Universidade Federal da Bahia
Universidade Estadual de Feira de Santana

DISSERTAÇÃO DE MESTRADO

A Markerless Augmented Reality Environment for On-Patient Medical Data Visualization

Márcio Cerqueira de Farias Macedo

Mestrado em Ciência da Computação – MMCC

Salvador
20 de Novembro de 2014

MMCC-Msc-0000

MÁRCIO CERQUEIRA DE FARIAS MACEDO

A MARKERLESS AUGMENTED REALITY ENVIRONMENT FOR ON-PATIENT MEDICAL DATA VISUALIZATION

Dissertação apresentada ao Mestrado em Ciência da Computação da Universidade Federal da Bahia e Universidade Estadual de Feira de Santana, como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Orientador: Antônio Lopes Apolinário Júnior

Salvador
20 de Novembro de 2014

Ficha catalográfica.

Macedo, Márcio Cerqueira de Farias

A Markerless Augmented Reality Environment for On-Patient Medical Data Visualization/ Márcio Cerqueira de Farias Macedo– Salvador, 20 de Novembro de 2014.

72p.: il.

Orientador: Antônio Lopes Apolinário Júnior.
Dissertação (mestrado)– Universidade Federal da Bahia, Instituto de Matemática, 20 de Novembro de 2014.

1. Realidade Aumentada. 2. Visualização Volumétrica. 3. Iluminação baseada em Imagem.
I. Apolinário, Antônio Lopes. II. Universidade Federal da Bahia. Instituto de Matemática. III Título.

CDD 20.ed.

TERMO DE APROVAÇÃO

MÁRCIO CERQUEIRA DE FARIAS MACEDO

A MARKERLESS AUGMENTED REALITY ENVIRONMENT FOR ON-PATIENT MEDICAL DATA VISUALIZATION

Esta dissertação foi julgada adequada à obtenção do título de Mestre em Ciência da Computação e aprovada em sua forma final pelo Mestrado em Ciência da Computação da UFBA-UEFS.

Salvador, 20 de Novembro de 2014

Prof. Dr. Antonio Lopes Apolinario Junior
Universidade Federal da Bahia

Profa. Dra. Veronica Teichrieb
Universidade Federal de Pernambuco

Prof. Dr. Vinicius Moreira Mello
Universidade Federal da Bahia

ACKNOWLEDGEMENTS

Here, I would like to thank everyone who has supported me during my master's degree.

First of all, I would like to thank my family. The first is the person that I love at most, my mother Vilma, who gave me the basis for what I am today, teaching me the right way to go when I get lost or confused. Looking at her life of hard work and overcoming I get the inspiration that I need everyday. Next, I would like to thank my grandmother Lucidalva, for giving me love and affection everytime I need. To my brother Danilo, for all the time we spent together when I am not studying. To my father Edson, for being on my side on the moments that I need his help. Finally, to the rest of my family, for all the moments of happiness lived together.

Related to my academic life, I would like to express my special gratitude to Prof. Antonio Carlos, who initiated my passion for augmented reality and is not only an academic colleague, but a personal friend, if not my "second father". I wish to thank Prof. Dr. Antônio Apolinário, for all the moments of insightful discussions. Many of the ideas which resulted on well-ranked papers came from his brilliant mind. I would like to thank Prof. Dr. Gilson Giraldi, for the detailed reviews and comments that he has provided for some papers we have published together. Finally, I would like to thank all my colleagues from Labrasoft, in special Prof. Luiz Machado, for sharing with me great experiences of applying augmented reality for education, and all my colleagues from the master's program.

I would like to thank all my personal friends, in special those whom I have played football, Mario Kart and a lot of other games and activities. Everyone helped me relaxing during these two years of intense study.

In terms of professional life, my special gratitude for everybody from Computação Brasil for all the laughs, "Counter-Strike" sessions of gameplay and soccer matches. In there, I have enjoyed a workspace of great atmosphere.

RESUMO

Visualização de dados médicos no paciente é desejável em muitas situações, como planejamento cirúrgico e treinamento médico. Atualmente, esta visualização é possível com o uso da realidade aumentada, uma tecnologia que habilita a visualização da anatomia virtual do paciente na localização da anatomia real em um *display* convencional. Nesta dissertação, é apresentado um ambiente de realidade aumentada sem marcadores para visualização de dados médicos no paciente com suporte ao fotorrealismo, a métodos sofisticados para composição de dados reais e virtuais e a características adicionais como oclusão. A partir de uma avaliação do ambiente proposto, os resultados obtidos mostram que o ambiente executa em tempo real e provê boa qualidade visual da cena aumentada.

Palavras-chave: Realidade Aumentada, Rastreamento sem marcadores, Visualização Volumétrica, Iluminação baseada em Imagem, Visualização por foco e contexto.

ABSTRACT

On-patient medical data visualization is desirable in several situations such as surgical planning and training. Currently, it becomes possible with the augmented reality, a technology which enables the visualization of the patient's virtual anatomy at the location of the real one in a conventional display. In this dissertation, we present a markerless augmented reality environment for on-patient medical data visualization which supports: photorealism, advanced methods for real and virtual data composition and additional features such as occlusion. From an evaluation of the proposed environment, the results obtained highlight that it runs in real-time and provides good visual quality of the augmented scene.

Keywords: Augmented Reality, Markerless Registration, Volume Rendering, Image-Based Lighting, Focus + Context Visualization.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

Physicians see medical data, typically patient's anatomical structures, on a monitor and they must analyze and mentally compose what is shown on the screen to the patient. This mental model of the patient's anatomy will serve as basis for health care in routine examinations or time-critical situations. Therefore, the physicians must have sufficient knowledge over patient's and human's anatomy to proceed appropriately during any operation (e.g. diagnosis, surgery). In this context, an important question arises: *what is the best way to visualize patient's relevant information so that physicians may be able to complete a medical procedure properly?* [Bichlmeier 2010].

With the availability of the augmented reality (AR) technology, one can take over this task of mental mapping by transferring it to a computer. Therefore, the physician is able to visualize, at the same time, the patient and a part of his anatomy. On-patient or *in-situ* medical data visualization can be used for surgical planning, training, medical diagnosis and post-operative examination.

Augmented Reality is a technology which augments the view of a real scene with additional virtual information. Medical AR is a sub-field of AR in which the virtual entity is a medical data. In an ideal scenario, a successful medical AR environment must support the following requirements:

1. Real-Time Performance - In computer graphics, an application is considered to be in real-time if it runs equal or above 15 frames per second [Akenine-Moller et al. 2002]. This concept is related to user interactivity, because the user must interact with the application and receive fast feedback (i.e. without too much delay);

2. Accurate Tracking - By registering two surfaces or textures captured at previous and current moments from the scene, the current camera pose can be estimated. Therefore, the virtual object can be positioned and tracked into the augmented scene giving the illusion that it coexists at the same location in the real world;

3. Volume Rendering - When the virtual entity consists of a 3D medical data, it requires a special attention to the way it will be rendered. Volume rendering is a field which is concerned with techniques for generating images for this kind of data, therefore being necessary for a medical AR application;

4. High Visual Quality - To improve user's perception of the augmented scene, real and virtual medical data must be rendered according to simple methods such as superimposition and blending or advanced methods such as focus + context visualization, depending on the medical AR application;

5. Photorealistic Rendering - To allow a seamlessly integration of the virtual medical data into the augmented scene, it must be rendered as realistic as possible;

Traditionally, AR applications use fiducial markers for tracking and polygonal models for rendering. However, in medical AR, fiducial markers can be intrusive in the scene, which motivates the development of markerless tracking methods. Moreover, volumetric models are commonly used for rendering due to the popularity of 3D scanners based on computed tomography (CT) and magnetic resonance imaging (MRI) for medical procedures. An AR environment proposed in the field of on-patient medical data visualization must take into consideration these additional features, still running in real-time on customer hardware.

To improve the visual understanding of the scene by the physician, it is fundamental for a medical AR application to provide high quality rendering of the virtual objects to be combined with the real scene. One solution for this question is to show patient's anatomic region of interest not as an isolated object, as done conventionally, but in the context of the patient's body. This not-so-new solution, formally known as focus + context (F+C) visualization paradigm [Card et al. 1999], has been studied and developed since many centuries ago. In 1510, Leonardo da Vinci drawn inner organs and muscles in the context of the human body. In 1543, Vesalius published the anatomy book *De Humani Corporis Fabrica*, in which bones, muscles, blood vessels among other structures are illustrated in the context of a human body. So, based on this paradigm, the physician may understand the relative positioning of patient's anatomy (focus region) while visualizing patient's body (context region) [Bichlmeier et al. 2007].

F+C visualization is one alternative to guarantee good visual quality for the AR application. Another alternatives such as volume clipping and illustrative context-preserving volume rendering [Bruckner et al. 2005, Bruckner et al. 2006] can be employed to allow a good composition between patient's anatomy and body as well. However, while these three paradigms enhance the visual quality of the application, they alone lack realistic rendering for the external part of the patient's anatomy. To solve this issue, there are solutions in the field of image-based lighting to compute local illumination components (e.g. diffuse and specular terms) from the real dynamic environmental lighting in real-time.

In the end, there are several methods developed in academy which can be applied for every step of the application, however none of the state-of-the-art works have developed an integrated solution for on-patient medical data visualization which supports all those requirements listed before. In this dissertation, we show that from the recent advances in hardware, as GPUs and 3D sensors, and the recent techniques proposed in academy, it is possible to create an application with real-time performance and good visual quality.

1.1 STATEMENT OF DISSERTATION

In this dissertation we present, to the best of our knowledge, the first markerless AR (MAR) environment for on-patient medical data visualization based on photorealistic local illumination with support to F+C visualization. To achieve this objective, we propose a novel integrated solution for MAR environment, shown in Figure 1.1. Markerless tracking is achieved by generating a 3D reference model of the patient's region of interest (ROI) and tracking it during the live stream. 3D reference model reconstruction requires markerless tracking to align the different viewpoints acquired from patient's ROI. Photorealistic rendering is achieved by computing spherical harmonics (SH) coefficients from a light probe image and by using them to extract diffuse and specular terms from the real environmental lighting. From the estimated camera pose, medical volume can be rendered and displayed to a physician inside patient's anatomy at the location of the real one. Volumetric medical data is rendered according to standard volume rendering techniques and real-world local illumination components are used to lit the medical volume in real-time. After volume rendering, F+C visualization techniques are used to define which parts of the volume will be visualized in the final augmented scene. Real-time performance is achieved by implementing the markerless tracking and 3D reference model reconstruction on the graphics processing unit (GPU) and by using an optimized implementation for photorealistic local components estimation in central processing unit (CPU). Moreover, to improve user's perception of the augmented scene, the application supports occlusion.

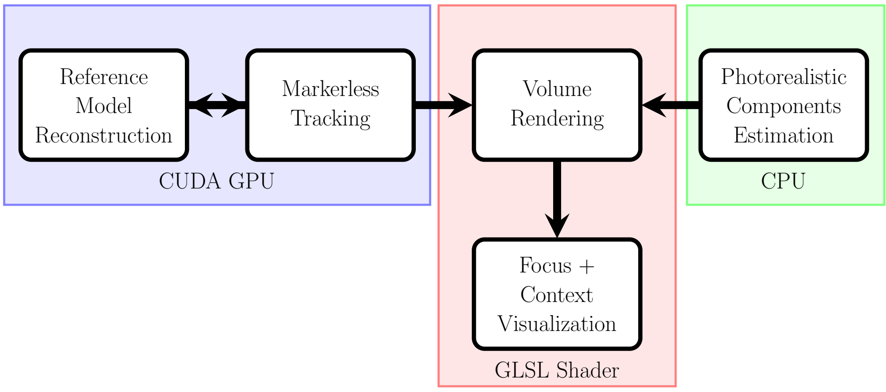


Figure 1.1 Integrated solution for photorealistic markerless on-patient medical data visualization viewed as components and their relationships.

The focus of this dissertation is to evaluate the proposed solution in terms of performance, visual quality and accuracy.

1.2 CONTRIBUTIONS

During the development of the proposed solution, some contributions have been made, as it will be briefly described in the next subsections.

1.2.1 Markerless Medical Augmented Reality Environment

The first contribution of this dissertation is the proposition of a novel medical MAR environment for on-patient medical data visualization. We have designed a new markerless tracking solution which is robust to rigid tracking failures. In a global analysis, we also have a contribution for designing a novel integrated solution for medical AR, as shown in Figure 1.1.

1.2.2 Focus + Context Visualization based on Volume Clipping

Instead of superimposing the virtual medical data onto the patient, a better solution is to show the patient's anatomy as a focus region in the context of the patient's body. This process is known as F+C visualization paradigm. In this context, we take advantage from the effect of volume clipping to propose three new F+C visualization techniques: smooth contours, visible background on CT data and visible background on MRI data.

1.2.3 On-Patient Medical Data Visualization based on Photorealistic Rendering

F+C visualization alone lacks realistic rendering. Therefore, we propose the first application for on-patient medical data visualization which supports photorealistic volume rendering based on local illumination components.

1.3 ORGANIZATION

The remaining of this dissertation is organized as follows:

Chapter 2 gives an overview on relevant work in the fields of augmented reality, focus + context visualization, on-patient medical data visualization and image-based lighting. A review on recent related work and a discussion of basic concepts for each one of those fields are presented.

Chapter 3 presents the markerless tracking solution proposed in this dissertation. Methods for 3D reference model reconstruction and live tracking are described. An evaluation is conducted to validate the approach in terms of performance, accuracy and robustness. Moreover, the use of this markerless algorithm for a typical polygon-based AR application is presented.

Chapter 4 presents the MAR environment for on-patient medical data visualization, describing how the medical volumes can be rendered efficiently and with high quality, integrated into the augmented scene. Also, a semi-automatic medical volume-to-patient registration, which takes advantage from our markerless tracking basis representation, is shown. In this context, the F+C visualization techniques employed are presented. The environment is evaluated in terms of performance and visual quality.

Chapter 5 shows how photorealistic volume rendering can be done based on real local illumination components. Performance and visual quality evaluations are performed in order to validate the proposed solution.

This dissertation is concluded in Chapter 6, where a discussion of future work and our final considerations are presented as well.

Chapter 2

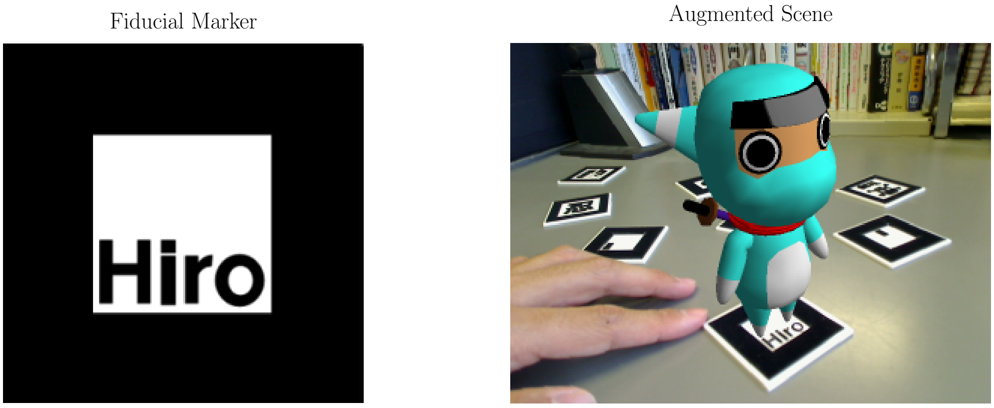
SURVEY OF RELEVANT CONTRIBUTIONS

The techniques described in this dissertation are mainly distributed over four fields of Computer Science: augmented reality, focus + context visualization, on-patient medical data visualization and image-based lighting. It is fundamental to know the basic concepts of these fields to fully understand the techniques that will be described along this document and the issues they overcome. In this chapter, we give an historical overview of AR, focusing on tracking methods, F+C visualization applied in the fields of AR and on-patient medical data visualization, and image-based lighting in order to give a comprehensive description of the main achievements and challenges faced by researchers so far. Moreover, a review on recent related work specifically for the field of on-patient medical data visualization is performed in order to highlight the advantages and disadvantages of the proposed approach over the state-of-the-art ones found on the literature.

