataset and then picking a point contained on it (see Figure 5.7).

The user study has been performed in an immersive virtual reality setup composed of a 2.7 × 2

meters passive stereo PowerWall (see Figure 5.6). Users were tracked using an Intersense IS-900 Motion Tracking System device consisting on a Head Tracker and a MiniTrax Wanda with a joystick and five programmable buttons.

The results show that users required a significant smaller amount of movement with DAAPMed than with CP and that the selection performed is more accurate with DAAPMed than with the CP technique.



Figure 5.6: Immersive virtual reality setup used for the user study.

5.4.1 Design details of the Clipping Plane technique in Virtual Reality

In order to compare the DAAPMed selection technique with the classical approach using a clipping plane (CP) for anchor point selection, we ported this metaphor to a VR setup in the following way. Using the buttons of the input device, the user sets the action to be performed: rotating or translating the clipping plane. While the user is pressing the corresponding button, the clipping plane is rotated or translated according to the user's hand movement. The rotation is based in the paradigm of the *Rolling Ball* [Kir92]. The translation is always done in the direction of the plane's normal. Once the plane is fixed, the user can select a point on it using the ray-cursor paradigm. While the user is pressing another button, the intersection of the ray with the plane is computed and the intersection point is visualized – this action is enhanced with the Heisenberg reduction add-in (see Section 5.2.1). In this way, every point inside the volume belonging to the plane, can be a candidate point to be selected.

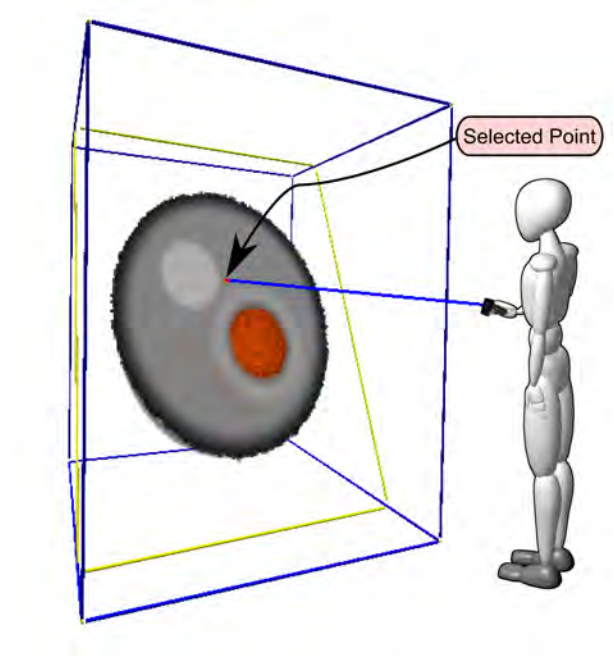


Figure 5.7: Porting the clipping plane technique for point selection to VR.

5.4.2 Test design

Medical doctors often address two different point selection problems: selection of well-established anatomical points, and distance measurement between arbitrary (or non-arbitrary) points.

As a consequence, we decided to test three different tasks: the selection of individual marked points (T_1), the measurement of distances (T_2) and the selection of specific anatomical points in a volume dataset (T_3). The processes of each task were defined as:

- In T_1 task, users had to introduce two anchor points (P_1 and P_2) at positions which were marked in the model with a cone (see Figure 5.8).
- In T_2 task, it was required to calculate the distance between two points (see Figure 5.11).
- In T_3 task, users had to locate, as accurately as possible, a set of anatomical points indicated on a reference image, shown at the bottom left corner of the screen (see Figure 5.9).

The experiments were performed in two sessions. One session consisted of tasks T_1 and T_2 – we called this session $Test_1$. The other session, called $Test_2$, was conformed by task T_3.

In these experiments we wanted to evaluate the efficiency and the accuracy of the DAAPMed technique with respect the CP technique. So, throughout the tests several magnitudes were measured that would provide information on the amount of displacement (and thus, effort) required by each technique.

We recorded the following indicators for each task (when the full name did not fit in the tables that summarize the results, we show in italics how the indicators are used there):

- Task completion time (*Time*): It measures the amount of time devoted to complete each task.
- Input Device Footprint (*Device Mov.*): It measures the length of the total path followed by the device to complete each task.
- User footprint (*User Mov.*): It measures the user displacement inside the VR environment done while carrying out a task.
- Accuracy: This value measures the error in the selection with respect to the reference points, taking into account the size of the voxel as a metric of the error made.

Data preparation

We prepared two different datasets for $Test_1$. The first one was used for training, while the other was used for the test. The training model consisted of a set of four spheres of different materials (Figure 5.8-left). The second model consisted of a tooth, a typical CT dataset in volume visualization, using a transfer function which shows the outside and the inner shape of it (Figure 5.8-right). The anchor points used in task T_1 for the tooth model, include both external and internal characteristics of the model (see Figure 5.8).

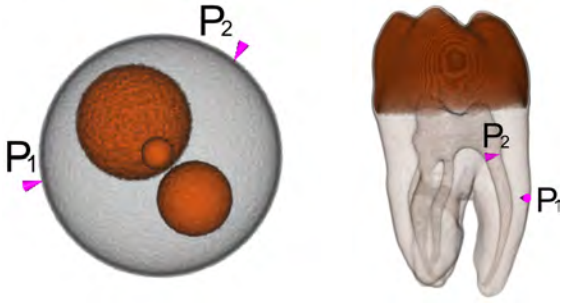


Figure 5.8: The training (left) and testing (right) datasets used in $Test_1$. These two models were obtained from *The Volume Library* repository [Roe]. The figures show the anchor points to be selected in T_1 task. We include points in very semi-transparent structures, which are difficult to select with other techniques.

The model used in $Test_2$ (task T_3) consisted of a skeleton. Figure 5.9 shows the guiding image presented to users. The selection of this point set came from a real desktop medical application in which doctors have to introduce this specific set of points for performing some automatic calculations [BQA+13].

The models have different dimensions, ranging from 128×128×128 to 512×512×512 (see Table 5.1).

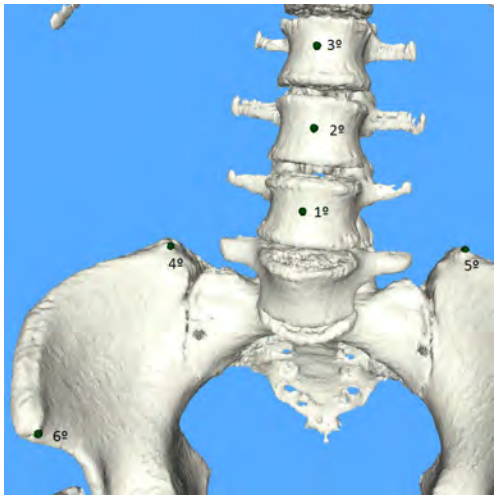


Figure 5.9: Relevant points that medical doctors selected from a real desktop medical application for performing some automatic calculations [BQA+13]. These points are the ones used in task T_3 to analyze precision.

Subjects and procedure

17 subjects participated in the evaluation; 13 male and 4 female, ranging between 23 and 63 years old. Subjects were asked to classify (as Low, Medium or High) their previous experience in a VR setup, their previous experience with input devices and their expertise in 3D application. All of the participants were people from our department: computer scientists at different levels of studies (master and PhD students) and faculty staff.

All the subjects participated in $Test_1$. Only a subset of them participated in $Test_2$ (13 subjects: 10 male and 3 female, ranging between 23 and 40 years old). Every user performed each test once.

Figure 5.10 shows the scheme of $Test_1$ and $Test_2$. As mentioned before, $Test_1$ consisted of two tasks: selecting two predefined points (T_1), and measuring a certain distance (T_2). For T_1, we asked the users to introduce two anchor points (P_1 and P_2) at positions that were marked in the model with the use of a cone (see Figure 5.8). Once completed, we stopped tracking the movements of the user until he or she was ready for the next task. T_2 consisted of taking a measure (calculated as a distance between two anchor points). The specification of this task was accompanied with an oral explanation of the goal and the pictures shown in Figure 5.11. None of the users involved in the experiment had any problem understanding the objective of the task.

Before $Test_1$ started, a complete training (using the spheres dataset) was performed for the users to get familiar with the two interaction techniques to evaluate: DAAPMed and CP techniques. Each test was divided into two blocks, one for each technique. The order of the blocks was chosen randomly in order to avoid skewing one of the techniques with a learning effect. Users were allowed to repeat the selection of a point as many times as needed, until the point was validated.