2.1 AUGMENTED REALITY

Augmented Reality is a field of Computer Science in which the user's view of a real scene is augmented with virtual information. Accurate tracking, or camera pose estimation, is required for the proper registration of virtual objects. Real-time performance is needed to maintain user's interactivity with the application. Photorealistic rendering and occlusion support are necessary to allow a realistic composition of the virtual object into the real scene. Since this concept has become popular by the seminal works of Ronald Azuma and colleagues [Azuma 1995, Azuma 1997, Azuma et al. 2001], many researches have been conducted focusing on how to solve the problem of accurate and real-time tracking, which is one of the main technical challenges of AR.

Since the end of 90s, the main method to position a virtual object and track a real scene in real-time relies on the use of an artificial fiducial marker. The squared shape of the marker allows its fast recognition and tracking, while the symbol inside it is used as an marker ID, which identifies the virtual object that must be rendered (Figure 2.1). With the advance in terms of computational resources for customer hardware, many techniques have been proposed to allow the use of any object visible in the scene as a natural marker. In these cases, tracking is based on the selection of keypoints (e.g. pixels with high curvature or localized on corners) or edges extracted from the color image. Due to its simplicity, ARToolKit was one of the libraries most used to help on the creation of AR applications, even if it uses the traditional fiducial marker as basis for tracking [Kato and Billinghurst 1999]. Recently, some extensions for this library have been proposed, such as: FLARToolKit [Koyama 2009], FLARAS [Souza et al. 2012], SACRA [Santin 2008] among others.


Fiducial Marker

Augmented Scene

Figure 2.1 Marker-based tracking. Fiducial Marker image is courtesy of *http://www.mquter.qut.edu.au/*. Augmented Scene image is courtesy of *http://kougaku-navi.net/*.

In the simplest case, an AR application requires only a computer equipped with a webcam and a fiducial marker. Therefore, tracking must operate only on the image space, according to the features of the pixels. Several methods have been proposed to solve this problem. The most common ones are based on the principle of optical flow [Lucas and Kanade 1981, Horn and Schunck 1981]. The main drawbacks for this color or texture-based tracking are the susceptibility to illumination conditions and marker occlusion, which may affect the accuracy of the tracking algorithm. A survey on this subject can be found in [Teichrieb et al. 2007].

In parallel to this scenario, a new hardware has gained popularity due to its robustness and grown accessibility, the 3D depth sensor. By using a technique to compute the distance between the objects present in the scene to the sensor, with this hardware it becomes possible to capture the depth map: an image which store for each pixel the depth value associated to the corresponding 3D point for a given scene.

Depth sensors can be classified into active or passive. Active sensors provide and control their own illumination; Passive sensors only absorb environmental radiation and extract depth information from it [Young 1994].

One of the most known methods to compute depth for passive sensors is the stereo vision, which simulates the process of depth measurement done by the human vision system [Marr and Poggio 1979]. For active sensors, the most used methods are: Time-of-Flight (ToF) and structured light pattern. For the ToF technology, scene's depth is computed through the time difference between emission and reception of an infrared (IR) wave sent from an IR emitter [Hansard et al. 2013]. For structured light technology, depth maps are computed through the triangulation done between a light emitter and a sensor. A known light pattern is emitted on the scene and then captured by the sensor. Depth information is retrieved from an analysis on the distortion of this pattern [Besl 1988, Posdamer and Altschuler 1982].

With the recent availability of low-cost real-time depth sensors, the use of depth-based tracking algorithms has been made popular. This comes from the fact that they are robust to the variation of illumination conditions, as they are based on the depth of the scene instead of its color. Also, they operate over real depth values measured from the scene, which helps on object positioning and occlusion support as well. These methods depend on the presence of geometric information on the scene and have a limitation in terms of range of depth measurement, being inadequate for tracking in outdoor environments (e.g. streets) [Whelan et al. 2012, Peasley and Birchfield 2013]. Nevertheless, the use of this technology is suitable to the scope of this dissertation, as the human body is not symmetric, situation in which there is not much variation on depth values captured by the sensor. Moreover, on-patient medical data visualization applications are intended to be used for indoor scenes (e.g. surgical rooms).

From the high performance and accuracy of the existing depth-based tracking algorithms, the use of fiducial markers for positioning of the virtual object and tracking of the real scene is no longer essential for AR applications. In this way, AR can be markerless, where a part of the real scene is used as natural marker instead of using an artificial fiducial marker. In this scenario, tracking becomes more complex, however, as there are not intrusive markers that are not part of the original scene, markerless AR is desirable in several situations such as the ones for on-patient medical data visualization. The algorithm most used for depth-based tracking is the real-time variant of the Iterative Closest Point (ICP) algorithm [Rusinkiewicz and Levoy 2001], which has already GPU implementation [Izadi et al. 2011].

2.2 FOCUS + CONTEXT VISUALIZATION

Some applications for AR, such as the ones for on-patient medical data visualization, demand a special attention to the composition between virtual and real entities of the AR environment. Recently, many approaches have been proposed in the field of F+C visualization to dynamically define how this composition will be done. Those techniques are also known as ghosting or x-ray vision techniques (Figure 2.2) and all of them share the concept of importance or saliency map, a mask which controls how real and virtual entities should be blended, similarly to an alpha mask.

Sandor et al. designed a method for importance map computation based on characteristic regions of both real and virtual objects inspired by three features [Sandor et al. 2010]. They are: Luminosity - to preserve regions with high illumination; Hue - to preserve strong colors and Motion - to preserve moving structures in the final rendering. This algorithm runs on shader, increasing little time for the application. As stated by the authors, this work was an extension of [Avery et al. 2009] which is based on edge overlay to improve spatial perception.

Mendez et al. proposed a F+C technique in which lightness and color contrast for a

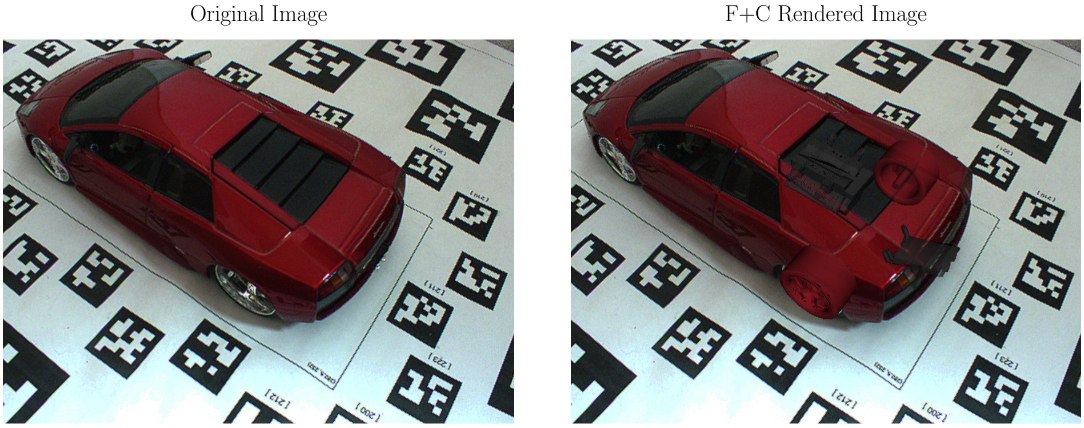


Figure 2.2 In a naive rendering, only the occluder is rendered for an augmented image. By using a ghosting technique, the occludee parts of the model are considered for the final rendering. In this case, they are not simply overlaid over the original image, but inserted as a focus region in the context of the resulting augmented scene. Images courtesy of [Kalkofen et al. 2013].

given image are modified according to the importance map computed from a live color video [Mendez et al. 2010]. By adding subtle changes on the image, they guarantee temporal and spatial coherence between frames. The problem with this approach is its performance, which does not achieve full 30 frames per second (FPS) even implemented in GPU.

An adaptive F+C visualization technique was recently introduced by Kalkofen et al. [Kalkofen et al. 2013]. In their approach, an importance map is computed for the occluder [Mendez et al. 2010] and the occludee is inserted into the scene. Then, another importance map is computed for this resulting image and then compared against the first map computed. The regions on the first importance map that are not present in the final rendering are then emphasized to be visible. This approach improves the visual quality of the augmented scene and it runs in real-time. However, it is not suitable for MAR environments, as it alone requires $33ms$ of processing time. Therefore, this additional time would decrease severely a MAR application performance.

F+C rendering was also proposed for visualization of underground structures in a street scene [Schall et al. 2009, Chen et al. 2010, Padilha et al. 2013]. In these approaches, a method is used to dynamically compute from still color images when underground structures must be rendered in relation to moving objects present in the scene. While the final visual quality is good, the performance of existing techniques is not full real-time.

Traditional methods which compute importance maps from live color video of the real scene are prone to errors because they are dependent on illumination and material properties of the real environment. To overcome these problems, Mendez and Schmalstieg proposed a method to compute an importance mask based on the 3D model of the scene [Mendez and Schmalstieg 2009]. It is done by using techniques such as mesh saliency [Lee et al. 2005] or through user interaction in a preprocessing step. The problem associated with this approach is that the importance mask creation requires some processing time. Therefore, the user cannot change interactively the importance mask during an AR live stream.

The methods for F+C visualization proposed in the literature for general AR application capture the features of the image, however their importance maps are not accurate enough to be used for medical applications.

F+C visualization has not been only proposed for AR, but also for volume rendering to define how internal structures of the volume (e.g. bone, organ) should be visualized in the context of the volume's soft tissue (Figure 2.3).

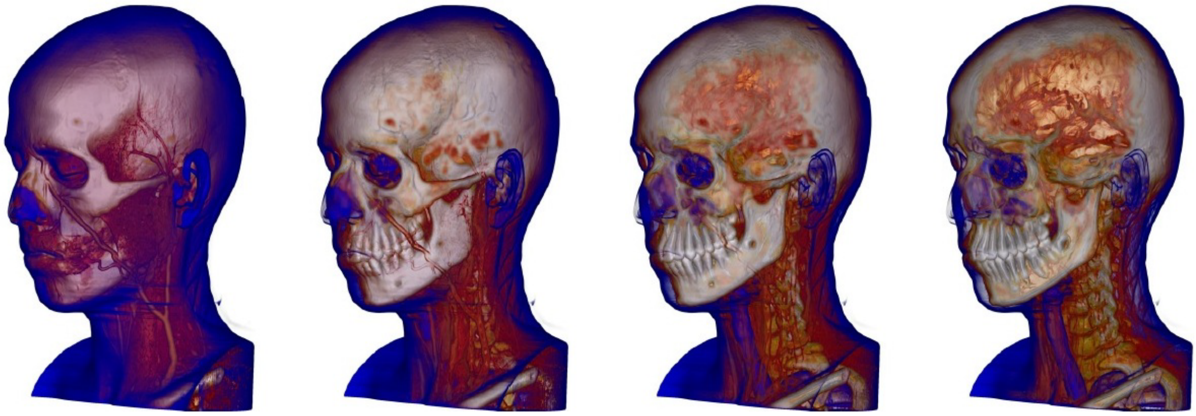


Figure 2.3 Focus + context visualization applied for volume rendering. Images courtesy of [Bruckner et al. 2005].

Brucker et al. proposed a novel algorithm for context-preserving volume rendering [Bruckner et al. 2005, Bruckner et al. 2006]. From factors such as: shading intensity, gradient magnitude, distance to the eye point and previously accumulated opacity, the method allows the user definition of the F+C rendering according to only two parameters which controls these four factors to interactively change the transparency level between internal and external structures of the volume. The technique is easy to implement and runs directly on the shader. An extension of this algorithm was proposed in [Sikachev et al. 2010]. It incorporates rotation, scale, position and mouse click to dynamically select focus and context regions.

Kruger et al. proposed the ClearView [Kruger et al. 2006]. Four layers (i.e. focus and context structures, isosurface's normal and curvature) are generated and composed for each frame in order to define the final visualization. The main disadvantages for this method are: it is naturally multi-pass (i.e. one shading pass is required to compute every layer) and the layers must be recomputed for every viewpoint changing. Therefore, the approach has a considerable cost in terms of performance.

Kirmizibayrak et al. proposed a volumetric brush method for interactive definition of focus and context regions in volumetric models [Kirmizibayrak et al. 2014]. The approach runs in real-time and provides a good alternative to traditional ways to visualize medical data for physicians, specially for applications such as radiation therapy.

F+C solutions for specific scenarios have also been proposed in the literature to solve: complex fiber distributions [Röttger et al. 2012], blood flow [Gasteiger et al. 2011], structured biomedical data [Abellan et al. 2008] and ultrasound [Schulte zu Berge et al. 2014] to cite a few recent work.

The first technique proposed for F+C visualization in the field of on-patient medical data visualization was the Contextual Anatomic Mimesis proposed by Bichlmeier et al. [Bichlmeier et al. 2007]. Its importance map is defined by three parameters: curvature of patient's skin surface, angle of incidence factor (i.e. angle between the normal on the skin surface and a vector pointing from the position of the surface and the eye) and distance falloff (i.e. distance between each point on the surface and the intersection point of the line of sight and the skin surface). Different from the color-based methods mentioned for the F+C techniques used in the AR scenario, this one operates directly on the shader and does not depend on illumination or texture for the importance map definition. Although it provides improved perception of the 3D medical data into the scene, it does not give special attention for some volume rendering techniques, such as volume clipping.

2.3 ON-PATIENT MEDICAL DATA VISUALIZATION

Medical AR systems for on-patient medical data visualization have been driven by different approaches in recent years. In this section, we classify the approaches based on their tracking technology. A comparison between our approach and the ones proposed in literature is done in the last subsection.

2.3.1 Marker-Based

Kutter et al. proposed a marker-based method for real-time on-patient visualization of volumetric medical data on a Head Mounted Display (HMD) [Kutter et al. 2008]. Their work focuses on efficient implementations for high quality volume rendering in an AR environment. They also provide occlusion handling for physician hands. An improved version of this work was proposed by Wieczorek et al. to handle with occlusions due to medical instruments as well [Wieczorek et al. 2010]. Also, they included additional effects in the system, such as virtual mirror and multi-planar reformations.

Debarba et al. proposed a method to visualize anatomic hepactetomy (i.e. anatomic liver resections) in an AR environment [Debarba et al. 2012]. The use of a fiducial marker made possible the positioning and tracking of the medical data in the scene. A mobile device was used to allow the visualization of internal structures of the patient's body.

Lee et al. proposed a registration framework for a medical AR system [Lee et al. 2012]. They used three cameras: two of them are mounted to form a stereo vision system and reconstruct patient's head; the other camera is used to capture the images of the patient in real-time. In a pre-processing step, a surface is reconstructed from CT and a variant of the ICP algorithm is used to do the image-to-patient registration. The estimation of the third camera pose is done by using a fiducial marker.

Suenaga et al. proposed a method for on-patient medical data visualization of maxillofacial regions [Suenaga et al. 2013]. A 3D optical tracking system and a fiducial marker are used to track the patient. A semi-transparent display is placed in front of the mouth region of the patient. The display shows the maxillofacial medical data. This method runs in only 5 FPS.

2.3.2 Markerless

Maier-Hein et al. proposed a method for mobile MAR for on-patient visualization of medical images [Maier-Hein et al. 2011]. They proposed a system in which a ToF camera is mounted on a mobile and portable device (e.g. tablet, iPad) and the physician might move the portable device along the body of the patient to see his anatomical information. To estimate the camera pose, they used a graph matching algorithm [dos Santos et al. 2010] and a anisotropic variant of the ICP algorithm [Maier-Hein et al. 2010] to align the surfaces continously captured by the ToF camera. This method runs in 10 FPS.

In the field of anatomic education, Blum et al. proposed the *mirracle*, a magic mirror for teaching anatomy [Blum et al. 2012]. They used a display device and a Kinect sensor to allow volume visualization of a CT dataset augmented onto the user. To track the pose of the user, they used a skeleton tracking algorithm. As the system is designed for educational purposes, they could use a generic CT volume which was scaled to the size of the user and augmented onto him. Based on the assumption that the Kinect provides inaccurate skeleton tracking, Meng et al. proposed an extension to *mirracle* in which landmarks are used to improve the accuracy of the positioning and tracking of the medical data [Meng et al. 2013]. A recent extension of this application for bone anatomy learning has been proposed in [Stefan et al. 2014].

Mercier-Ganady et al. presented a novel MAR application for on-patient brain activity visualization [Mercier-Ganady et al. 2014]. User's head is tracked by using a face tracking algorithm native from Microsoft Kinect SDK [Microsoft 2014]. Brain activity is computed from an electroencephalography cap (EEG), which is worn by the user.

Kilgus et al. proposed a mobile MAR environment for on-patient medical data visualization to support forensic pathologists during autopsy [Kilgus et al. 2014]. A structured light sensor is mounted on a mobile device, similarly as done in [Maier-Hein et al. 2011], and it is used to capture color and depth sensor of the body being analyzed. These data are sent to a server which estimates the current camera pose, projecting the color information of the medical data at the position of the body.

2.3.3 Discussion

The approaches described in Section 2.3.1 share the same drawback: they use markers to help in the calibration, positioning and tracking of the objects in the scene. The use of fiducial markers provides fast and accurate tracking. However, these markers are still intrusive, because they are not part of the original scene. Moreover, the hardware of the optical tracking system in some applications is too expensive. The approaches that do not use such marker-based hardware, in general, do not obtain real-time performance in its application due to the computational cost of the markerless tracking in conjunction with the volume rendering techniques used, which require additional processing time in comparison with polygonal model rendering. One exception of this is the *mirracle* system and its extensions. However, the main drawback of the skeleton tracking used is that it does not track accurately some parts of the body, such as head. Another exception is the approach proposed for brain activity visualization. However, the authors have used a specific technique for face tracking. Different from them, our approach based on markerless tracking runs entirely in real-time with low-cost hardware components. Although the solution proposed in this dissertation is evaluated only for a scenario where the region of interest is a head, we believe that with minor adaptations the method would be general in the sense that it would track any part of the body with enough accuracy, as its basis representation is flexible for such generality as proven in the original papers [Rusinkiewicz and Levoy 2001, Izadi et al. 2011]. Moreover, in terms of visual quality, we integrate into our approach state-of-the-art techniques proposed in this field [Bichlmeier et al. 2007] to enhance the augmented visualization of the virtual internal structures on the patient, instead of superimposing the virtual content on the scene.

Method	**MAR**	**RTP**	**F+C**	**OCC**	**PR**	**ACC**
Kutter et al. [2008]	No	Yes	Yes	Yes	No	Yes
Wieczorek et al. [2010]	No	Yes	Yes	Yes	No	Yes
Debarba et al. [2012]	No	Yes	No	No	No	No
Lee et al. [2012]	No	Yes	No	No	No	No
Suenaga et al. [2013]	No	No	No	No	No	Yes
Maier-Hein et al. [2011]	Yes	No	No	No	No	No
Blum et al. [2012]	Yes	Yes	Yes	No	No	No
Meng et al. [2013]	Yes	Yes	Yes	No	No	No
Stefan et al. [2014]	Yes	Yes	Yes	No	No	No
Mercier-Ganady et al. [2014]	Yes	Yes	No	No	No	No
Kilgus et al. [2014]	Yes	No	No	No	No	No
Proposed method	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**No**

Table 2.1 Classification of on-patient medical data visualization methods proposed in the literature according to the following attributes: markerless augmented reality environment without any artificial marker (MAR); full real-time performance at 30 FPS (RTP); support for F+C visualization (F+C); Occlusion handling (OCC); Photorealistic volume rendering (PR); and accuracy at surgery level ($< 1mm$) (ACC). Information extracted from original papers and supplementary materials when available.

A general classification of the state-of-the-art methods presented in this section including ours can be seen in Table 2.1. A few of them support occlusion and only ours supports photorealistic volume rendering. The only drawback of our approach is accuracy, as it still cannot be used for surgical environments. In this sense, only three approaches provide high accuracy for the application, however none of them are markerless, nor support photorealistic rendering.

2.4 IMAGE-BASED LIGHTING

Photorealistic rendering is a field which has been driven by several approaches in the literature [Karsch et al. 2011, Knecht et al. 2012, Karsch et al. 2014]. The techniques proposed in this field are specially important for AR, as they promote a seamlessly integration of the virtual object into the augmented scene. In this scenario, it is fundamental to capture real-world lighting coefficients, such as dominant lighting color and direction, to lit the virtual object so that it will be consistent with the illumination of the real scene (Figure 2.4).

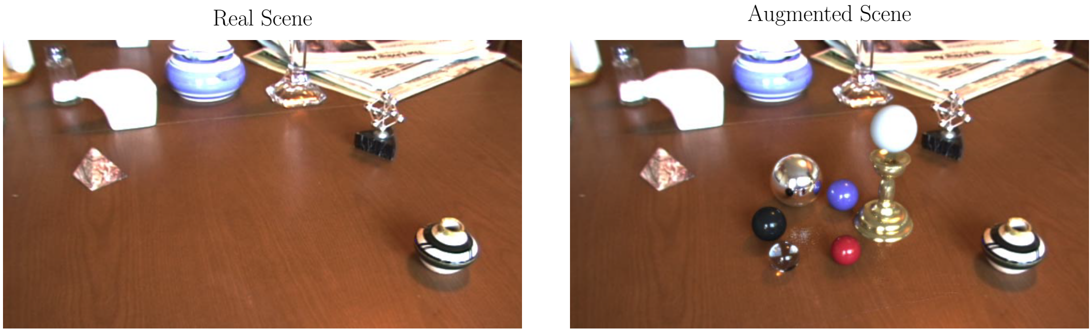


Figure 2.4 Seamlessly integration of virtual objects into a real scene through photorealistic rendering. Images courtesy of [Debevec 1998].

Techniques for solving the problem of real-world lighting estimation have been proposed in the field of image-based lighting. One of the first solutions was proposed in the seminal work of [Debevec 1998], where a high dynamic range (HDR) image captured previously is used to lit the virtual objects. The main difference between the low and high dynamic range images relies on its range of storing. Low dynamic range (LDR) image is the kind of image captured, processed and stored from customer cameras. As it is already known, in general it is limited to store a range of 255 values for each color channel (i.e. red, green and blue) of the image. On the contrary, HDR images cover the true radiance values for a given scene. Therefore, depending on the encoding method used, we may have high dynamic range up to 79 orders of magnitude to represent more precisely the radiance values captured from the scene [Reinhard et al. 2005].

The traditional ways to capture HDR images are by using a HDR-video camera or by capturing and merging multiple LDR photographs of a light probe (i.e. mirrored sphere) image taken with different exposures from the real scene [Debevec and Malik 1997]. As HDR-video cameras are still expensive nowadays, the method for LDR merging is the most common for creating HDR content. Accordingly, many applications have used HDR images for photorealistic AR [Meilland et al. 2013, Pessoa et al. 2010, Pessoa et al. 2012]. However, as the HDR content was previously captured based on a static scene, it does not give support for real-time lighting estimation in dynamic scenes, where lighting condition changes on the environment. To solve this issue, a solution is to capture the LDR image from the light probe and compute the lighting coefficients from such image in real-time. This solution implies in loss of visual quality in the final rendering, but introduces the support for dynamic lighting conditions. Also, the result can be improved by using available algorithms to compute HDR images from still LDR [Banterle et al. 2013, Kovaleski and Oliveira 2014] increasing the computational cost of the application.

To compute the lighting coefficients in real-time from a light probe image, SH basis

[Sloan 2009] is commonly used due to its compact and efficient representation. In this sense, some techniques have already been proposed taking one step further in this representation to compute local illumination components in real-time based on only nine SH coefficients [Ramamoorthi and Hanrahan 2001, Nowrouzezahrai et al. 2011]. However, such works were originally proposed for relighting of polygonal models and were never applied in the context of an application for on-patient medical data visualization.

2.5 SUMMARY

In this chapter we have presented the fundaments for augmented reality, focus + context visualization and image-based lighting. Moreover, recent related work in the field of on-patient medical data visualization were presented and compared against our approach.

Markerless tracking is desirable for on-patient medical data visualization because it does not introduce intrusiveness in the scene. Also, software and hardware technologies evolved such that this kind of tracking can be used for augmented reality in real-time and with high accuracy, although there is not current markerless technology which provides accuracy at surgery level.

Focus + context visualization is a concept which has been used in several fields of Computer Science, including augmented reality, volume rendering and specifically on-patient medical data visualization. Several real-time techniques have been proposed so far, but most of them cannot be used for medical applications because they are not accurate enough and the time they require to be processed makes their use unsuitable for markerless applications.

To favor a realistic composition of the virtual object into the augmented scene, photorealistic rendering based on local illumination components must be employed. Real-time solutions for this scenario have been proposed in the field of image-based lighting. However, to support dynamic environmental lighting estimation, the techniques exchange high quality for the final rendering by low cost in terms of performance.

Many approaches have been proposed to solve the problem of on-patient medical data visualization. However, none of them support photorealistic volume rendering and a few support occlusion. These are some of the issues that our work overcomes while still running in real-time and providing high-quality final rendering for the application. To do so, many contributions have been proposed and are described in the next chapters of this dissertation.

Chapter

3

MARKERLESS TRACKING

In this chapter we present the markerless tracking solution proposed in this dissertation. This chapter covers the methods proposed on the following authored publications [Macedo et al. 2013a, Macedo et al. 2013b, Macedo et al. 2013c, Macedo et al. 2014c], [Macedo et al. 2014b]. An overview of the proposed markerless tracking solution in an AR environment can be seen in Figure 3.1.

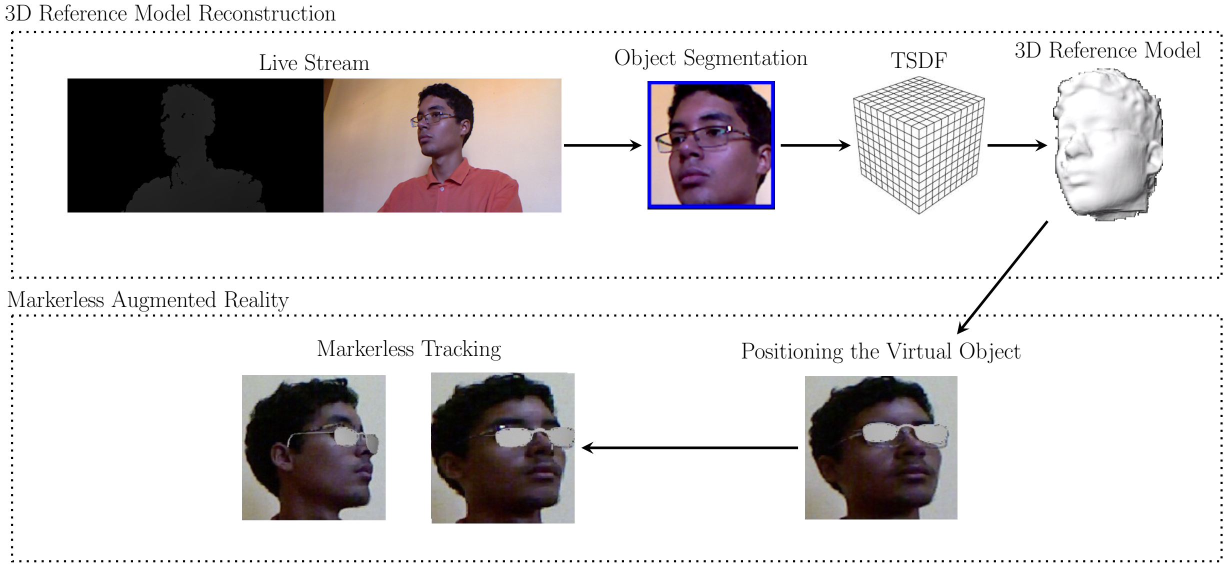


Figure 3.1 Overview of the proposed markerless solution.

3.1 VERTEX AND NORMAL MAP GENERATION

In this section we describe the algorithms that are used for every input frame to generate vertex and normal maps, with the exception of the segmentation procedure to extract patient's ROI, which is only performed during the reconstruction stage. All these algorithms run on the GPU. We call vertex map an organized point cloud which resembles an organized image (or matrix) in terms of structure, being the data organized in rows and columns. Such representation is common for point clouds directly captured from depth sensors. An organized dataset helps on the processing of nearest neighbour operations [Rusu and Cousins 2011, Izadi et al. 2011]. Inspired by the same motivation, we refer to normal map as an organized matrix which stores the normal vector for each vertex in the vertex map.

The proposed approach is based on an RGB-D sensor and a computer with GPU, therefore being accessible to any user. We have chosen to use the Kinect as RGB-D sensor to capture patient's color and depth information. Real-time performance is achieved by using parallel processing power of the GPU in all critical algorithms.

The Kinect has two sensors that capture color (I) and depth (D) information of the scene [Cruz et al. 2012]. I is a color image which stores for each pixel three color channels corresponding to red, blue and green intensities measured from the scene. D is a depth map which stores for each pixel a single channel corresponding to depth. To enable a mapping between these two images, intrinsic and extrinsic calibrations are performed. For convenience, we simply define this process as:

$$I = T_{cal} D \tag{3.1}$$

where T_{cal} is the transformation that maps D to I.

By calibrating the sensor, we have, in fact, a four-dimensional image which stores color and depth information from the scene.

In order to segment patient's ROI from the scene, as our ROI consists in patient's head, we apply the Viola-Jones face detector implemented in GPU to locate and segment the face in I (Figure 3.1). Viola and Jones described a method for robust real-time face detection in color images [Viola and Jones 2004]. The key contribution to achieve this goal is the use of a representation called integral image to compute Haar-like features quickly. In this representation, each pixel contains the sum of the pixels above and to the left of the original position. Once the features are computed, a combination of simple classifiers built using the Adaboost learning algorithm [Freund and Schapire 1995] is employed to detect the face regions.

From the segmented ROI and by considering that Equation 3.1 has already been applied, the segmented location can be mapped from I to D. Doing so, we achieve a more restricted area of D to be used through the other steps of our approach.