In the $Test_2$ session, we proceeded in the same way as in $Test_1$. Each participant performed the

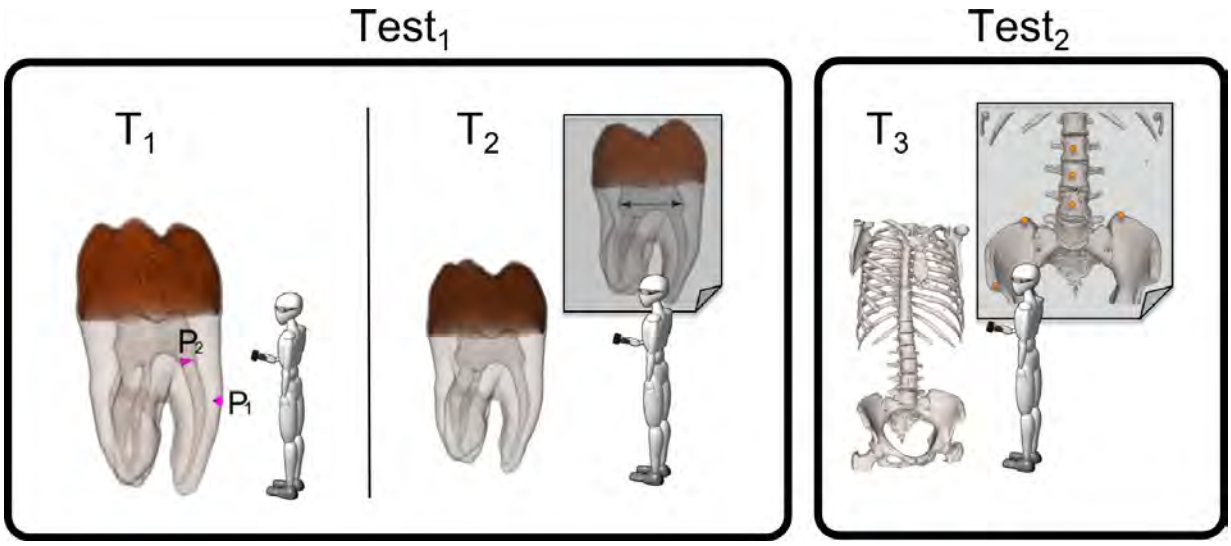


Figure 5.10: Scheme followed in $Test_1$ and $Test_2$. $Test_1$ consisted of two tasks. In the first task (T_1), users had to select, as precisely as they could, the points shown in the volume dataset. In the second task (T_2), users had to calculate a measurement described with the use of an oral explanation and a picture. In $Test_2$ (task T_3), users had to select, as precisely as they could, the points described in the reference picture.

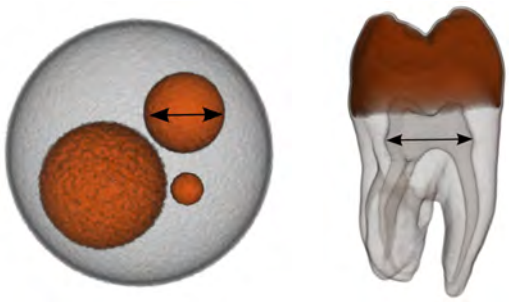


Figure 5.11: Images showing the description of task T_2, as presented to the participants in the test.

test once. Before the experiment, users were provided with a very short (1-3 min.) training session. The test was divided into two blocks, one for each technique to evaluate: CP and DAAPMed. The order of the blocks was chosen randomly in order not to introduce a learning effect.

5.4.3 Statistical results

A repeated measures within subjects design was used. The independent variable was the technique and the dependent variables were the set of tracked variables. A one-way analysis of variance (ANOVA) comparing both techniques was used.

$Test_1$ results

Table 5.2 summarizes the statistical analysis of the relevant variables for test $Test_1$. For each variable the mean and the standard deviation are shown. Task T_1 is tagged as P_1 and P_2, corresponding to the two anchor points to be selected. Regarding the mean and the standard deviation, DAAPMed is superior to CP for all the recorded measures. The one-way ANOVA analysis show which differences are statistically significant.

	CP			**DAAPMed**		
	P_1	P_2	T_2	P_1	P_2	T_2
Time	62.42 ± 34.08	73.8 ± 47.1	119.8 ± 65.5	43.07 ± 36.54	41.1 ± 25.7	84.1 ± 43.4
Device Mov.	3.711 ± 2.75	4.86 ± 4.8	7.92 ± 5.57	2.33 ± 3.26	1.88 ± 1.46	5.42 ± 3.31
User Mov.	1.94 ± 1.53	2.41 ± 2.33	4.281 ± 3.23	1.33 ± 1.87	1.26 ± 1.04	2.835 ± 1.79
Accuracy	0.76 ± 0.23	0.93 ± 1.37	1.15 ± 0.81	0.56 ± 0.23	1.37 ± 3.13	1.08 ± 0.79

Table 5.2: The overall statistical results of the evaluation shown as means and standard deviations of the variables measured for the tooth model. Note that regarding the mean and the standard deviation, DAAPMed is superior to CP for all the recorded measures.

Regarding **Completion Time** (*Time*), there is a significant evidence in all the experiments that DAAPMed performed better than CP. For P_1 ($p = 0.028$, $F = 5.83$), for P_2 ($p = 0.008$, $F = 9.35$) and for T_2 ($p = 0.044$, $F = 4.79$). Figure 5.12 shows a boxplot of the total time for each technique.

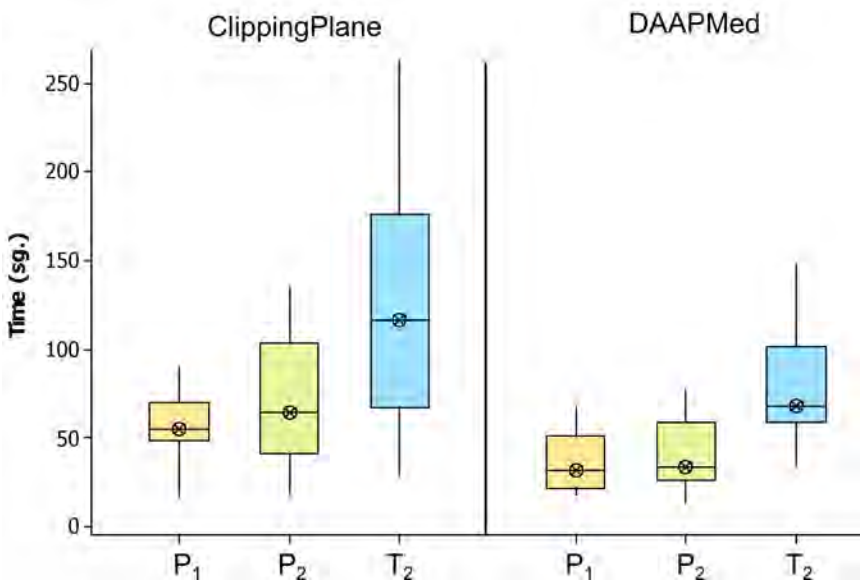


Figure 5.12: Results of the completion task timings for $Test_1$. The boxes show the interquartile range with the median as the horizontal bar. The whiskers extend to the minimum and maximum of the data. CP exhibits longer selection times than DAAPMed.

Regarding the **Input Device Footprint** (*Device Mov.*), we measured the length of the total path the

device covered to complete the experiment. We have found a significant effect on the *Input Device Footprint* variable for P_1 ($p = 0.036$, $F = 5.24$) and for P_2 ($p = 0.004$, $F = 11.70$). Figure 5.13 illustrates the effect of the reduction of the footprint for DAAPMed technique. The reduction of footprint is especially important since a handheld 6-DOF device is being used, which can lead to fatigue with extended use [WS91].

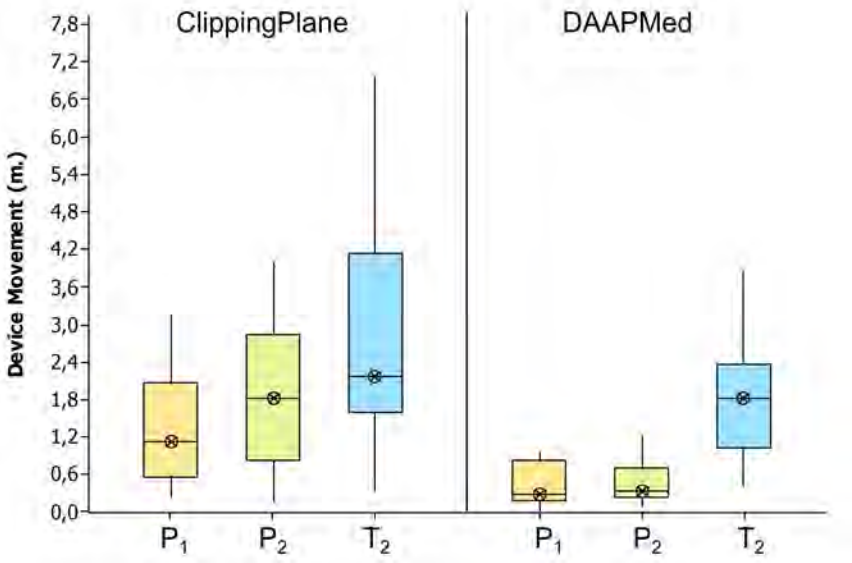


Figure 5.13: Input device footprints. The boxes show the interquartile range with the median as the horizontal bar. The whiskers extend to the minimum and maximum of the data. For point selection it is clear that DAAPMed method performed significantly better than CP.

We also split the movement of the device to take into account whether the movement was due to the exploration phase (rotating or translating the model) or due to the selection phase. We had only found significant statistical difference between the two techniques for P_2 ($p = 0.007$, $F = 9.44$). For the rest of the experiments, DAAPMed performed better comparing means and standard deviations. With CP technique, the user performs similar amount of moves during the exploration and during the selection. On the other hand, when selecting using DAAPMed technique, users devoted a larger effort to the exploration phase than to the selection one (see Figure 5.14).