The Kinect sensor is known to provide noisy depth data. To deal with this issue, D is denoised by using a bilateral filter [Tomasi and Manduchi 1998], as shown in Equation 3.2 and 3.3. In this filtering technique, discontinuities of the raw depth map are preserved by using a nonlinear combination of nearby image intensities based on geometric proximity and photometric similarity.

$$D_f(u,v) = \frac{\sum\limits_{i=u-m}^{i=u+m}\sum\limits_{j=v-n}^{j=v+n} D(u,v)w(u,v,i,j)}{\sum\limits_{i=u-m}^{i=u+m}\sum\limits_{j=v-n}^{j=v+n} w(u,v,i,j)} \tag{3.2}$$

w is defined by:

$$w(u,v,i,j) = \text{expf}(-\frac{(u-i)^2+(v-j)^2}{2\sigma_d^2} - \frac{||D(u,v)-D(i,j)||^2}{2\sigma_c^2}) \tag{3.3}$$

$D(u,v)$ returns the depth intensity for the pixel located at image coordinates u and v and D_f is the filtered depth map. From empirical tests, we have set $m = 3$, $n = 3$, $\sigma_c = 30$ and $\sigma_d = 4.5$.

Depth of the background scene is segmented by applying a Z-axis threshold on D_f. We define this threshold initially $1.3m$, as related work [Fanelli et al. 2011], but this value can be changed by the user.

D_f is then converted into a vertex (V) and a normal (N) map. V can be computed by the following product:

$$V(u,v) = Df(u,v)T_{int}^{-1}[u,v,1]^T \tag{3.4}$$

where the superscript ^{T} denotes transposition and T_{int} is the Kinect depth sensor's intrinsic calibration matrix, which can be defined as:

$$T_{int} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}$$

where $f_x = f_y = 525$, $c_x = 320$ and $c_y = 240$ for the Kinect sensor [Burrus 2014].

Normal map is constructed by computing the eigenvector of smallest eigenvalue of the local covariance matrix computed for every vertex. This technique produces normal maps with less error than traditional approaches based on neighboring points for normal estimation [Holzer et al. 2012].

3.2 3D REFERENCE MODEL RECONSTRUCTION

With the patient's ROI properly segmented, we need to track it through the RGB-D sensor's live stream. We could track it by using the current and previous depth data directly coming from the depth sensor. However, we have chosen to integrate the different viewpoints acquired from the scene into a single reference model (Figure 3.1) and use this model as basis for registration, in order to improve tracking accuracy [Izadi et al. 2011].

KinectFusion algorithm is used in this context to reconstruct a 3D reference model of patient's ROI in real-time [Izadi et al. 2011]. This algorithm integrates raw depth data from a Kinect into a volumetric grid to produce a high-quality 3D reconstruction of a scene. The grid stores at each voxel the distance to the closest surface (i.e. Signed

Distance Function - SDF) [Osher and Fedkiw 2002] and a weight that indicates uncertainty of the surface measurement. For a single view, SDF values are positive in-front of the surface, negative behind and zero-crossing where the sign changes. In KinectFusion algorithm, SDF is only stored in a narrow region around the surface, which means that just a truncated SDF (TSDF) is stored. These volumetric representation and integration are based on VRIP algorithm [Curless and Levoy 1996]. Surface extraction is achieved by detecting zero-crossings through a ray caster. All these operations run on the GPU [Izadi et al. 2011].

3D reference model reconstruction is done only on the 3D reconstruction stage and is the basis for MAR live tracking. The reconstruction is stopped by the user and the virtual data can be positioned (Figure 3.1).

3.3 LIVE TRACKING

After virtual data positioning, MAR live tracking is started (Figure 3.1). In fact, live tracking is done in two steps: during reconstruction of the 3D reference model and during MAR with the user and the virtual data. We use a real-time variant of the Iterative Closest Point (ICP) algorithm [Rusinkiewicz and Levoy 2001]. The main goal of the ICP algorithm is to align two surfaces, represented as point clouds, given a good initial guess. It was originally proposed by [Besl and McKay 1992, Chen and Medioni 1992], adapted for real-time in [Rusinkiewicz and Levoy 2001] and finally implemented in GPU in [Izadi et al. 2011]. In our approach, the ICP algorithm is used to estimate the transformation $T_{live} = [R_{live}|t_{live}]$ that aligns the current point cloud (V_{curr}) captured by the Kinect sensor with the previous one (V_{prev}) extracted from the 3D reference model. The ICP variant used in our approach consists of the following six stages:

- **Selection of Points**: Points from V_{curr} and V_{prev} are chosen as samples to guide optimization to estimate the best rigid transformation. All the points visible ($D(u, v) > 0$) are selected for optimization;

- **Matching of Points**: Corresponding points between V_{curr} and V_{prev} are associated. It is used the projective data association [Rusinkiewicz and Levoy 2001] which matches points that are located at the same pixel. In this way, both V_{curr} and V_{prev} are reprojected to D_{curr} and D_{prev} for this association based on pixel matching;

- **Weighting of Pairs**: It is assigned constant weight for each association;

- **Rejecting of Pairs**: Pairs of correspondence are rejected if the Euclidean distance is greater than $10mm$ or angle between the corresponding normal vectors is greater than 20 degrees;

- **Error Metric**: It is used the well-known point-to-plane error metric to guide optimization. It is formally defined as:

$$argmin \sum_{u,v} ||(T_{live} V_{curr}(u,v) - V_{prev}(u,v)) \cdot N_{prev}(u,v)||^2 \qquad (3.5)$$

- **Error Minimization**: The error is minimized using a Cholesky decomposition on the linear system [Chen and Medioni 1992].

The use of the 3D reference model by the ICP allows a more consistent rigid registration with less incremental error [Izadi et al. 2011]. After the rigid transformation T_{live} estimation, it is applied to the virtual data, which can be composed with the real scene captured by the Kinect.

3.3.1 Robust Real-Time Face Tracking

As shown previously, our ICP uses projective data association to find correspondences between V_{curr} and V_{prev}. In this association, each point is transformed into camera coordinate space and perspective projected into image coordinates. The corresponding points are that on the same image coordinates. The ICP fails (i.e. does not converge to a correct alignment) when there is not a small pose variation between sequential frames. We detect this situation by checking if the linear system computed from Equation 3.5 is solvable (i.e. the matrix is invertible). To improve tracking robustness, we propose an improvement over the original ICP algorithm, adding a correction step such that if the ICP fails, taking advantage from the fact that our ROI is patient's head, we use a head pose estimation to give a new initial guess to the ICP to compute correctly the current transformation.

The use of the head pose estimation is shown in **Algorithm 1**. Given the previous depth frame D_{prev} and the current depth frame D_{curr}, the head pose estimation is used to estimate head orientation (R_{prev} and R_{curr}) and head center (Hc_{prev} and Hc_{curr}) of them. The head centers are converted from camera to global coordinates (lines 4 and 5). The incremental rotation matrix R_{inc} and the translation t_{inc} between the previous and the current head center are computed (lines 6 and 7). The translation t_{inc} is added to the current global translation t_{live} (line 8). The implicit surface is then raycasted to generate a new view (i.e. new previous depth frame) (line 9). The raycasted view is rotated around Hc_{curr} with R_{inc} (line 10). Finally, we apply again the ICP to estimate the current T_{live} (line 12).

As we propose the use of this approach in an AR environment, **Algorithm 1** must run in real-time when executed to maintain user's interactivity with the application. A real-time head pose estimation algorithm was proposed in [Fanelli et al. 2011]. In there, head pose estimation is treated as a classification problem. Decision trees are used to split an original problem into smaller ones, solving each one of them with simple predictors. In this representation, each node of the tree consists of a test which directs the problem to left or right child. During training, trees are built to favor good predicitions. Standard decision trees suffer from overfitting and trees trained randomly (i.e. random regression forests) can be applied to solve this problem [Breiman 2001]. Fanelli et al. trained random forests to estimate head pose from low-quality depth images [Fanelli et al. 2011].

Algorithm 1 ICP + Head Pose Estimation algorithm

```
for every input frame do
    run the ICP algorithm.
    if ICP failed then
        headPoseEstimator($D_{prev}$, $*R_{prev}$, $*Hc_{prev}$)
        headPoseEstimator($D_{curr}$, $*R_{curr}$, $*Hc_{curr}$)
        $R_{inc} \leftarrow R_{curr} R_{prev}^{-1}$.
        $t_{inc} \leftarrow Hc_{prev} - Hc_{curr}$.
        $t_{live} \leftarrow t_{live} + t_{inc}$.
        raycast the TSDF using $T_{live}$ to generate the new depth frame.
        rotate the raycasted view around $Hc_{curr}$ with $R_{inc}$.
        run again the ICP algorithm.
    end if
end for
```

To train the trees, each depth map was annotated with labels indicating head center and Euler rotation angles. These labels were estimated automatically using ICP after a 3D facial reconstruction. After the labeling and training, the head pose can be estimated letting every image region to vote it. The vote consists of a classification whether the image region contains a head and a retrieval of a Gaussian distribution computed during the training and stored at the leaf. This probabilistic approach achieves high accuracy and runs in real-time on CPU. Therefore, by using this approach, we could develop a robust real-time face tracking algorithm for a MAR application.

3.4 RESULTS AND DISCUSSION

In this section we analyze performance, accuracy and tracking robustness of the markerless tracking algorithm. Not only for the tests presented in this chapter, but for the entire dissertation, we used an Intel® Core™ i7-3770K CPU@3.50Ghz, 8GB RAM, NVIDIA GeForce GTX 660. Also, we used the open-source C++ implementation of the KinectFusion [Izadi et al. 2011] released by PCL project [Rusu and Cousins 2011]. User's head was reconstructed with KinectFusion using a grid with volume size of $70cm \times 70cm \times 140cm$ and resolution of 512^3.

3.4.1 Performance Evaluation

Figure 3.2 shows the time measured for each step of the 3D reference model reconstruction. In summary, it takes $\approx 25ms$ per frame (40 FPS) and requires about 15 seconds to be completed if the user has provided sufficient viewpoints. This processing time could be improved by reducing KinectFusion's volume resolution from 512^3 to 256^3, which would reduce 3D reference model accuracy as well. As can be seen in Figure 3.3, by using the resolution 256^3, 3D reference model reconstruction takes $\approx 19ms$ per frame (53 FPS). When head pose estimation was used, the total frame was increased in $40ms$, dropping application's performance to 15 FPS (for resolution of 512^3) or 17 FPS (for resolution of

256^3).

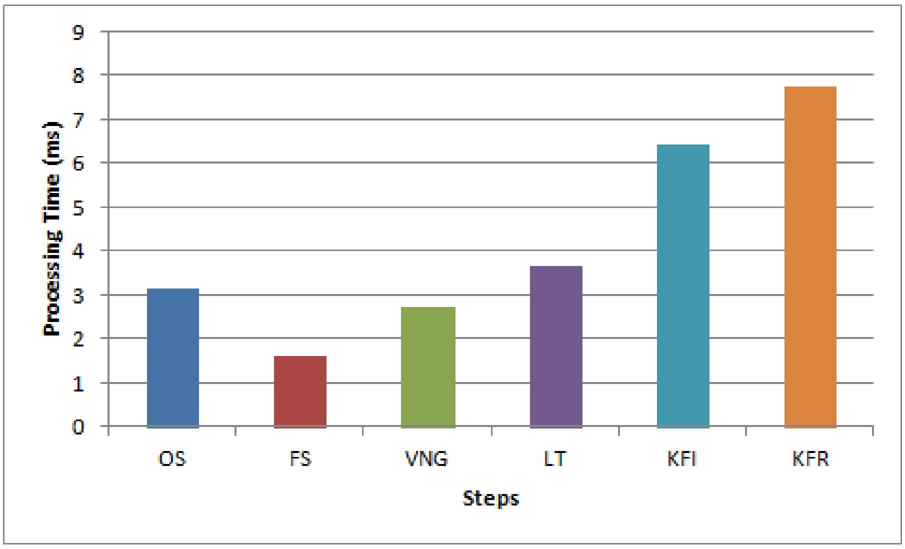


Figure 3.2 Performance results measured in average milliseconds for each step of our approach. OS - Other Steps (i.e. display timing, Kinect latency), FS - Face Segmentation, VNG - Vertex and Normal Map Generation, LT - Live Tracking, KFI - KinectFusion's grid integration, KFR - KinectFusion's grid raycasting. Times were measured running our approach with the KinectFusion's grid in resolution 512^3.

3.4.2 Accuracy Evaluation

3D reference model reconstruction has accuracy of $\approx 10mm$ [Meister et al. 2012]. Live tracking has an accuracy of $\approx 2mm$, which is not incremental (i.e. the error is not accumulated during frames). The accuracy of the registration between virtual data and reference model depends on user's adjustment. Given these accuracy measurements, we observe that, visually, the registration is accurate. In this sense, the full MAR solution proposed here may be suitable for training purposes. However, scenarios in which high accuracy is required for the MAR environment, such as the ones for surgery procedures, our solution is not recommended.

The accuracy of the head pose estimation is the same as stated in Fanelli's paper [Fanelli et al. 2011]: angle error about 8 degrees in each axis; head center error about $10mm$.

3.4.3 Tracking Robustness Evaluation

We can analyze tracking robustness for two different scenarios: fast translation and fast rotation of user's face.

When the user translated his face fastly in front of the camera, there was not sufficient points at the same image coordinates and the ICP failed. Then, the head pose estimation algorithm could give a correct initial guess to the ICP. This situation can be seen in Figure

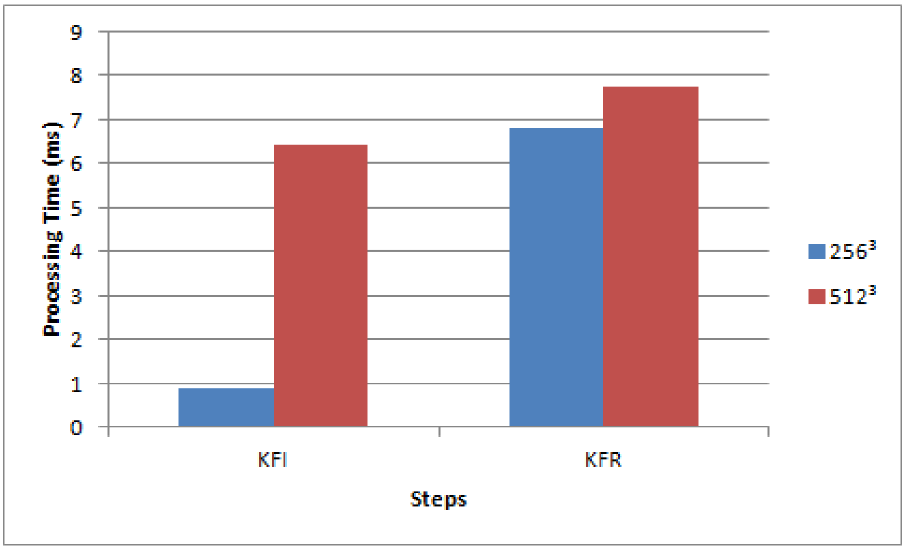


Figure 3.3 Performance results measured in average milliseconds for each step of the KinectFusion. KFI - KinectFusion's grid integration, KFR - KinectFusion's grid raycasting. Times were measured running our approach with the KinectFusion's grid in resolutions 256^3 and 512^3.

3.4.

Figure 3.4 Left image: The user translated his face fast. A small number of points were at the same image coordinates and the ICP failed. Right image: By applying our approach we solved this problem.

The algorithm ICP + Head Pose Estimation slightly improved the tracking performance when the user rotated his face and the ICP failed. The reason is that the larger the pose variation, the larger the non-overlapping region, and there are cases that the ICP is not appropriate in the presence of non-overlapping regions even if the head pose estimation provides the initial guess. When the tracking algorithm fails, as shown in Figure 3.5, the user needs to reposition his face to the tracking algorithm to align correctly the raw depth data. One can use non real-time up-to-date algorithms, such as the Sparse ICP [Bouaziz et al. 2013], to solve this issue, however this decision may decrease severely application's performance (i.e. in this case, less than 1 FPS).

Figure 3.5 An example of tracking failure. The user needs to reposition his face to the tracking algorithm to align correctly the raw depth data to the reference 3D model.

3.5 SUMMARY

In this chapter we have presented the MAR environment in which this work is based on. We used the KinectFusion to reconstruct patient's ROI and ICP algorithm to track it by estimating the current rigid transformation that defines the motion between current depth frame captured by the RGB-D sensor and the previous one represented by the 3D reference model reconstructed. To improve tracking robustness, we have extended the ICP algorithm by using a real-time head pose estimation algorithm to give the initial guess to the ICP when it failed. From the tests conducted, we have shown that our approach can handle more pose variations than ICP. In addition, we have shown that the use of the head pose estimation proposed in [Fanelli et al. 2011] is suitable for AR applications, as it runs in real-time.

Once with the MAR environment properly designed, we must see how to add support for medical volume rendering and consequently on-patient medical data visualization.

Chapter 4

ON-PATIENT VOLUMETRIC MEDICAL DATA VISUALIZATION

In this chapter we present the on-patient medical data visualization solution proposed in this dissertation. This chapter covers the methods proposed on the following authored publications [Macedo et al. 2014c, Macedo and Apolinario 2014b, Macedo et al. 2014a], [Macedo et al. 2014b, Macedo and Apolinario 2014a]. An overview of the proposed MAR environment with support to on-patient medical data visualization can be seen in Figure 4.1.

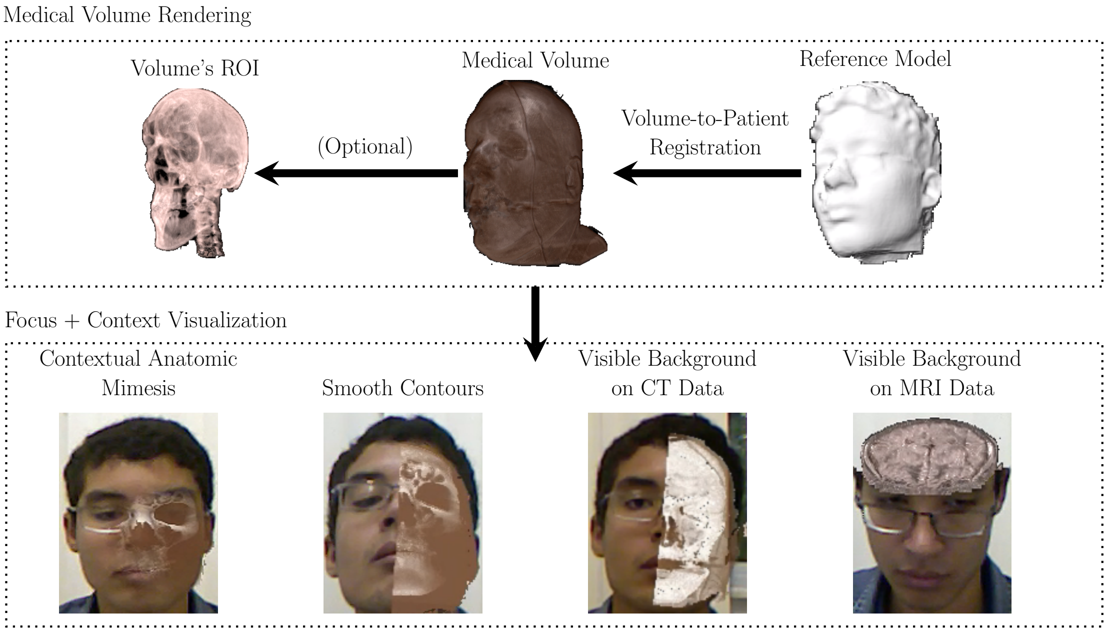


Figure 4.1 Overview of the proposed solution.

4.1 VOLUME RENDERING

Volume rendering is concerned with techniques for generating images from volume data [Hadwiger et al. 2006]. The majority of volume rendering algorithms are based on the volume rendering integral. This formulation is based on an emission-absorption optical model as shown in Equation 4.1.

$$R(s_f) = R_0 e^{-\int_{s0}^{sf} k(t)dt} + \int_{s0}^{sf} q(s) e^{-\int_{s}^{sf} k(t)dt} ds. \tag{4.1}$$

The radiance energy $R(s_f)$ is the result of integrating from entry point into the volume ($s = s_0$) to the exit point toward the camera ($s = s_f$). The absorbed energy and emission components are represented by the absorption and emission coefficients k and q respectively. The term R_0 is the radiance in the entry point s_0.

The volume is rendered according to a compositing scheme, which gives the numerical computation of the volume rendering integral:

$$R(D) = \sum_{i=0}^{n} c_i \prod_{j=i+1}^{n} Tr_j \tag{4.2}$$

where $c_i = R(s_i)$ and $Tr_i = Tr(s_{i-1},\ s_i) = e^{-\int_{s_{i-1}}^{s_i} k(t)dt}$.

Two of the most known compositing schemes are the direct volume rendering (DVR) and the maximum intensity projection (MIP). The DVR is the discretization presented in the Equation 4.2 and it is based on a front-to-back or back-to-front compositing scheme. The most common is the front-to-back DVR:

$$C_{dst} = C_{dst} + (1 - \sigma_{dst})C_{src} \tag{4.3}$$

$$\sigma_{dst} = \sigma_{dst} + (1 - \sigma_{dst})\sigma_{src} \tag{4.4}$$

where $C_{dst} = c_{i+1}$, $C_{src} = c_i$, $\sigma_{dst} = 1 - Tr_{i+1}$, $\sigma_{src} = \sigma_i$ given the voxel i being traversed. C represents the color contribution and σ the opacity of the voxel (see [Hadwiger et al. 2006] for more details).

Different from the DVR compositing scheme, MIP is computed according to the following compositing equation:

$$C_{dst} = max(C_{dst}, C_{src}) \tag{4.5}$$

The final result is the maximum color contribution along a ray [Hadwiger et al. 2006]. This compositing scheme is particularly important in virtual angiography (i.e. the display of the vessel structures in medical scans) [Prokop et al. 1997].

The volume data is usually represented as a 3D texture with associated colors. These are color components that have already been weighted by their corresponding opacity values [Blinn 1994]. This representation allows the generation of images with higher quality than the solution based on 2D texture because of the interpolation performed on the discrete set of color points [Hadwiger et al. 2006]. Moreover, it avoids the presence of color bleeding artifacts [Wittenbrink et al. 1998].

The scalar values stored in the volume data represent some spatially varying physical property. Transfer functions can be applied on these scalar values to improve user's visual perception and data interpretation of the volume. The transfer functions map the values to colors in RGB space. In this sense, pre-integrated transfer functions are used to capture the high frequencies introduced in the transfer functions with low sampling rates [Engel et al. 2001].

To render the medical data based on DVR or MIP, the ray casting algorithm proposed initially by [Levoy 1988] and implemented in GPU by [Kruger and Westermann 2003, Roettger et al. 2003] is the technique most used. The start positions of the ray are obtained by rasterizing the front faces of the volume bounding box and the exit positions of the ray are obtained by rasterizing the back faces of the bounding box. Direction is computed from the difference between the exit and start positions. Ray casting is performed by sampling the space in-between the volume bounding box through the use of an adaptive sampling rate. Ray casting can be done on GPU in a single rendering pass on the fragment shader [Hadwiger et al. 2009].

One of the main advantages of ray casting is that it is flexible in the sense that many other techniques can be integrated to improve performance or image quality of the rendering.

The most common techniques to optimize performance of the volume rendering are empty space skipping, adaptive sampling and early ray termination. For empty space skipping, the volume is subdivided into an octree. In order to detect empty spaces, each block stores the minimum and maximum scalar values. The visibility of each block can be determined after evaluation of the transfer function [Li et al. 2003]. If the block is considered invisible, the step size of the ray can be increased, skipping the region where it is already known that there is not relevant data to be processed. Otherwise, by using adaptive sampling, the step size of the ray can be adaptively decreased to compute accurately the composition scheme. Early ray termination stops ray traversal if the opacity accumulated is too high, situation in which the contribution of new samples is irrelevant for the final rendering.

To improve image quality, several techniques can be used. To reduce sampling artifacts, a stochastic jittering (i.e. random ray-start off-setting) is commonly applied to the ray start position. To reduce the filtering artifacts, fast GPU-Based tri-cubic filtering [Sigg and Hadwiger 2005, Ruijters et al. 2008] and GPU pre-filter for accurate tri-cubic filtering [Ruijters and Thévenaz 2012] can be used to replace native tri-linear filtering.

The volume rendering integral presented in the Equation 4.1 does not account for illumination effects caused by external light sources. Such illumination effects, however, add a great deal of realism to the resulting images. This is specially important in an AR environment, where this illumination effect serves as an approximation for the illumination of the real scene. To compute local illumination, it is used Blinn-Phong shading [Blinn 1977] with on-the-fly gradient computation by central or forward differences on the GPU. Non-polygonal iso-surface rendering is realized by first hit ray-casting. The local illumination is included in Equation 4.1 by extending the emission coefficient $q(s) = q_{ea}(s) + q_{il}(s)$, where $q_{ea}(s)$ is the emission coefficient of the emission-absorption model and $q_{il}(s)$ is the coefficient that adds the local illumination [Hadwiger et al. 2006].

Sometimes, the goal of the volume rendering is to extract and emphasize important parts of a volume. This issue can be solved by using volume clipping, which cuts away unimportant regions of the volume [Hadwiger et al. 2006]. In general, the volume is clipped according to one of six planes parallel to the faces of the volume bounding box.

Finally, our approach supports DVR and MIP compositing schemes, iso-surface rendering, single-pass GPU ray casting, empty space leaping of non-visible voxels, early ray termination and adaptive sampling to optimize performance, stochastic jittering, tri-cubic filtering, pre-integrated transfer function and local illumination to improve image quality, and other features such as volume clipping.

4.1.1 Context-Preserving Volume Rendering

When the medical volume is rendered based on DVR, at first glance, the final rendering may not be the result desired by the user. In this sense, transfer functions can be used to change the visual aspect of the volume, enhancing the features of the medical data. However, the process to find an appropriate transfer function can be a complex and time-consuming task. An alternative is to pre-segment the volume in regions of interest and define transfer functions for these regions separately. This solution improves the quality of the final rendering, but it is more time-consuming than defining the transfer function without pre-segmentation. Therefore, it would be desirable in this context a new solution fast, ease to handle and that provides good visual quality for the volume rendered.

Volume clipping can be used to reveal hidden structures of the volume by completely cutting away occluding areas. Traditionally, clipping planes are used to perform this task in a faster and intuitive way [Hadwiger et al. 2006], although several up-to-date techniques, such as [Wu et al. 2013], have proposed alternatives to the plane as a cutting object. However, when clipping a volume, some important information can be lost in this process if the information is located in the cutted region. Therefore, volume clipping is not the best way to enhance the visualization of the volume.

Illustrative volume rendering is a field which aims to improve the visualization of volumetric structures by the use of non-photorealistic strategies integrated into the rendering algorithm. Inspired by the fields of volume clipping and illustrative volume rendering, Bruckner et al. proposed the context-preserving volume rendering [Bruckner et al. 2005, Bruckner et al. 2006], which uses a function of:

1. Shading intensity - to decrease opacity in regions which receive high illumination intensity;
2. Gradient magnitude - to preserve and enhance contours;
3. Distance to the eye point - to simulate the cut effect of a clipping plane;
4. Previously accumulated opacity - to turn structures located behind semi-transparent regions more opaque.

This model is able to reduce the opacity in less important data regions and it is controlled by two user-specified parameters: one to explore the dataset and another to control the sharpness of the visualization.

In our MAR environment, we use this technique not only to enhance features of the volume, but also to separate some structures in the visualization, such as bones, soft tissue and organs (Figure 4.1). This process can be done just by controlling the two parameters mentioned before.

4.2 MEDICAL VOLUME-TO-PATIENT REGISTRATION

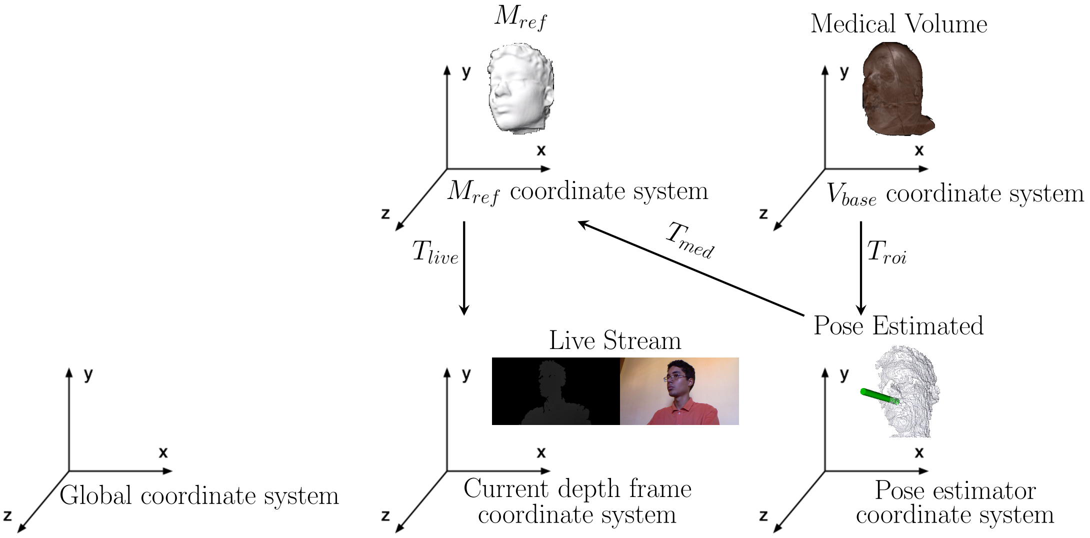


Figure 4.2 Relationships among the coordinate systems used for medical volume-to-patient registration.

After volume rendering, the next step is the placement of the medical data into the scene (Figure 4.1). However, as will be discussed in Section 4.5, in this work the patient's ROI is augmented with a generic volumetric data instead of his own. Therefore, we do not use an automatic registration method. Instead, we propose a new method for coarse registration between medical volume and patient's 3D reference model. Then, the result can be fine adjusted by the user. An overview of the coordinate systems used for the coarse registration process is shown in Figure 4.2.