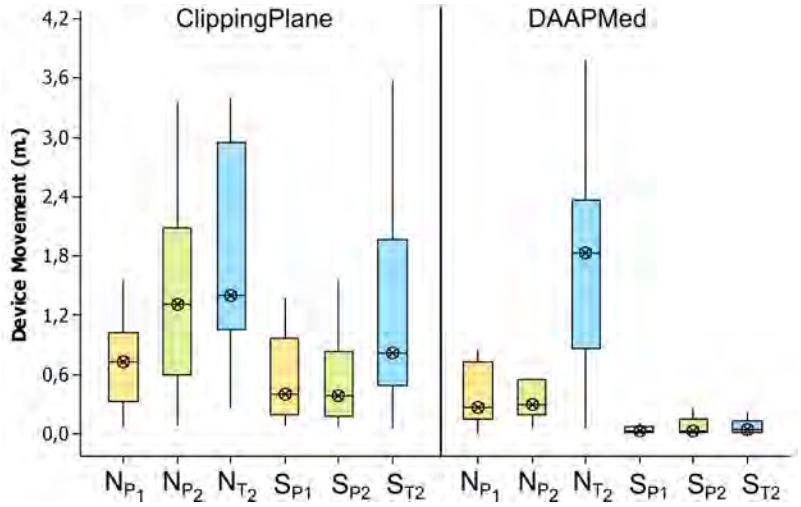


Figure 5.14: Input device footprints. The displacement carried out by the device is split in two states: navigation $N_{\{P_1,P_2,T_2\}}$ and selection $S_{\{P_1,P_2,T_2\}}$. The boxes show the interquartile range with the median as the horizontal bar. The whiskers extend to the minimum and maximum of the data. As shown, in DAAPMed, once the user has decided the best viewpoint for performing the selection, he or she performs the selection task with a small device movement with respect the movement done in the exploration task.

We also measured the movement carried out by the user (*User Mov.*). In all cases, DAAPMed required a lower amount of user movement. The analysis shows that the movement done in DAAPMed is significantly less than with CP for P_2 ($p = 0.009$, $F = 8.72$) and for T_2 ($p = 0.03$, $F = 5.62$) (see Figure 5.15).

Concerning the **accuracy**, the mean values show better performance for our technique. However, we did not find significant statistical differences. A possible explanation is that with the CP technique you can get enough precision if you know exactly which point you have to select (because we are using a cone to indicate the exact position of the point the user has to select, it is much easier to locate the clipping plane correctly). In order to do a deeper analysis, we performed another test, $Test_2$, which is closer to a real medical scenario since we use points with anatomical significance.

We also tracked an additional set of variables, but we could not extract any behavior or pattern from the results:

- Hit rate: This variable tracks the number of hits the user has to do. Since each introduced point may be changed if it is not satisfactory, we count the number of times a point is selected before its validation.
- Exploration or selection rate: This variable tracks the number of exploration versus selection phases the user has to perform to complete a task.

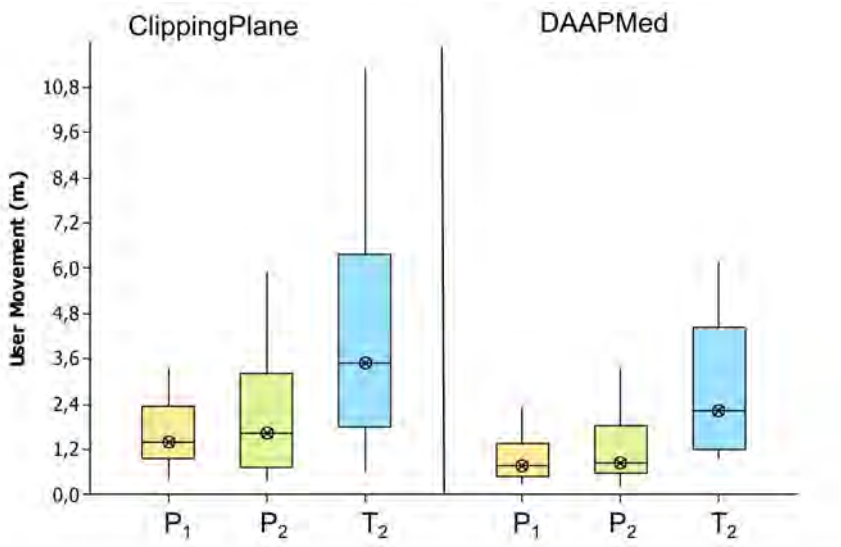


Figure 5.15: User footprint. The boxes show the interquartile range with the median as the horizontal bar. The whiskers extend to the minimum and maximum of the data. In all cases, DAAPMed required a lower amount of user movement.

$Test_2$ results

As has been exposed in Section 5.4.2, we have carried out a second experiment where the workflow is closer to a real medical environment. The points to be selected by the users have an specific anatomic meaning, and are commonly used to take measurements. The objectives of this test were twofold: a) finding whether DAAPMed technique was more accurate than the CP technique, and b) Testing if our VR application was as accurate as a desktop application.

Table 5.3 summarizes the statistical analysis of the relevant variable (**accuracy**) for test $Test_2$. The first and second rows show the mean and the standard deviation for each technique. The third row shows the statistical significance information (p and F). For all the points introduced (except P_4 and P_5), the DAAPMed technique shows a statistically significant improvement with respect to CP. We do not have a clear idea on the lack of significance of points P_4 and P_5, but it might show that the specification of their corresponding positions was not as clear as with the others. Figure 5.16 is a boxplot of all the performed tasks.

Although the goal of this test was to investigate the accuracy of our technique, we also tracked the required time to finish each task. Figure 5.17 is a convincing graphic showing the improvement in efficiency that the DAAPMed technique achieves.

	P_1	P_2	P_3	P_4	P_5	P_6
CP	2.944 ± 1.30	3.019 ± 1.49	3.171 ± 1.729	2.34 ± 0.88	2.066 ± 1.09	2.07 ± 1.17
DAAPMed	1.29 ± 0.67	1.70 ± 0.70	1.60 ± 0.50	1.77 ± 0.57	1.79 ± 0.42	0.28 ± 0.08
p,F	0.002 - 16.55	0.011 - 9.01	0.005 - 11.58	0.187 - 1.96	0.385 - 0.81	0.001 - 17.42

Table 5.3: The overall statistical results of the evaluation shown as means and standard deviations of the tolerance error. We can clearly see how the DAAPMed metaphor provides better results for all the points than the CP method.

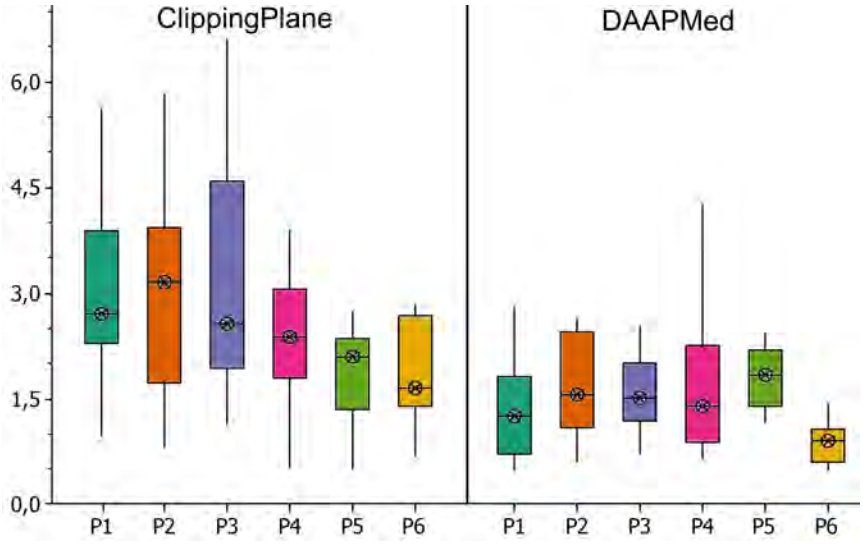


Figure 5.16: Accuracy by technique. The boxes show the interquartile range with the median as the horizontal bar. The whiskers extend to the minimum and maximum of the data. We can clearly see how DAAPMed achieves better accuracy for all the points than the CP method.

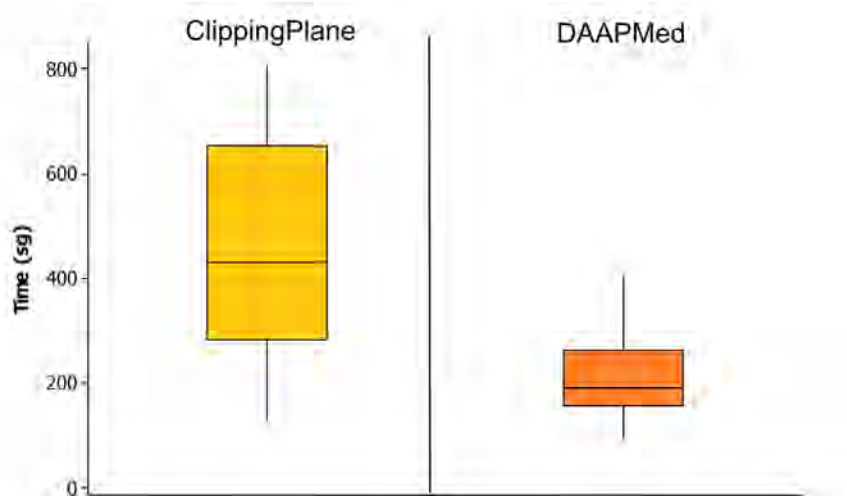


Figure 5.17: Total time to complete the introduction of the six points by technique. CP exhibits longer selection times than DAAPMed.

Additionally, using the same volume dataset, we have also compared the precision of DAAPMed in VR with a desktop application for the morpho-analysis of the abdominal air [MMPN+13]. In this application, a subset of the users which perform the user study, had to mark the set of points used in $Test_2$ on the skeleton, in order to infer some measures. This application works with triangle meshes and allows point selection on these meshes using a tool which follows the first hit in a ray-based paradigm. In order to do the comparison, the same isosurfaces used in the VR setup in the task T_3 were extracted using the Marching Cubes algorithm. The comparison showed that DAAPMed was as accurate as the desktop-based application. In both cases, the error performed was below the voxel size.

5.4.4 Post-questionnaire results

To complete the information, participants were asked to fill some questionnaires. This information provided additional insight about the preferences of the users between the two techniques. All responses in the questionnaire were measured on a Likert scale of 1-5, where 1 meant the worst value and 5 was the best value. The results are shown in Figure 5.18. The answers seem to indicate that DAAPMed metaphor is more suitable than the CP technique.