The process to position the medical volume into the patient is shown in **Algorithm 2**. Given the reference model M_{ref} extracted from KinectFusion's volume and the medical volume inside a normalized unitary bounding box, the scale factor S_{med} can be computed for each axis based on the average size of the bounding box sides of M_{ref} (line 1). Next, the centroid c_{ref} is computed from M_{ref} (line 2). The rotation matrix of the medical data R_{med} is initially set to be a pre-defined initial pose R_0 (line 3). Pose R_{roi} and center of mass c_{roi} of patient's ROI are estimated with a trained pose estimator based on the region of interest (lines 4-5). c_{roi} is assigned to be the translation vector of the region of interest: t_{roi} (line 6). As the ROI in this work is a head, we estimate these values from D_{curr} captured by the Kinect sensor with the real-time head pose estimation proposed in [Fanelli et al. 2011]. The centroid of the ROI's center of mass rotated c_{rot} is computed (line 7). Because we want the centroids of the medical data and the reference model to be the same (i.e. at the same position), and assuming that the initial centroid of the volume is in the origin, the translation vector of the medical data can be computed by the subtraction of c_{ref} and c_{rot} (line 8).

Algorithm 2 Estimating coarse pose

```
1: $S_{med} \leftarrow$ compute scale factor from $M_{ref}$.
2: $c_{ref} \leftarrow$ compute centroid from $M_{ref}$.
3: $R_{med} \leftarrow R_0$.
4: $R_{roi} \leftarrow$ estimate rotation matrix for the region of interest.
5: $c_{roi} \leftarrow$ estimate centroid of the region of interest.
6: $t_{roi} \leftarrow c_{roi}$.
7: $c_{rot} \leftarrow R_{roi} c_{roi}$.
8: $t_{med} \leftarrow c_{ref} - c_{rot}$.
```

After the computation of the coarse pose, the medical volume data Vo can be positioned into the scene for every frame by the following expression:

$$Vo_{live} = T_{live} T_{med} T_{roi} Vo_{base}. \tag{4.6}$$

where T_{live} is the 3D rigid transformation estimated by the tracking algorithm. $T_{roi} = [R_{roi}|t_{roi}]$ is the 3D rigid transformation that registers the medical data in relation to the pose measured for the region of interest, Vo_{base} means the volume in an initial position, and T_{med}, which is defined by the following equation:

$$T_{med} = [R_{med} diag(S_{med})|t_{med}]. \tag{4.7}$$

is the 3D transformation that gives the coarse placement of the volume into the scene. The function *diag* creates a 3×3 diagonal matrix from the input vector. The final result, Vo_{live}, is the volume aligned to the patient in the current frame.

Once in the origin on the unitary bounding box, the sequence of transformations can be applied to place the volume on the current pose of the patient's ROI. The user can adjust volume's position and orientation by controlling T_{med}.

4.3 FOCUS + CONTEXT VISUALIZATION

An application for on-patient medical data visualization requires a special attention to the mixing between real (i.e. patient's image captured by the sensor) and virtual (i.e. patient's anatomy stored in the computer) entities of the AR environment. Instead of superimposing the virtual content onto patient's real image, a better solution is to show a part of the virtual anatomy as a focus region in the context of the patient's body. In this sense, our MAR environment supports four state-of-the-art F+C visualization techniques, which are described in the next subsections. We focus on the description of the last three techniques (i.e. smooth contours, visible background on CT and MRI data) as they were developed as novel contributions for this dissertation.

4.3.1 Contextual Anatomic Mimesis

Bichlmeier et al. proposed the Contextual Anatomic Mimesis (CAM) to improve depth perception in a medical AR application [Bichlmeier et al. 2007]. First, the medical data is not entirely overlaid over the patient's image. Focus point and radius are defined by line of sight. Together, they act like a mask in the AR scene, in which the medical data can be visualized only inside it. Next, Bichlmeier and colleagues proposed the control over the AR visualization by adjusting the following parameters:

1. Curvature (α_{curv}) - The curvature of the patient's surface allows regions with high curvature (e.g. wrinkled, bumpy and sinuous regions) to remain visible in the final rendering;

2. Distance Falloff ($\alpha_{distFalloff}$) - The distance between each point on the surface and the focus point allows a smooth visualization between the patient's medical data and the real image;

This method was integrated in our environment, as can be seen in Figures 4.1 and 4.3, because it has already proven to provide high-quality on-patient medical data visualization [Kutter et al. 2008, Wieczorek et al. 2010].

Figure 4.3 Contextual Anatomic Mimesis applied on our environment.

4.3.2 Smooth Contours

One of the issues related to the integration of the volume clipping into the AR environment is the presence of the visible edge located in the intersection between the volume and the clipping plane. As discussed in [Bichlmeier et al. 2007], it is desirable a smooth transition between the volume in the focus region and the rest of the AR scene. In order to achieve this goal, it is proposed here a new method for F+C visualization based on **smooth contours**, an algorithm that adds a smooth transition between the image resulting from the volume clipping and the context region. An overview of this algorithm can be seen in Figure 4.4.

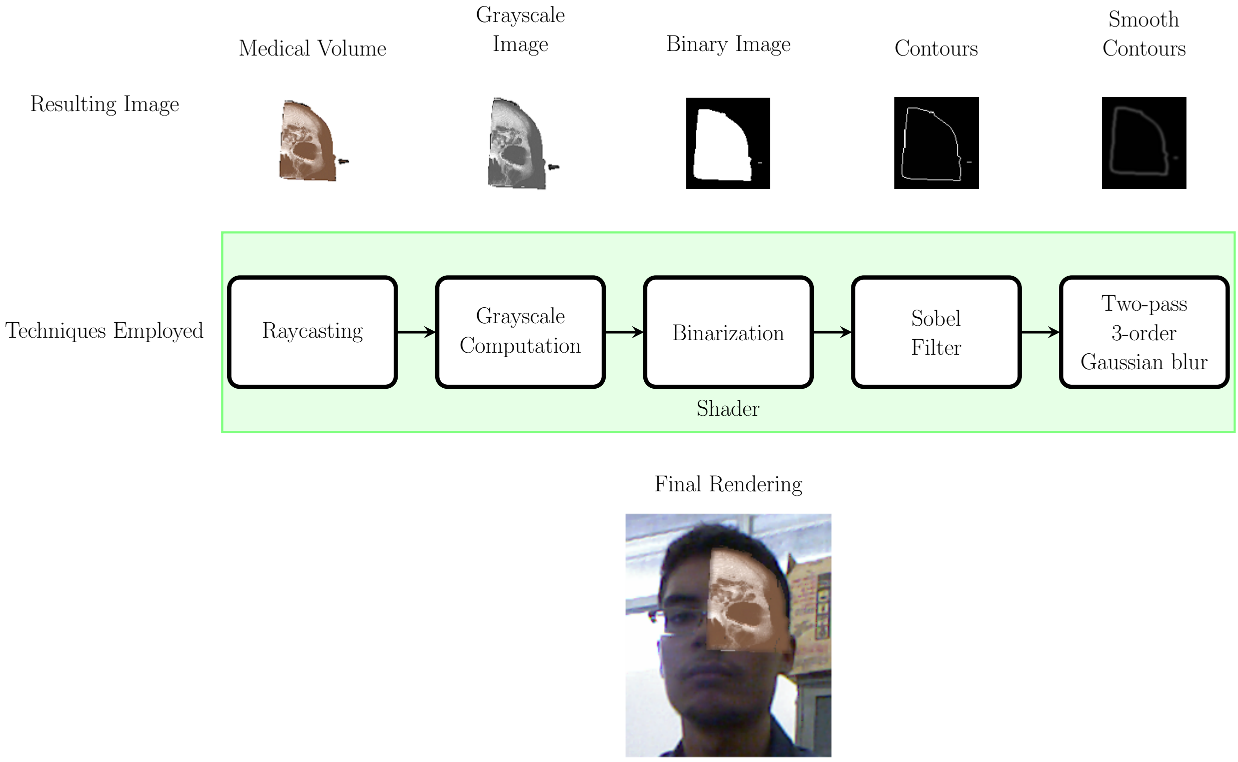


Figure 4.4 Smooth contours pipeline.

The image of the volume is converted into a gray-scale image and then is binarized by a pre-defined threshold t_b, which operates on the gray intensity of each pixel (empirically we have found $t_b = 0.1$ a good threshold for such task). The contours of the binarized image are extracted by using the Sobel edge-detector filter. The contours are blurred by a n-order two-pass gaussian blur. From empirical tests, $n = 3$ produced the best results. The resulting image $\alpha_{smoothContours}$ is a mask that gives a weight to the blending between the volume contours and the scene. Also, a factor w_c can be dynamically defined by the user to adjust the level of smoothing in the contours (Equation 4.9). It ranges from 0, where the contours are rendered, to $+\infty$, where the contours are smoothed and then suppressed in the final rendering. All these operations run on the shader.

The influence of the smooth contours can be seen in Figures 4.1 and 4.5. It is visible that the proposed algorithm adds a smooth transition between the volume and the real scene.

This method can be easily integrated with the F+C method proposed by Bichlmeier et al. [Bichlmeier et al. 2007]. An example of the result of the proposed integration can be seen in Figure 4.6.

Taking advantage from the clipping effect, we propose another two methods for F+C visualization, now based on **visible background**. In these methods, we take advantage from the type of scanning (CT or MRI) to enable new ways for physicians to explore the medical data on the patient.

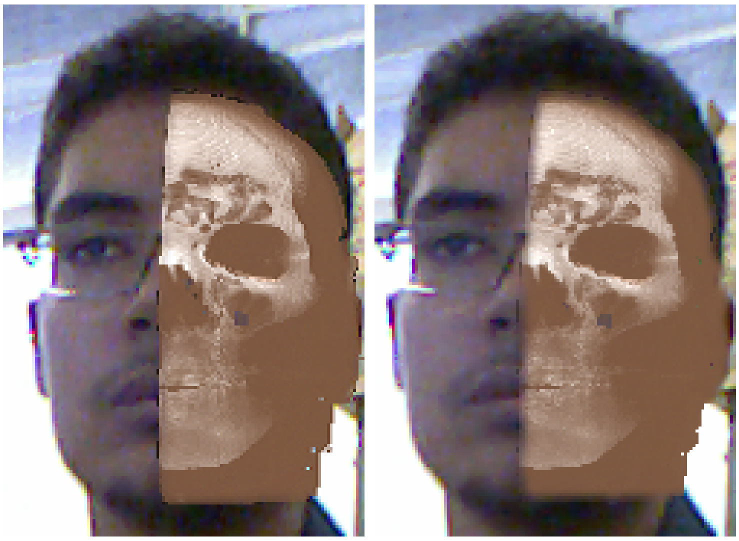


Figure 4.5 Influence of the smooth contours in the final rendering. Left image: Direct volume rendering with clipping. Right image: Volume clipped rendered according to the proposed algorithm.

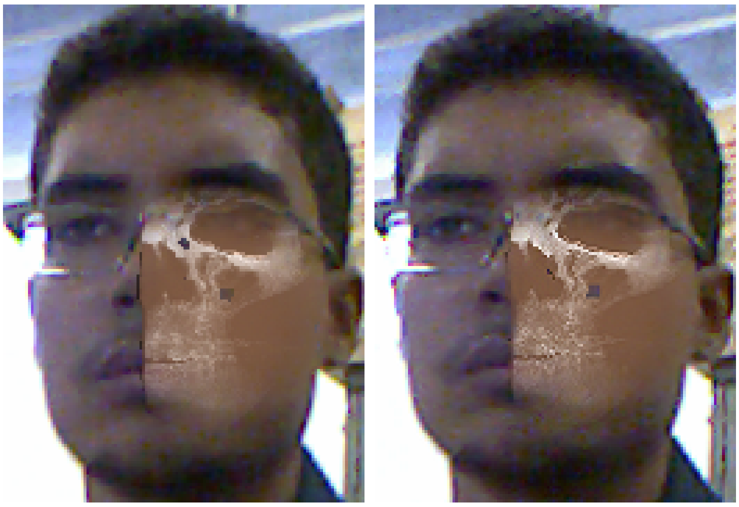


Figure 4.6 F+C visualization based on distance falloff parameter proposed by CAM algorithm (left image) and its extension with the smooth contours (right image).

4.3.3 Visible Background on CT Data

In volume rendering, CT data can be used to enable the visualization of internal structures such as bones. In the case of a head, this structure can be the patient's skull. By designing an appropriate transfer function, the skull can be visualized apart from the soft tissue of the volume. In this situation, it is desirable to see the real background scene when looking at the region between the skull and the soft tissue or the soft tissue itself, because both are context regions of the volume and the virtual background can be seen, depending on the transfer function chosen (Figure 4.1). An overview of the proposed approach to enable this kind of visualization can be seen in Figure 4.7.

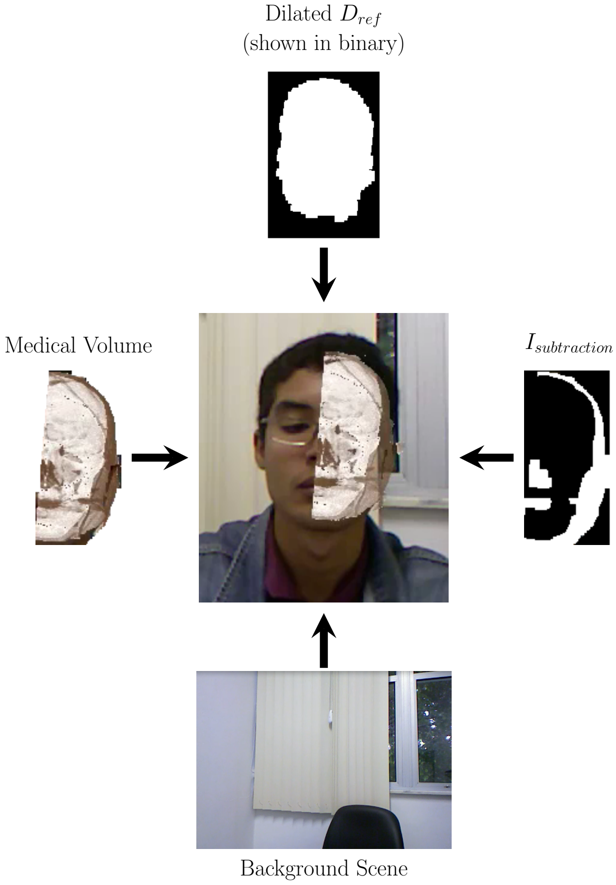


Figure 4.7 Visible Background on CT data overview.

For this kind of F+C visualization, the background scene is captured and stored in the memory. Next, the image of the volume after clipping is binarized and sent to the shader as a foreground subtraction mask $I_{subtraction}$. This mask identifies the region where the background can be visualized based on the gray intensity of the volume. In our case, D_{ref}, the depth image of the 3D reference model, does not overlap perfectly the patient's ROI. To solve this problem, D_{ref} is dilated only on its contours to preserve the original depth of the 3D reference model and sent to the shader to represent the patient's ROI. The pseudocode of the shader to render the F+C visualization based on visible background on CT data can be seen in the **Algorithm 3**, lines 1-15, 22-24.

The color image captured from the Kinect sensor is rendered in the region that does not represent the patient's ROI (i.e. where the depth of the 3D reference object is 0, as it was not reconstructed) (lines 2-4). The captured color image is also rendered when the volume is occluded and the occludee has depth (e.g. it is not in a hole region) (lines 5-7). Next, if the fragment is in the subtraction mask region, the volume or the background scene are rendered. Otherwise, the fragment is in the clipped region and the real color image is rendered (lines 23-24). The gray intensity is computed from the volume (by the *gray* function) and assigned to β. Considering that the bone is rendered with a gray level greater than the soft tissue's and than a user-defined threshold $w_{grayLevel}$, it is rendered without the background scene. As the bone and the soft tissue have different gray intensities, $w_{grayLevel}$ can be adjusted to render the bone with its real color and the soft tissue linearly interpolated with the background scene (lines 8-15).

Algorithm 3 F+C visualization based on visible background

```
 1: for in parallel do
 2:     if $D_{ref} == 0.0$ then
 3:         return $I_{real}$;
 4:     end if
 5:     if $D_{live} < D_{ref}$ and $D_{live}$ != 0.0 then
 6:         return $I_{real}$;
 7:     end if
 8:     if $I_{subtraction} == 1.0$ then
 9:         if CT data then
10:             $grayLevel \leftarrow gray(I_{medical})$;
11:             $\beta \leftarrow grayLevel$;
12:             if $grayLevel < w_{grayLevel}$ then
13:                 return $\beta * I_{background} + (1 - \beta) * I_{medical}$;
14:             end if
15:             return $I_{medical}$;
16:         else
17:             if $I_{medical} == 0.0$ then
18:                 return $I_{background}$;
19:             end if
20:             return $I_{medical}$;
21:         end if
22:     end if
23:     return $I_{real}$;
24: end for
```

4.3.4 Visible Background on MRI Data

In volume rendering, MRI data can be used to enable the visualization of internal structures in the body such as organs. In an AR environment, the best way to visualize these data is by clipping not only the volume but also the corresponding region on the patient's image (Figure 4.1). In this situation, it is desirable to see the background scene in the region clipped. An overview of the proposed approach to visualize this kind of scenario can be seen in Figure 4.8.

The background scene is saved. Next, taking advantage from the volumetric representation of the KinectFusion, in which the 3D reference model is stored, patient's ROI

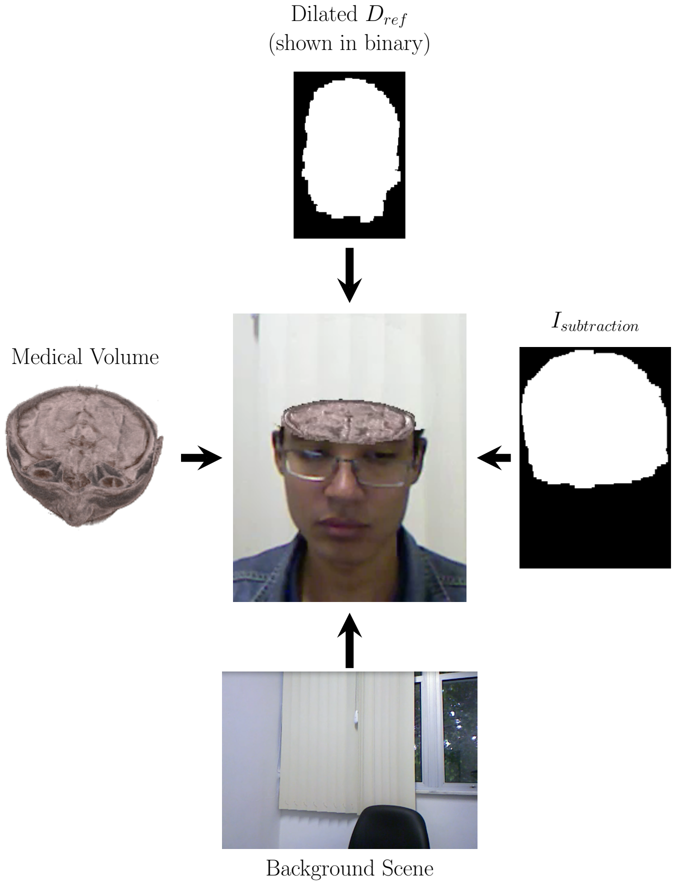


Figure 4.8 Visible Background on MRI data overview.

is clipped. The algorithm to render an image from the clipped 3D reference model can be seen in **Algorithm 4**. This algorithm is an extension of the pseudocode presented in [Izadi et al. 2011]. The algorithm ray casts the TSDF and when the ray traverses a zero-crossing position and it is in the clipped region, the voxel's corresponding pixel is rendered in the output image. To mitigate the presence of visible artifacts at the intersection between the clipping plane and the 3D reference model, traditional adaptive sampling is employed to reduce step size of the ray when it is near the zero-crossing position (i.e. silhouette of the 3D reference model stored in the volume). We check this proximity by using a specific threshold (t_{prox}) over the TSDF stored at the voxel g being traversed (g_{tsdf}). When near the zero-crossing, step size of the ray casted is reduced to a value w_s to perform a more accurate traversal. From empirical tests, we have set $t_{prox} = 0.5$ and w_s equals to one-fourth of the original step size. The output image from this algorithm is $I_{subtraction}$, as it will be used to indicate whether the background image must be rendered. The medical volume is clipped separately and sent to the shader. Both $I_{subtraction}$ and D_{ref} are dilated because of the problem of overlapping described before. The pseudocode of the shader to render the F+C visualization based on visible background on MRI data can be seen in the **Algorithm 3**, lines 1-8, 16-24.

The color image captured from the Kinect sensor is rendered in the same way as described in visible background on CT data. The main difference here is that if the subtraction mask is active (i.e. patient's ROI is clipped) and if there is medical data to be visualized, it is rendered. Otherwise, the background image is rendered.

Algorithm 4 Raycasting the clipped region of the 3D reference model volume

```
for each pixel $\mathbf{u} \in$ output image $I_{subtraction}$ in parallel do
    $I_{subtraction}(u) \leftarrow 0$;
    $ray^{start} \leftarrow$ back project [$\mathbf{u}$, 0]; convert to grid pos
    $ray^{next} \leftarrow$ back project [$\mathbf{u}$, 1]; convert to grid pos
    $ray^{dir} \leftarrow$ normalize $(ray^{next} - ray^{start})$
    $ray^{len} \leftarrow 0$
    $g \leftarrow$ first voxel along $ray^{dir}$
    while voxel $g$ within volume bounds do
        $ray^{len} \leftarrow ray^{len} + stepsize$
        $g^{prev} \leftarrow g$
        $g \leftarrow$ traverse next voxel along $ray^{dir}$
        if $g_{tsdf} < t_{prox}$ then
            $stepsize \leftarrow w_s$
        end if
        if zero crossing from $g$ to $g^{prev}$ and $g$ is in the clipped region then
            $I_{subtraction}(u) \leftarrow 255$;
        end if
    end while
end for
```

4.4 FINAL RENDERING

In a naive implementation, when the volume is clipped and its image is rendered in an AR environment, we obtain a result similar to the one presented in the Figure 4.9, left image, where there is no occlusion between the internal region of the volume visible after the clipping and the patient. If desirable, this effect can be removed in the single-pass ray casting, as shown in **Algorithm 5**, which is an extension of the pseudocode for ray casting algorithm presented in [Hadwiger et al. 2006]. We check if the first hit position of the ray in the volume is in the clipped region. If it is, the ray is discarded from rendering, otherwise, the ray traverses the volume as normally done in the standard ray casting algorithm. The result of the application of this algorithm can be seen in Figure 4.9, right image.

After volume rendering, color frame buffer is sent to the shader for blending with the patient's RGB data coming from the RGB-D sensor. For CAM and smooth contours F+C visualization techniques, blending is done by the following linear interpolation:

$$I_{final} = \beta I_{real} + (1 - \beta) I_{medical} \tag{4.8}$$

where I_{real} is the image captured by the sensor, $I_{medical}$ is the image generated by the

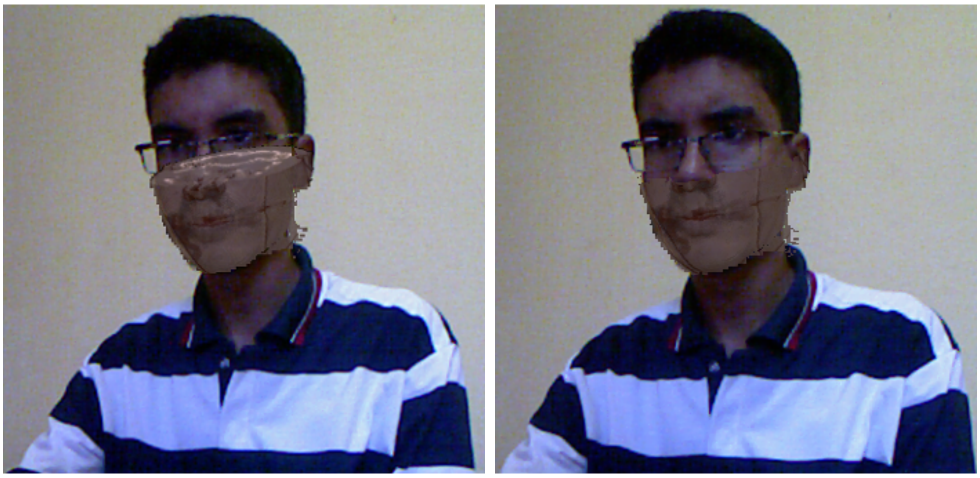


Figure 4.9 Occlusion between volume's internal structures and the patient's region of interest. Left image: Direct volume rendering with clipping. Right image: Volume clipped rendered according to the proposed algorithm.

volume rendering applied to the medical volume, and I_{final} is the resulting image. In our approach, β is defined dynamically, for every fragment/pixel, by the F+C visualization techniques mentioned before, according to the following equation:

$$\beta = clamp(max(\alpha_{curv}, \alpha_{distFalloff}, w_c \alpha_{smoothCont})) \tag{4.9}$$

clamp is a function that clamps its input parameter to the interval [0,1].

For the visible background-based F+C techniques, the shader proposed in **Algorithm 3** is used, as Equation 4.8 is not valid because it does not include background rendering.

To solve occlusion, D_{curr} and D_{prev} are used. Where D_{curr} is in front of D_{prev}, volume is the occludee. Otherwise, it is the occluder.

4.5 RESULTS AND DISCUSSION

In this section we analyze visual quality and performance of the whole approach. Particularly, we evaluate the techniques employed for volume rendering and each one of the main methods for F+C visualization discussed in this dissertation.

4.5.1 Experimental Setup

The medical dataset used is a CT volumetric data of a head from the Visible Human Project [National Library of Medicine 2014] of resolution 128 × 256 × 256 and a MRI data of a head from the MRI Head available in Volume Library [Volume Library 2014] of resolution 256^3. Both are of typical resolution of a head medical volume and therefore do not affect our performance evaluation.

We evaluate the performance and visual quality of the proposed techniques in a scenario where the patient's head is augmented with a volumetric dataset of a head. The

Algorithm 5 Solving the occlusion between patient and volume's internal structures

```
Determine volume entry and exit positions.
Compute ray direction.
while ray position in volume do
    bool clip ← check if the current position is in clipped region;
    if not first hit in volume then
        Access data value at current position.
        if opacity > threshold and clip is true then
            return;
        else
            first hit in volume;
        end if
    end if
    if clip is false then
        Access data value at current position.
        Compose color and opacity;
    end if
    Advance position along ray;
end while
```

use of a generic volume does not affect our visual quality evaluation since the volume is scaled and positioned semi-automatically by the user.

4.5.2 Performance Evaluation

Figure 4.10 shows the time measured for each step of the MAR live tracking. It takes ≈ 22ms per frame (45 FPS). Occlusion computation, which transfers data stored in GPU to CPU, converts 3D reference model and 3D object coming from Kinect to the same coordinate system and sends their depth maps to the shader, takes 5ms in our approach. Meanwhile, our optimized direct volume rendering takes the lowest time.

As the Kinect sensor provides depth and color maps at 30 FPS, our approach can process every input frame during 3D reference model reconstruction and MAR live tracking with on-patient medical data visualization. Therefore, we can conclude that our approach runs in real-time.

We have observed that the user takes about 10 seconds to position and adjust the volume in the scene. The algorithm for coarse medical volume-to-patient registration is used only once (i.e. at the transition between 3D reference model reconstruction and on-patient medical data visualization) and takes 60ms.

In a second test, we evaluated the influence of the volume resolution on the overall performance of our approach. It has never dropped below 29 FPS using medical and KinectFusion volumes of resolution 256^3 and 512^3 with DVR. Therefore, we can use the maximum KinectFusion's volume size to generate a more accurate 3D reference model.

In a third test, the average processing time for various volume rendering compositing schemes was measured. The performance results can be seen in Figure 4.11. From an

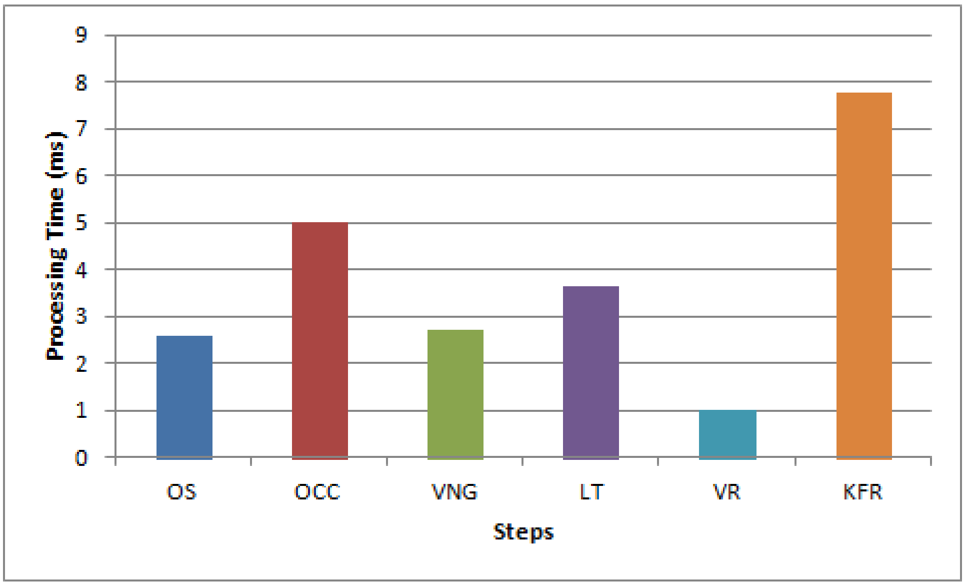


Figure 4.10 Performance results measured in average milliseconds for each step of our approach. OS - Other Steps (i.e. display timing, Kinect latency), OCC - Occlusion Computation, VNG - Vertex and Normal Map Generation, LT - Live Tracking, VR - Volume Rendering, KFR - KinectFusion's grid raycasting. Times were measured running our approach with the KinectFusion's grid in resolution 512^3 and medical dataset in resolution 256^3.

analysis on application's performance, if the volume rendering takes less than 10ms, the application still keeps the real-time performance. By assuming that the typical resolution of a head medical volume is 256^3 and considering the reported processing times in Figure 4.11, we conclude that our approach runs in real-time because its performance is greater than 30 FPS. However, with a volume of resolution 512^3, depending on which mode is chosen, we have a loss in the performance, dropping application's performance to 20 FPS in the worst scenario.

As described in Section 4.1, the volume is stored as a discrete 3D texture. When the ray is casted into the volume, it accesses the space between the discrete samples of the volumetric data. In this case, the trilinear interpolation is used to reconstruct a continuous representation of the volume based on the eight closest neighbours samples of that space. This is the most expensive operation in volume rendering based on ray casting as it requires eight memory access to perform the interpolation. Based on this statement, it is possible to evaluate the variation in performance of the different volume rendering modes.

In the simplest DVR, trilinear interpolation is performed only once for each position of the ray casted. Therefore, this is the rendering mode which takes the lowest processing time. As consequence, it produces the simplest visual effects, which can be seen in Figure 4.12-A.