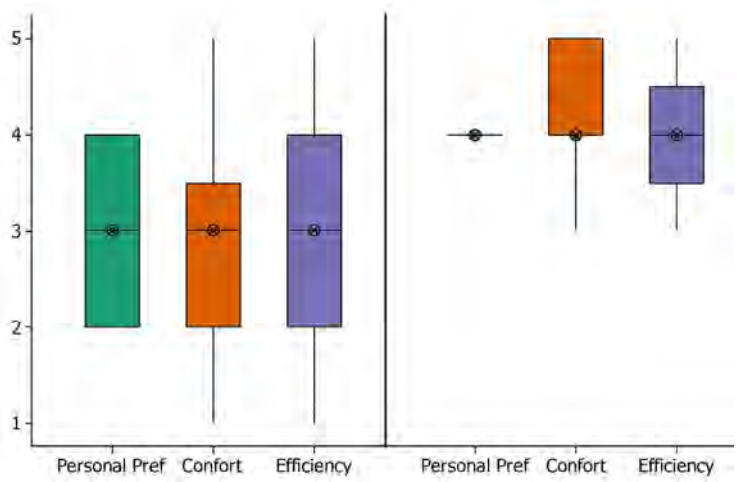


Figure 5.18: Results obtained from a personal preference evaluation questionnaire. These results show that the users' perceptions are quite positive with DAAPMed selection metaphor.

Although we got a positive feedback, users also mentioned two problems with respect to our technique. The first one is the inherent jittering of the tracker, that made selection affect user performance. Only two users believed it affects more to the ray-based selection than to the plane-based. Although, in all the experiments, the ray-based approach showed better behavior than the clipping-planes approach. The second issue was the lack of ray refinement: most users suggested that a fine tuning of the ray after its initial positioning would be welcome.

5.4.5 Discussion

One of the main conclusions the user study revealed was that DAAPMed is easy to learn and to use. None of the participants in the study had any problem in understanding the technique and in using it. In terms of accuracy, DAAPMed obtained better results than the classical selection using clipping planes (CP). Furthermore, when comparing our technique with a desktop-based application, DAAPMed obtained an accuracy comparable to the desktop one. In terms of comfortability, DAAPMed reduces the amount of movements and time required for the anchor points selection when compared with CP. Users felt more comfortable and achieved better results in less time with DAAPMed than with the CP technique.

However, the user study revealed us some problems inherent to the interaction with 3D input devices in VR setups. Working with 3D input devices requires a steady hand in order to obtain an accurate selection due to the inherent jittering of the tracker. Moreover, the developed solution to prevent the Heisenberg effect (see Section 5.2.1) increases the time users have to maintain, as stable as they can, the device, in order to wait until the circle has been completely drawn. The increment in time (which is negligible for steady hands) with the combination of quivery hands can make the selection task very exhausting to the user. Next section explains the solution we adopted to solve this problem.

Shake filtering

As has been exposed, trembling hands affect the overall performance and make the users end the VR experience with a bad sensation of using this kind of input devices. This may produce a complete refusal of the use of virtual reality. In order to reduce the effect of quivery hands, we propose to combine the use of the freezing time (see Section 5.2.1) with an averaging of the captured position. This averaging is computed while the circle of the freezing timer is being painted. The algorithm refines the position of the ray by taking the average of the last 20 captured positions by the tracker. It further checks whether the final position falls within a maximum tolerance range from the position at which the selection button was initially pressed. This filtering technique adds stability to the selection process. Algorithm 4 sketches the pseudo-code of this mechanism.

Algorithm 4 Shake filtering algorithm.

```
position_freezing = captured position                ▷ when user presses the corresponding button
repeat
    position = average of the last last 20 captured positions
    if distance(position, position_freezing) > Tolerance_mov then
        breakMov = true
    end if
until user releases the button or breakMov
if user releases the button when timer has finished then
    selection ray is locked at position_freezing
end if
```

User	Without Shake filtering		With Shake filtering		Perception Time
		Error	Time	Error	
user1	0.021	11.245	0.013	10.581	Better with smoothing
user2	0.012	30.848	0.011	22.458	Not clear (depending the trial)
user3	0.011	17.417	0.012	24.939	Indifferent
user4	0.012	17.48	0.011	10.598	Indifferent

Table 5.4: Results for the evaluation performed for analyzing the effect of the smoothing applied to the captured data. Users 1 and 2 were previously classified as bad steady hand. Users 3 and 4 were classified as good steady hand.

Although we have not performed a full user study after introducing this filter, we asked the two participants of the user study that showed a bad steady hand to experiment with the improved method. Also, two participants with a good steady hand performed the evaluation. The participants with bad steady hand performed better using the improvement introduced (see Table 5.4). One of them obtained slightly better results (with a precision improvement of around the 10%), but the second one showed an increase in precision of around the 40%. These results look promising, but further tests have to be carried out. The users with a good steady hand did not reveal any clear preference or inconvenient with both methods.

5.5 Conclusions

In this chapter we have presented DAAPMed, a new interaction technique for selecting points in a volume dataset [MVN13]. DAAPMed is based in the *ray casting* paradigm, enhanced with an automatic calculation of the set of potential anchor points by an on-the-fly determination of the isosurfaces along the ray path. While the user is interacting with the tool, we incorporate a visual feedback with a meaningful visualization called *Helper Views*. These provide context for the ray selection and show occluded candidate points that would be otherwise invisible to the user without posterior and ad-hoc volume manipulation. The user study demonstrated that DAAPMed technique is easy to learn and to use. Furthermore, it also reduces the efforts (hand displacements) and time required for the selection as compared with a clipping plane-based selection technique. Users felt more comfortable and achieved better results with DAAPMed than with the clipping plane technique. The user study also showed that the proposed technique is effective, with an accuracy comparable to that of a selection tool in a desktop-based application with a mouse. Some users suggested that a fine tuning of the ray, after its initial positioning, would be welcome. We let this work for future improvements.

The main limitations of our technique reside in the quality of the data of the volume dataset with respect to the presence of a considerable amount of noise. For example, the acquisition of head CT-scans for patients with metal implants produce images with streak artifacts in the areas surrounded by the metal implant. In these areas, our technique will detect points, despite not belonging to a bound-

ary of any anatomical structure. Another limitation resides where the boundaries between different anatomical structures can not be determined using only the medical images. In these cases, a segmentation process will be needed. In this case, we could adapt the test used to detect the candidate points set very easily. We simply should have to test if the point belongs to the boundary of the segmented region, which can be tested analyzing just the neighborhood of this point.

It is important to notice that taking into account that DAAPMed works in a VR setup and the level of interaction with the user is very high, a minimum frame-rate of 15 frames per second has to be guaranteed in order to achieve a good response from the users. As was stated in Section 5.3.3, the overall performance strongly depends on the visualization of the volume dataset, and this depends on the size of the volume dataset and also on the transfer function. It has to be taken into account that the volume dataset has to be rendered three times for each frame (one for the visualization of the volume model and two times for the Helper Views visualization). In addition, the scene has to be rendered twice: once for each viewpoint. So, in total, we have to render the volume dataset six times each frame (although the viewport used for each rendering has different size). Thus, depending on the size of the volume dataset and the transfer function in use, some optimization techniques should be applied in the way of handling with large volume datasets in order to achieve the required frame-rate.

CHAPTER 6

CONCLUSIONS

Throughout the development of the thesis we have proposed different techniques oriented to improve the user experience in two of the main blocks of the classical pipeline of a medical application: *Analysis* and *Exploration*.

In this Chapter we summarize the contributions of the thesis, propose some topics of future research, and list the publications that it has produced.

The main contributions are listed below:

Example-guided Segmentation The new algorithm presented in Chapter 2 permits the quasi-automatic segmentation of bones, specially those which are in joints. Segmentation of joint bones is particularly challenging, because they are too close to each other to be distinguishable in Computed Tomography (CT) images. We have proposed a model-based algorithm guided by deformation techniques inspired both by Geometric Processing techniques and by volume region-based information. The algorithm uses an example mesh of the same anatomical structure, usually from a different person, to drive the segmentation process. The final result is based both on the patient's captured volume information and on the geometrical shape of the example mesh. The algorithm is based on an energy minimization scheme to deform the initial example mesh while following the features of the captured volume data in a local and adaptive way. The algorithm has been tested on foot bones, obtaining a good convergence rate and reasonable residual errors. The resulting average error is of the order of the scale of the size of the voxels. This work has been published in [P_5] (The publications are listed in Section 6.2).

Good Views for Volume Models The approach described in Chapter 3 allows users to obtain a quick previsualization of a volumetric dataset in a short time with the use of an automatic algorithm. The technique is based on entropy measures for the generation of good viewpoints and on a

complexity-based metric for the calculation of a set of representative views. Our proposal works upon a model (a raw volume dataset) classified through the definition of a transfer function and, optionally, the specification of a region of interest. Starting from this minimal information, it automatically generates, both a set of representative views of the model and an exploration path that allows users to get an initial comprehension of the volume dataset before beginning the exploration task. The algorithm only uses the images generated through a DVR algorithm. In most cases the required total time is comparable to the loading time of the volume model. The different parts of this work have lead to several contributions, starting with the theoretical formulation [P_1], its application to the medical domain [P_2], and its extensions providing performance improvements and the explanation of more experiments that evaluate the quality of our measure [P_4].

The Virtual Magic Lantern Metaphor The approach described in Chapter 4 yields a friendly and usable interaction technique that facilitates the inspection of medical models with the simultaneous use of two different transfer functions or rendering styles. The *Virtual Magic Lantern* (VML) metaphor behaves like a lantern whose illumination cone determines the region of interest. The lantern is guided by a 3D pointer device that provides the axis direction and the apex position of the VML shape. The region of interest is rendered using another transfer function providing a feature rich volume inspection experience. It addresses the occlusion management problem, facilitating the inspection of inner structures without the total elimination of the exterior structures, offering in this way a context-based visualization of the overall structures. VML is particularly useful in Virtual Reality setups, because the interaction becomes very natural. The showrooms have demonstrated a very good acceptance of these techniques from the medical community and its potential use in concrete areas of the medical practice. The user study showed that our technique is easy to use and effective. This work has been partially published in [P_3].

DAAPMed: a data-aware picking technique This new selection technique presented in Chapter 5 allows users to easily select anchor points in non necessarily segmented volume datasets rendered using DVR in VR setups. It is based on the use of a ray emanating from the user, whose trajectory is enriched with the information on the points of intersection with the structures traversed by it. While the user is interacting with the tool, we incorporate visual feedback with a meaningful visualization called *Helper Views*. These views provide context for the ray selection and show occluded candidate points that would otherwise be invisible to the user without posterior and ad-hoc volume manipulation. The user study showed that our technique is easy to learn and to use. Furthermore, it also reduces the efforts (hand displacements) and time required for the selection when compared with the clipping plane technique. Users felt more comfortable and achieved better results with DAAPMed than with the clipping plane technique. This work has been published in [P_6]. A deeper explanation of the theoretical formulation and of the user

study carried out and its possible extensions was accepted for publication in a Lecture Notes Series.

6.1 Future research

The use of VR in the area of medicine is continuously growing and the introduction of affordable interaction devices promises its incorporation in the daily medical practice. In the future we want to continue researching in this area. We are focused on improving the task of exploration and manipulation of a volume dataset with more natural and more comfortable techniques. We should not forget that the introduction of affordable 3D input devices comes with reduced precision. Therefore, the study and development of new interaction techniques to overcome this limited precision is mandatory.

Moreover, as has been pointed out in the Conclusions of the different techniques developed for Virtual Reality environments, special attention has to be devoted to the performance achieved in order to guarantee a good response to the users. As exposed, the overall performance of the developed techniques strongly depends on the visualization of the volume dataset, and this depends both on the size of the volume dataset and on the design of the transfer function. Thus, depending on the size of the volume dataset and the transfer function in use, some optimization techniques should be applied in the way of handling with large volume datasets in order to achieve the required frame-rate. Therefore, we shall closely follow the research, currently and in the future, in the subject of big datasets, in order to evaluate which of these works could be adapted to our techniques in VR setups.

Currently, we are working in two different projects. First of all, we want to study the possibilities that VML (see Chapter 4) and DAAPMed (see Chapter 5) techniques can offer if they work together. We guess that this idea will do DAAPMed more powerful, increasing its capabilities by coupling it with the VML metaphor (Figure 6.1). The *Ray cursor tool* from DAAPMed is located in the axis of the VML shape. The set of candidate selection points are calculated taking into account the transfer function used inside the VML region. In this way, the user can pick points inside a volume dataset without losing the overall context provided by the other transfer function. We want to explore the different possibilities for the contents of the Helper Views and how the exchange between both techniques has to be managed in order to not lose the anchor points introduced once the VML is out of their scope.

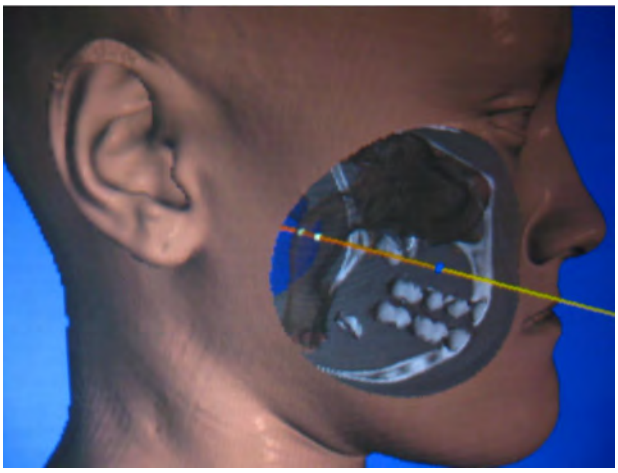


Figure 6.1: Image showing the integration of DAAPMed selection technique in VML metaphor. The image shows a head consisted of the skin, while the VML region shows the air-cavities interior to the body. The trajectory of the ray shows the potential anchor points located at the intersections of the ray with the surfaces shown by the transfer function used in the VML region of interest.

In parallel, we are working on a new technique focused on improving the *Exploration* task by minimizing the user movements required to explore a concrete part of a volume model. We are developing a zooming technique, called *Zoom-in-Place*, which builds the zooming result in the same virtual position of the initial interaction while still maintaining a contextual view on the region of interest and its surroundings (see Figure 6.2). This way, the user reduces the amount of movements required to explore a model. Preliminary results has been presented in [P_7].

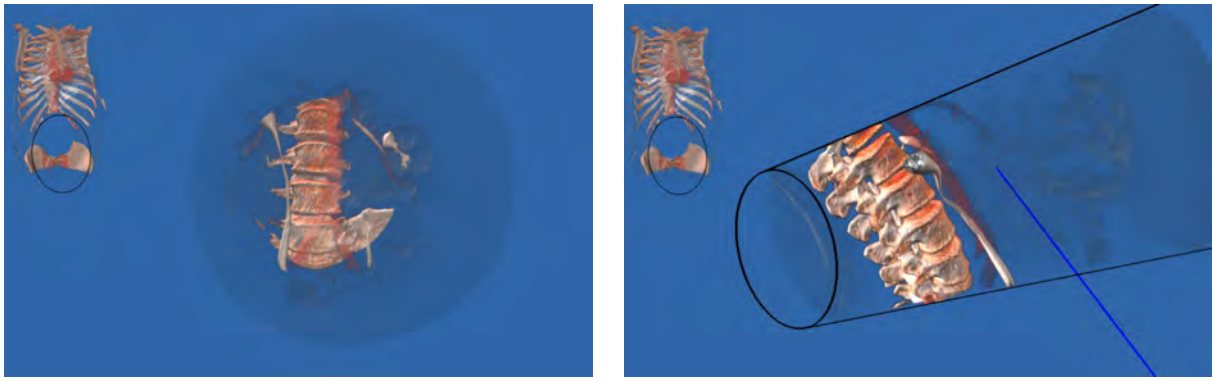


Figure 6.2: Magnification effect and interactive exploration of the magnified region. After the selection of the region of interest, this region is uncoupled from the rest of the volume model and moved closer to the user. At the same time, the remainder volume is moved away from the center of the view, and a reference image remains on the top left corner to provide context to the selected region. Then, the region of interest can be interactively explored by the user.

6.2 Publications

The following papers have been published as a result of the research developed in this thesis:

[P_1] Pere-Pau Vázquez, Eva Monclús, and Isabel Navazo. Representative views and paths for volume models, *Symposium on Smart Graphics*, pages 106-117, Rennes, France. August 27-29, 2008.

[P_2] Eva Monclús, Pere-Pau Vázquez, Isabel Navazo, Javier Herrero, and Jordi López. Automatic generation of exploration paths for medical models, *International Journal of Computer Assisted Radiology and Surgery*, 3(1):52-53, 2008.

[P_3] Eva Monclús, José Díaz, Isabel Navazo, and Pere-Pau Vázquez. The Virtual Magic Lantern: an interaction metaphor for enhanced medical data inspection, *ACM Symposium on Virtual Reality Software and Technology*, pages 119-122, Kyoto, Japan. November 18-20, 2009.

[P_4] Pere-Pau Vázquez, Eva Monclús, and Isabel Navazo. Efficient selection of representative views and navigation paths for volume data exploration. *Visualization in Medicine and Life Sciences II*, pages 135-153. Springer-Verlag, Mathematics + Visualization series, 2012.

[P_5] Antoni Chica, Eva Monclús, Pere Brunet, Isabel Navazo, and Àlvar Vinàcua. Example-Guided segmentation. *Graphical Models*, Vol.74(6), pages 302-310, 2012.

[P_6] Eva Monclús, Pere-Pau Vázquez, and Isabel Navazo. DAAPMed: a Data-Aware Anchor Point selection tool for Medical models in VR environments. *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, pages 308-317, Barcelona, Spain. February 21-24, 2013. Best Student Paper Award.

[P_7] Eva Monclús, Pere-Pau Vázquez, and Isabel Navazo. Minimizing user movement with Zoom in Place. 2^{ND} *International Workshop on Inmersive Volumetric Interaction (WIVI 2014) at IEEE Virtual Reality*. Minneapolis, MN, USA. March 29, 2014.

Moreover, the author has also collaborated in other medical research. The most relevant publications, related to the topics of the thesis, are:

- Norberto Ezquerra, Isabel Navazo, Tahía Infantes, and Eva Monclús. Graphics, Vision and Visualization in Medical Imaging. A state-of-the-art Report. *Eurographics'99*, pages 21 – 80, Milán, Italy. September 1999.
- Eva Monclús, Pere-Pau Vázquez, and Isabel Navazo. MTCut: GPU-based Marching Tetra Cuts, *Theory and Practice of Computer Graphics Conference*, pages 37 – 44, Bangor, UK. June 13-15, 2007.

- Joan Antoni Hueto, Gerard Raspall, Eva Monclús, Pere-Pau Vázquez, José Díaz, and Isabel Navazo. 3D Virtual model for training in endoscopical repair of subcondilar fractures of the Mandible, *International Journal of Computer Assisted Radiology and Surgery*, 3(1):292. 2008.