For MIP, a variation of DVR mode, trilinear interpolation is also done only once. Therefore, it has the same performance measured for DVR mode.

In the non-polygonal iso surface rendering, Blinn-Phong shading is computed when the

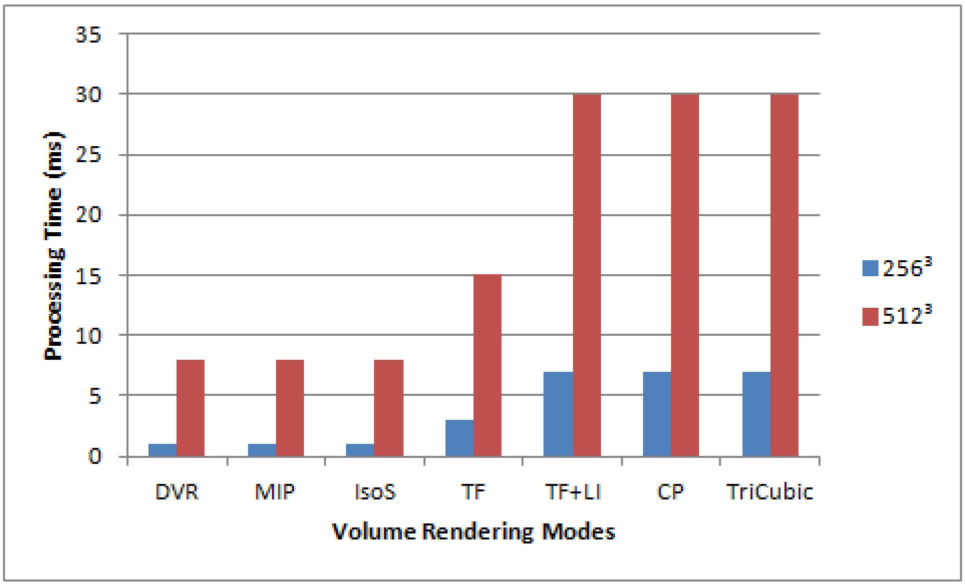


Figure 4.11 Performance results measured in average milliseconds for various volume rendering compositing schemes. DVR - Direct Volume Rendering. MIP - Maximum Intensity Projection. IsoS - Non-Polygonal Iso Surface. TF - DVR + Transfer Function with Pre-Integration. LI - DVR + Local Illumination via Blinn-Phong shading. CP - Context-Preserving Volume Rendering. TriCubic - Fast Tricubic Filtering.

ray traverses a voxel with isovalue greater than a threshold defined semi-automatically. The normal vector for a given voxel is computed by the normalization of the central differences of the neighbouring voxels. This gradient estimation requires six trilinear interpolations. However, as it is not computed for every voxel being traversed, it does not increase significantly the computational cost of the volume rendering. An example of non-polygonal iso surface rendering can be seen in Figure 4.12-D.

In the DVR with pre-integrated transfer function, after the trilinear interpolation of the voxel, the scalar value of the previous and current voxel being traversed are used as a look-up in a 2-D pre-integration table. This lookup is performed with a bilinear interpolation. It increases the volume rendering processing time to 15 ms per frame and slightly decreases the performance of the application to 35 ms per frame (28 FPS). Despite of this fact, we have a more pleasant visualization of the volume. An example of such effect can be seen in Figure 4.12-B.

In the DVR with transfer function and local illumination, for every voxel being traversed, the transfer function is accessed (with a bilinear interpolation) and the illumination is computed (with six trilinear interpolations). These interpolations decrease significantly the performance of the volume rendering to 30 ms per frame and the application to 50 ms per frame (20 FPS), which is not prohibitive, as the user can still interact with the application with some delay. In the final result, the illumination effects add realism to the resulting image. An example can be seen in Figure 4.12-C.

Context-Preserving volume rendering adds some computation for DVR with transfer function and local illumination. However, there is not a new trilinear interpolation to be

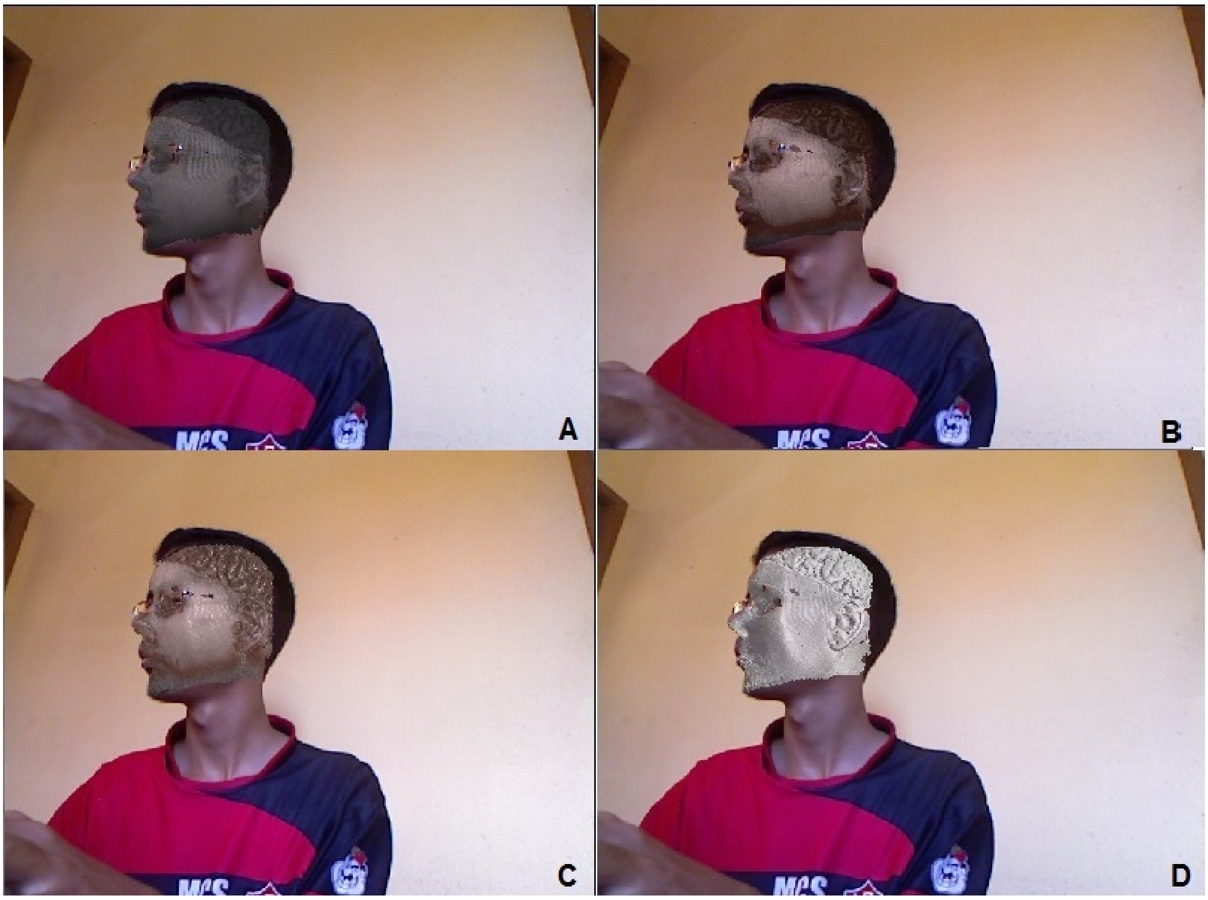


Figure 4.12 Some of the visualization options. A) Direct volume rendering (DVR). B) DVR with pre-integrated transfer function. C) DVR with pre-integrated transfer function and Blinn-Phong illumination. D) Non polygonal iso surface volume rendering.

Figure 4.13 A volume rendering (left) with stochastic jittering (center) and tri-cubic filtering (right). The stochastic jittering reduces the wood-grain artifacts in the volume, however it is almost imperceptible in this scene. The tri-cubic filtering smoothes the volume data, reducing the artifacts present in the volume rendered with trilinear filtering.

performed. Therefore, Context-Preserving mode has the same performance as the case mentioned above.

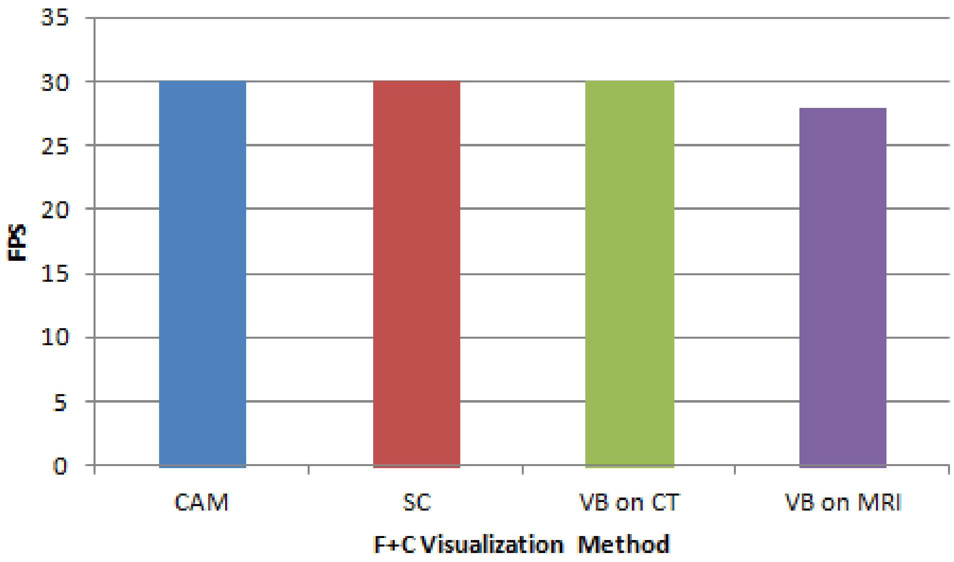


Figure 4.14 Performance results measured in FPS for each one of the F+C visualization techniques discussed in this dissertation. CAM refers to the technique proposed in [Bichlmeier et al. 2007]. SC - Smooth Contours, VB on CT - Visible Background on CT Data, VB on MRI - Visible Background on MRI Data. Times were measured running our approach with the KinectFusion's grid in resolution 512^3.

In the test performed with a simple DVR and fast tri-cubic filtering, for every voxel being traversed, eight trilinear interpolations are computed to return one tricubic interpolation. These interpolations decrease the performance similarly to the situation of transfer function and local illumination. The volume requires 30 ms per frame to be rendered and the application requires 50 ms per frame (20 FPS). The visual influence of the tri-cubic interpolation against the trilinear one can be seen in Figure 4.13.

The performance of our application for each F+C visualization technique can be seen in the Figure 4.14.

The F+C visualization proposed in [Bichlmeier et al. 2007] runs in full real-time because it operates directly on the shader by changing two parameters described in the Section 4.3.1. This result stays for the F+C visualization based on smooth contours, which runs in full real-time because operates directly on the shader.

The F+C visualization based on visible background on CT data runs in full real-time because most of this processing is done on the shader. The dilation applied on D_{ref} is negligible is terms of performance.

The F+C visualization based on visible background on MRI data is slightly slower than the one based on CT data because of the ray casting performed on the KinectFusion's volume to render the clipped region. When it is desirable high-quality rendering, the step size of the ray casted can be decreased, at the cost of loss in performance.

In applications where the performance is a critical factor, the volume size of the KinectFusion's volume can be reduced from 512^3 to 256^3. The drawback of this changing is the presence of more artifacts in the final composition, as the 3D reference model will be reconstructed with a lower quality and it has a direct relation with the proposed techniques for two reasons: it is the reference for live tracking and, from it, the region of interest is dilated or clipped.

4.5.3 Visual Quality Evaluation

Figure 4.15 Occlusion support.

As mentioned in Section 4.1, the proposed approach supports various volume rendering modes. Some of them can be seen in Figure 4.12. As discussed in previous subsection, each one of them has some impact on application's performance, although it still runs in real-time for typical-sized medical data.

Figure 4.13 shows the influence of some techniques used to improve the image quality of the volume rendering. Artifacts are reduced without prohibitive increase in the computational cost.

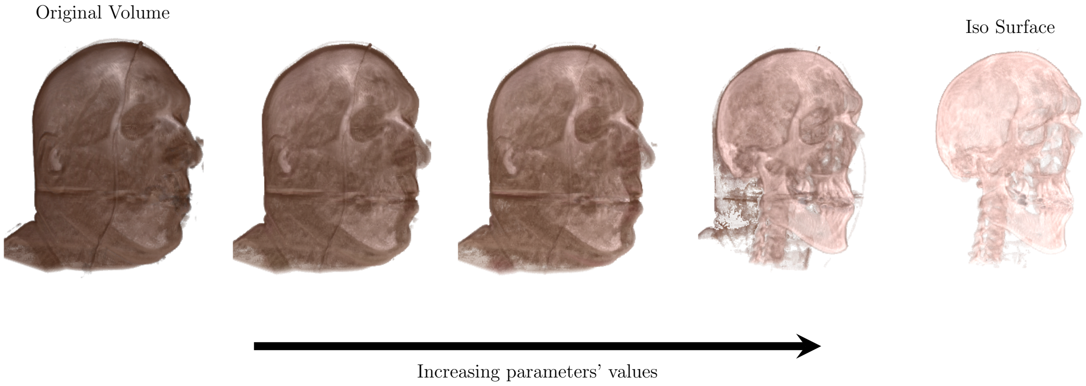


Figure 4.16 From volume to iso surface rendering by controlling context-preserving volume rendering parameters.

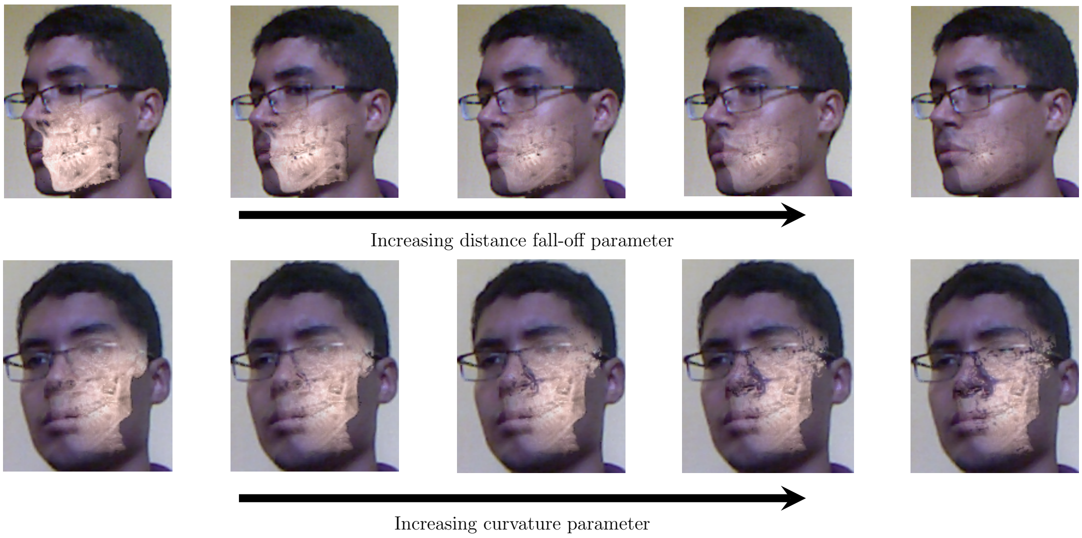


Figure 4.17 Focus+Context visualization to improve human's perception of the augmented scene.