- José Díaz, Eva Monclús, Isabel Navazo and Pere-Pau Vázquez. Adaptative cross-sections of anatomical models. *Computer Graphics Forum*, 31(7), pages 2155 – 2164. 2012.

- Imanol Muñoz, Eva Monclús, Pere Brunet and Gerard Conesa. Ventricular puncture trainer. *Spanish Computer Graphics Conference (CEIG'12)*, pages 45 – 48, Jaen, Spain. September 2012.

- Eva Monclús, Imanol Muñoz, Isabel Navazo, Pere-Pau Vázquez, Anna Accarino, Elisabeth Barba, Sergi Quiroga and Fernando Azpiroz. Morpho-volumetric measurement tools for abdominal distension diagnose, *Spanish Computer Graphics Conference (CEIG'13)*, pages 39 – 48, Madrid, Spain. September 2013.

- Elisabeth Barba, Sergi Quiroga, Anna Accarino, Eva Monclús, Carolina Malagelada, Emanuel Burri, Isabel Navazo, Juan-R Malagelada and Fernando Azpiroz. Mechanisms of abdominal distension in severe intestinal dysmotility: abdomino-thoracic response to gut retention. *Neurogastroenterology and motility*, 25(6):e389 – e394, 2013.

BIBLIOGRAPHY

[AA09] F. Argelaguet and C. Andujar. Efficient 3D pointing selection in cluttered virtual environments. *IEEE Computer Graphics and Applications*, 29:34–43, 2009.

[AB94] R. Adams and L. Bischof. Seeded region growing. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 16(6):641–647, 1994.

[Bai04] D. G. Bailey. An efficient euclidean distance transform. In *Combinatorial Image Analysis (IWCIA)*, pages 394–408, 2004.

[BG06] S. Bruckner and M. E. Gröller. Exploded views for volume data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1077–1084, 2006.

[BG07] S. Bruckner and M. E. Gröller. Style transfer functions for illustrative volume rendering. *Computer Graphics Forum*, 26(3):715–724, 2007.

[BGKG06] S. Bruckner, S. Grimm, A. Kanitsar, and M. E. Gröller. Illustrative context-preserving exploration of volume data. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1559–1569, 2006.

[BGL+98] C. Bennett, P. Gacs, M. Li, P. Vitányi, and W. Zurek. Information distance. *IEEE Transactions on Information Theory*, 44:1407–1423, 1998.

[BH06] L. D. Brown and H. Hua. Magic lenses for augmented virtual environments. *IEEE Computer Graphics and Applications*, 26(4):64–73, 2006.

[BHW+07] M. Burns, M. Haidacher, W. Wein, I. Viola, and E. Gröller. Feature emphasis and contextual cutaways for multimodal medical visualization. In *Eurographics / IEEE VGTC Symposium on Visualization (EuroVis 2007)*, pages 275–282. IEEE, 2007.

[BKLP04] D. Bowman, E. Kruijff, J. LaViola, and I. Poupyrev. *3D User Interfaces: Theory and Practice*. Addison-Wesley, Pearson Education, 2004.

[Boo09] W. Boonsuk. *Automated separation of bone joint structures for medical images*. PhD thesis, Iowa State University, 2009.

[BQA+13] E. Barba, S. Quiroga, A. Accarino, E. Monclús, C. Malagelada, E. Burri, I. Navazo, J.-R. Malagelada, and F. Azpiroz. Mechanisms of abdominal distension in severe intestinal dysmotility: abdomino-thoracic response to gut retention. *Neurogastroenterology and Motility*, 25(6):e389–e394, 2013.

[Bra] Bracco AMT. The Dextroscope platform. http://www.volumeinteractions.us/innovation.html.

[BRB+13] R. Bramon, M. Ruiz, A. Bardera, I. Boada, M. Feixas, and M. Sbert. An information-theoretic observation channel for volume visualization. *Computer Graphics Forum*, 32(3pt4):411–420, 2013.

[BS05] U. Bordoloi and H.-W. Shen. View selection for volume rendering. In *IEEE Visualization*, pages 487–494. IEEE Computer Society, 2005.

[BSP+93] E. A. Bier, M. C. Stone, K. Pier, W. Buxton, and T. D. DeRose. Toolglass and Magic lenses: the see-through interface. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '93, pages 73–80. ACM, 1993.

[BV09] Å. Birkeland and I. Viola. View-dependent peel-away visualization for volumetric data. In *Proceedings of the Spring Conference on Computer Graphics (SCCG)*, pages 121–128. ACM, 2009.

[BWC+02] D. Bowman, C. Wingrave, J. Campbell, V. Ly, and C. Rhoton. Novel uses of pinch gloves for virtual environment interaction techniques. *Virtual Reality*, 6:122–129, 2002.

[CAO07] M. Cebrián, M. Alfonseca, and A. Ortega. The normalized compression distance is resistant to noise. *IEEE Transactions on Information Theory*, 53(5):1895–1900, 2007.

[CMB+12] A. Chica, E. Monclús, P. Brunet, I. Navazo, and A. Vinàcua. Example-guided segmentation. *Graphical Models*, 74(6):302–310, 2012.

[CQWZ06] M.-Y. Chan, H. Qu, Y. Wu, and H. Zhou. Viewpoint selection for angiographic volume. In *Advances in Visual Computing*, volume 4291 of *Lecture Notes in Computer Science*, pages 528–537. Springer Berlin Heidelberg, 2006.

[CSC06] C. D. Correa, D. Silver, and M. Chen. Feature aligned volume manipulation for illustration and visualization. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1069–1076, 2006.

[CSS08] H.-L. J. Chen, F. F. Samavati, and M. C. Sousa. GPU-based point radiation for interactive volume sculpting and segmentation. *The Visual Computer*, 24(7):689–698, 2008.

[CV05] R. Cilibrasi and P. M. Vitanyi. Clustering by compression. *IEEE Transactions on Information Theory*, 51(4):1523–1545, 2005.

[CWA+08] A. Chica, J. Williams, C. Andújar, P. Brunet, I. Navazo, J. Rossignac, and A. Vinàcua. Pressing: Smooth isosurfaces with flats from binary grids. *Computer Graphics Forum*, 27(1):36–46, 2008.

[CZW+13] Y. Cheng, S. Zhou, Y. Wang, C. Guo, J. Bai, and S. Tamura. Automatic segmentation technique for acetabulum and femoral head in CT images. *Pattern Recognition*, 46(11):2969–2984, 2013.

[ENMM99] N. Ezquerra, I. Navazo, T. Morris, and E. Monclús. Graphics, vision, and visualization in medical imaging: A state of the art report. In *Eurographics'99*, pages 21–80, 1999.

[Erd12] M. Erdt. *Non-Uniform Deformable Volumetric Objects for Medical Organ Segmentation and Registration.* PhD thesis, Fraunhofer Institute for Computer Graphics Research, 2012.

[GB06] T. Grossman and R. Balakrishnan. The design and evaluation of selection techniques for 3D volumetric displays. In *Proceedings of the Symposium on User interface software and technology*, pages 3–12. ACM, 2006.

[GDPM08] L. Gallo, G. De Pietro, and I. Marra. 3D interaction with volumetric medical data: experiencing the wiimote. In *Proceedings of the 1st international conference on Ambient media and systems*, pages 14:1–14:6, 2008.

[GHP+95] J. Goble, K. Hinckley, R. Pausch, J. Snell, and N. F. Kassell. Two-handed spatial interface tools for neurosurgical planning. *Computer*, 28(7):20–26, 1995.

[GPE+13] S. Grau, A. Puig, S. Escalera, M. Salamó, and O. Amoros. Efficient complementary viewpoint selection in volume rendering. In *21st International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG)*, pages 69–78, 2013.

[GPZT98] E. Gobbetti, P. Pili, A. Zorcolo, and M. Tuveri. Interactive virtual angioscopy. In *Proceedings of the conference on Visualization*, VIS '98, pages 435–438. IEEE Computer Society Press, 1998.

[Gum02] S. Gumhold. Maximum entropy light source placement. In *Proceedings of the conference on Visualization*, VIS '02, pages 275–282. IEEE Computer Society, 2002.

[HBR+92] K.-H. Hohne, M. Bomans, M. Riemer, R. Schubert, U. Tiede, and W. Lierse. A volume-based anatomical atlas. *IEEE Computer Graphics and Applications*, 12(4):72–78, 1992.

[HD11] T. Heimann and H. Delingette. Model-based segmentation. In *Biomedical Image Processing*, Biological and Medical Physics, Biomedical Engineering, pages 279–303. Springer Verlag, 2011.

[HGM09] Hu, Grossberg, and Mageras. Survey of recent volumetric medical image segmentation techniques. *Biomedical Engineering*, pages 321–346, 2009.

[HJDP07] J. Hagedorn, P. Joy, S. Dunkers, and A. Peskin. Measurement tools for the immersive visualization environment: Steps toward the virtual laboratory. *Journal of Research of the National Institute of Standards and Technology*, 112(5):257–270, 2007.

[HKRs+06] M. Hadwiger, J. Kniss, C. Rezk-salama, D. Weiskopf, and K. Engel. *Real-time Volume Graphics*. A. K. Peters, Ltd., 2006.

[HM09] T. Heimann and H.-P. Meinzer. Statistical shape models for 3D medical image segmentation: A review. *Medical Image Analysis*, 13(4):543–563, 2009.

[Hou80] G. N. Hounsfield. Computed medical imaging. *Medical Physics*, 7(4):283–290, 1980.

[HPG94] K. Hinckley, R. Pausch, and J. Goble. A three-dimensional user interface for neurosurgical visualization. In *SPIE Conference on Medical Imaging*, pages 126–136. SPIE, 1994.

[HPGK94] K. Hinckley, R. Pausch, J. Goble, and N. Kassell. A survey of design issues in spatial input. In *Proceedings of the Symposium on User interface software and technology*, pages 213–222. ACM, 1994.

[HTEE98] P. Hastreiter, B. Tom, K. E. W. Eberhardt, and T. Ertl. Fast analysis of intracranial aneurysms based on interactive direct volume rendering and CT angiography. In *CTA, Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 660–669. Springer, 1998.

[IbHT+02] S. Iserhardt-bauer, P. Hastreiter, B. Tom, N. Kšstner, M. Schempershofe, U. Nissen, and T. Ertl. Standardized analysis of intracranial aneurysms using digital video sequences. In *Proceedings Medical Image Computing and Computer Assisted Intervention*, pages 411–418. MICCAI, Springer-Verlag, 2002.

[Int] Intersense, Inc. http://www.intersense.com/.

[JPIF10] D. J., V. P., N. I., and D. F. Real-time ambient occlusion and halos with summed area tables. *Computers & Graphics*, 34(4):337–350, 2010.

[JS06] G. Ji and H.-W. Shen. Dynamic view selection for time-varying volumes. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1109–1116, 2006.

[KD98] G. Kindlmann and J. W. Durkin. Semi-automatic generation of transfer functions for direct volume rendering. In *Proceedings of the 1998 IEEE symposium on Volume visualization*, VVS '98, pages 79–86. ACM, 1998.

[KDGB99] A. H. König, H. Doleisch, E. Gröller, and T. H. Brain. Multiple views and magic mirrors - fMRI visualization of the human brain, 1999.

[KEK03] Y. Kang, K. Engelke, and W. Kalender. A new accurate and precise 3-d segmentation method for skeletal structures in volumetric ct data. *Medical Imaging, IEEE Transactions on*, 22(5):586–598, 2003.

[Kir92] D. Kirk. *Graphic Gems III.* Academic Press, 1992.

[KRW+11] C. Kirmizibayrak, N. Radeva, M. Wakid, J. Philbeck, J. Sibert, and J. Hahn. Evaluation of gesture based interfaces for medical volume visualization tasks. In *Proceedings of the 10th International Conference on Virtual Reality Continuum and Its Applications in Industry*, pages 69–74. ACM, 2011.

[KSW06] J. Krüger, J. Schneider, and R. Westermann. ClearView: An interactive context preserving hotspot visualization technique. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):941–948, 2006.

[KVPL04] O. Konrad-Verse, B. Preim, and A. Littmann. Virtual resection with a deformable cutting plane. In *Proceedings of Simulation und Visualisierung*, pages 203–214, 2004.

[KWBH10] C. Kirmizibayrak, M. Wakid, S. Bielamowicz, and J. Hahn. Interactive visualization for image guided medialization laryngoplasty. *Computer Graphics International, 8-11 June 2010, Singapore*, 2010.

[KWT88] M. Kass, A. P. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[KWY+13] C. Kirmizibayrak, M. Wakid, Y. Yim, D. Hristov, and J. K. Hahn. Interactive focus+ context medical data exploration and editing. *Computer Animation and Virtual Worlds*, 2013.

[LCL+04] M. Li, X. Chen, X. Li, B. Ma, and P. Vitányi. The similarity metric. *IEEE Transactions Information Theory*, 50(12):3250–3264, 2004.

[LFG+98] L. Lorigo, O. Faugeras, W. Grimson, R. Keriven, and R. Kikinis. Segmentation of bone in clinical knee MRI using texture-based geodesic active contours. In *Medical Image Computing and Computer-Assisted Intervention, MICCAIÕ98*, volume 1496 of *Lecture Notes in Computer Science*, pages 1195–1204. Springer Verlag, 1998.

[LIGGM09] Y. Luo, J. A. Iglesias Guitián, E. Gobbetti, and F. Marton. Context preserving focal probes for exploration of volumetric medical datasets. In *Proceedings of the 2009 International Conference on Modelling the Physiological Human*, 3DPH'09, pages 187–198. Springer-Verlag, 2009.

[Lin91] J. Lin. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information theory*, 37:145–151, 1991.

[Lip] R. Lipschitz. http://en.wikipedia.org/wiki/Lipschitz_continuity.

[LRA+07] W. Li, L. Ritter, M. Agrawala, B. Curless, and D. Salesin. Interactive cutaway illustrations of complex 3d models. *ACM Transactions on Graphics*, 26(3), 2007.

[LUS+08] J. Liu, J. K. Udupa, P. K. Saha, D. Odhner, B. E. Hirsch, S. Siegler, S. Simon, and B. A. Winkelstein. Rigid model-based 3D segmentation of bones of joints in MR and CT images for motion analysis. *Medical Physics*, 35(8):3637 – 3649, 2008.

[LV93] M. Li and P. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer-Verlag, 1993.

[Mag] Magic Lantern Society. A history of the magic lantern. http://www.magiclantern.org.uk/index.html.

[MB06] T. McInerney and S. Broughton. Hingeslicer: interactive exploration of volume images using extended 3D slice plane widgets. In *GI'06: Proceedings of Graphics Interface*, pages 171–178. Canadian Information Processing Society, 2006.

[MDA01] J. Montagnat, H. Delingette, and N. Ayache. A review of deformable surfaces: topology, geometry and deformation. *Image and Vision Computing*, 19(14):1023–1040, 2001.

[MDMC90] O. Monga, R. Deriche, G. Malandain, and J. P. Cocquerez. Recursive filtering and edge closing: two primary tools for 3D edge detection. In *Proceedings of the first european conference on Computer vision*, ECCV 90, pages 56–65. Springer Verlag, 1990.

[MDNV09] E. Monclús, J. Díaz, I. Navazo, and P.-P. Vázquez. The virtual magic lantern: an interaction metaphor for enhanced medical data inspection. In *VRST '09: Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology*, pages 119–122. ACM, 2009.

[Min95] M. Mine. Virtual environment interaction techniques. Technical report, UNC Chapel Hill CS Dept, 1995.

[MMPN+13] E. Monclús, I. Muñoz-Pandiella, I. Navazo, P.-P. Vázquez, A. Accarino, E. Barba, S. Quiroga, and F. Azpiroz. Morpho-volumetric measurement tools for abdominal distension diagnose. In *XXIII Congreso Español de Informática Gráfica*, pages 39–48, 2013.

[MNTP07] K. Mühler, M. Neugebauer, C. Tietjen, and B. Preim. Viewpoint selection for intervention planning. In *EG/ IEEE-VGTC Symposium on Visualization*, pages 267–274, 2007.

[MRH08] J. Mensmann, T. Ropinski, and K. H. Hinrichs. Interactive cutting operations for generating anatomical illustrations from volumetric data sets. *Journal of WSCG – 16th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, 16(1-3):89–96, 2008.

[MT96] T. McInerney and D. Terzopoulos. Deformable models in medical image analysis: a survey. *Medical Image Analysis*, 1(2):91–108, 1996.

[MTB03] M. J. McGuffin, L. Tancau, and R. Balakrishnan. Using deformations for browsing volumetric data. In *Proceedings of IEEE Visualization 2003*, pages 401–408. IEEE, 2003.

[MVN13] E. Monclús, P.-P. Vázquez, and I. Navazo. A Data-Aware Anchor Point selection tool for Medical models in VR environments. In *8th International Conference on Computer Graphics Theory and Applications (GRAPP 2013)*, 2013.

[Nat] National Library of Medicine. The Visible Human Project. http://www.nlm.nih.gov/research/visible/visible_human.html.

[OF03] A. Olwal and S. Feiner. The flexible pointer: An interaction technique for augmented and virtual reality. In *Proceedings of ACM Symposium on User Interface Software and Technology*, pages 81–82, 2003.

[PB13] B. Preim and C. P. Botha. *Visual Computing for Medicine: Theory, Algorithms, and Applications*. Morgan Kaufmann, 2013.

[PBWI96] I. Poupyrev, M. Billinghurst, S. Weghorst, and T. Ichikawa. The Go-Go interaction technique: non-linear mapping for direct manipulation in VR. In *Proceedings of the 9th annual ACM symposium on User interface software and technology*, UIST '96, pages 79–80. ACM, 1996.

[PTSP02] B. Preim, C. Tietjen, W. Spindler, and H. O. Peitgen. Integration of measurement tools in medical 3D visualizations. In *Visualization '02*, pages 21–28. IEEE Computer Society, 2002.

[QM05] W. Qi and J.-B. Martens. Tangible user interfaces for 3D clipping plane interaction with volumetric data: a case study. In *Proceedings of the 7th international conference on Multimodal interfaces*, pages 252–258, 2005.

[RCD+10] I. Rössling, C. Cyrus, L. Dornheim, A. Boehm, and B. Preim. Fast and flexible distance measures for treatment planning. *International Journal of Computer Assisted Radiology and Surgery*, 5:633–646, 2010.

[Roe] S. Roettger. http://www9.informatik.uni-erlangen.de/External/vollib/.

[RSBB06] B. Reitinger, D. Schmalstieg, A. Bornik, and R. Beichel. Spatial analysis tools for virtual reality-based surgical planning. *3D User Interfaces*, 0:37–44, 2006.

[RSS01] J. Rossignac, A. Safonova, and A. Szymczak. Edgebreaker on a Corner Table: A simple technique for representing and compressing triangulated surfaces. In *Hierarchical and Geometrical Methods in Scientific Visualization*, pages 41–50. Springer Verlag, 2001.

[SCOL+04] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel. Laplacian surface editing. In *Proceedings of the Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, pages 179–188. ACM, 2004.

[SDM+01] L. Soler, H. Delingette, G. Malandain, J. Montagnat, N. Ayache, C. Koehl, O. Dourthe, B. Malassagne, M. Smith, D. Mutter, and J. Marescaux. Fully automatic anatomical, pathological, and functional segmentation from CT scans for hepatic surgery. *Computer Aided Surgery*, 6(3):131–142, 2001.

[SES05] N. Svakhine, D. S. Ebert, and D. Stredney. Illustration motifs for effective medical volume illustration. *IEEE Computer Graphics and Applications*, 25(3):31–39, 2005.

[SF00] M. Sonka and J. M. Fitzpatrick. *Handbook of Medical Imaging, Volume 2. Processing and Analysis*. SPIE, 2000.

[SGF+11] P. Song, W. B. Goh, C.-W. Fu, Q. Meng, and P.-A. Heng. WYSIWYF: exploring and annotating volume data with a tangible handheld device. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, CHI '11, pages 1333–1342. ACM, 2011.

[Sha48] E. C. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423,623–656, 1948.

[SMPA98] J. Starck, F. Murtagh, B. Pirenne, and M. Albrecht. Astronomical image compression based on noise suppression. *Publications of the Astronomical Society of the Pacific*, 108:446–455, 1998.

[STCK03] T. B. Sebastian, H. Tek, J. J. Crisco, and B. B. Kimia. Segmentation of carpal bones from CT images using skeletally coupled deformable models. *Medical Image Analysis*, 7(1):21 – 45, 2003.

[Tau95] G. Taubin. A signal processing approach to fair surface design. In *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, pages 351–358. ACM, 1995.

[TFTN05] S. Takahashi, I. Fujishiro, Y. Takeshima, and T. Nishita. A feature-driven approach to locating optimal viewpoints for volume visualization. In *IEEE Visualization*, pages 495–502. IEEE Computer Society, 2005.

[TLB+09] Y. Tao, H. Lin, H. Bao, F. Dong, and G. Clapworthy. Structure-aware viewpoint selection for volume visualization. *IEEE Pacific Visualization Symposium*, 0:193–200, 2009.

[TWK88] D. Terzopoulos, A. P. Witkin, and M. Kass. Constraints on deformable models: Recovering shape and non-rigid motion. *Artificial Intelligence*, 36(1):91–123, 1988.

[Váz03] P.-P. Vázquez. *On the Selection of Good Views and its Application to Computer Graphics*. PhD thesis, Politechnical University of Catalonia, Barcelona, Spain, 2003.

[Váz07] P.-P. Vázquez. Automatic light source placement for maximum illumination information recovery. *Computer Graphics Forum*, 26(2):143–156, 2007.

[VFSG06] I. Viola, M. Feixas, M. Sbert, and M. E. Gröller. Importance-driven focus of attention. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):933–940, 2006.

[VFSH01a] P.-P. Vázquez, M. Feixas, M. Sbert, and W. Heidrich. Viewpoint selection using viewpoint entropy. In *Proceedings of the Vision Modeling and Visualization Conference (VMV-01)*, pages 273–280, 2001.

[VFSH01b] P.-P. Vázquez, M. Feixas, M. Sbert, and W. Heidrich. Viewpoint selection using viewpoint entropy. In *Proceedings of the Vision Modeling and Visualization Conference 2001*, VMV '01, pages 273–280. Aka GmbH, 2001.

[VGC07] L. Vanacken, T. Grossman, and K. Coninx. Exploring the effects of environment density and target visibility on object selection in 3D virtual environments. In *IEEE Symposium on 3D User Interfaces, 3DUI'07*, pages 115–122, 2007.

[VM12] P.-P. Vázquez and J. Marco. Using normalized compression distance for image similarity measurement: an experimental study. *The Visual Computer*, 28(11):1063–1084, 2012.

[VMN08] P.-P. Vázquez, E. Monclús, and I. Navazo. Representative views and paths for volume models. In *Proceedings of the 9th international symposium on Smart Graphics*, SG '08, pages 106–117. Springer-Verlag, 2008.

[VS03] P.-P. Vázquez and M. Sbert. Perception-based illumination information measurement and light source placement. *Lecture Notes in Computer Science*, 2669:306–316, 2003.

[WEE03] D. Weiskopf, K. Engel, and T. Ertl. Interactive clipping techniques for texture-based volume visualization and volume shading. *IEEE Transactions on Visualization and Computer Graphics*, 9(3):298–312, 2003.

[WPVH13] A. Wiebel, P. Preis, F. M. Vos, and H.-C. Hege. 3D Strokes on Visible Structures in Direct Volume Rendering. In *EuroVis - Short Papers*, pages 91–95. Eurographics Association, 2013.

[WS91] C. Ware and L. Slipp. Using velocity control to navigate 3D graphical environments: A comparison of three interfaces. In *Human Factors and Ergonomic Studies (HFES) Meeting*, pages 25–32, 1991.

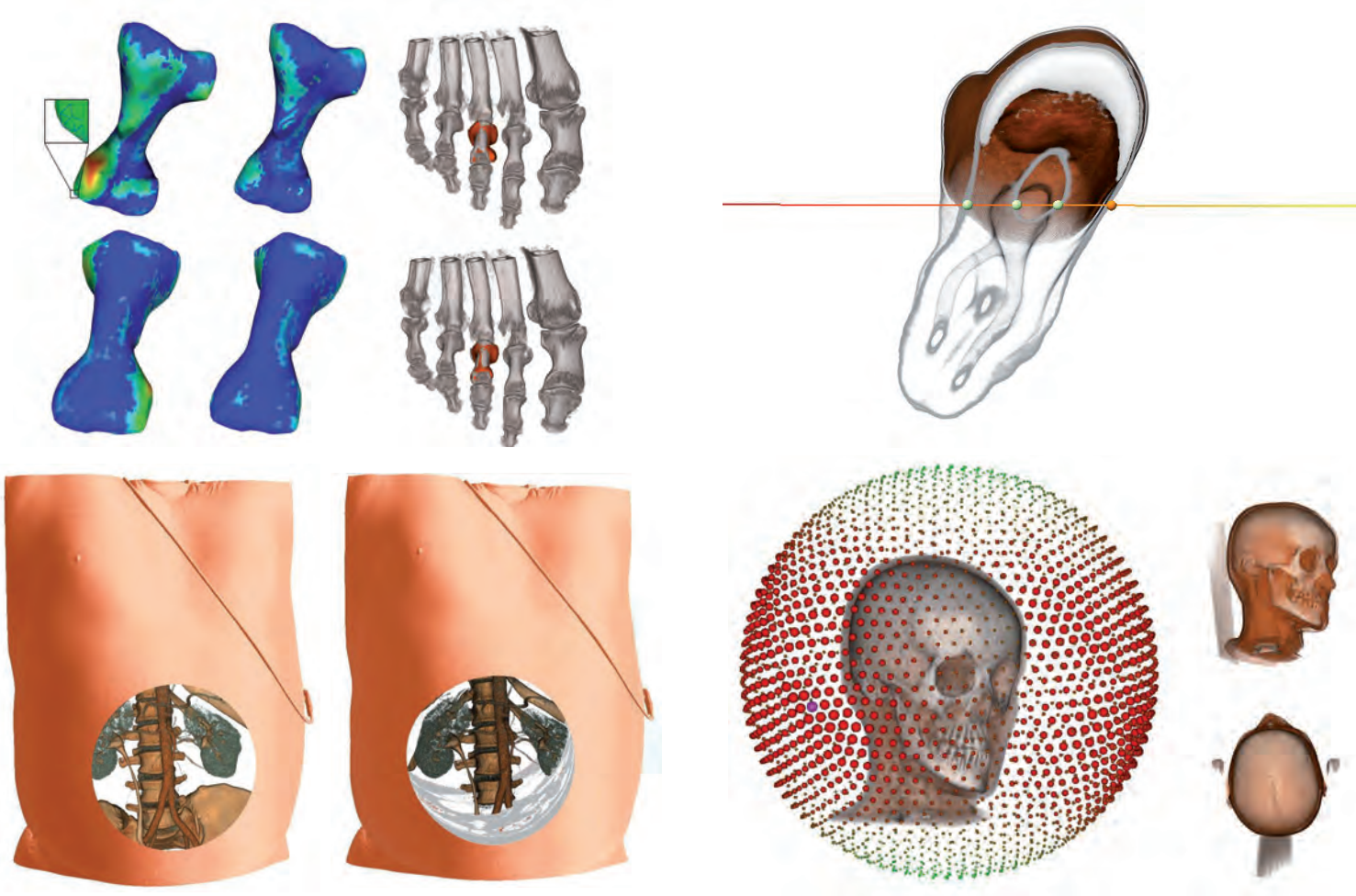
[WVFH12] A. Wiebel, F. M. Vos, D. Foerster, and H. Hege. WYSIWYP: What You See Is What You Pick. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2236–2244, 2012.

[WZMK05] L. Wang, Y. Zhao, K. Mueller, and A. Kaufman. The Magic Volume Lens: An interactive focus+context technique for volume rendering. In *Visualization, 2005. VIS 05. IEEE*, pages 367–374. IEEE Computer Society, 2005.

[WZZ+07] Y. Wang, D. Zhou, Y. Zheng, K. Wang, and T. Yang. Viewpoint selection using PSO algorithms for volume rendering. In *IMSCCS '07: Proceedings of the Second International Multi-Symposiums on Computer and Computational Sciences*, pages 286–291. IEEE Computer Society, 2007.

[ZAM11] Z. Zheng, N. Ahmed, and K. Mueller. iView: A feature clustering framework for suggesting informative views in volume visualization. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):1959–1968, 2011.

[ZHT02] J. Zhou, M. Hinz, and K. D. Tönnies. Focal region-guided feature-based volume rendering. In *International Symposium on 3D Data Processing Visualization and Transmission*, pages 87–90. IEEE Computer Society, 2002.



ViRVIG

UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH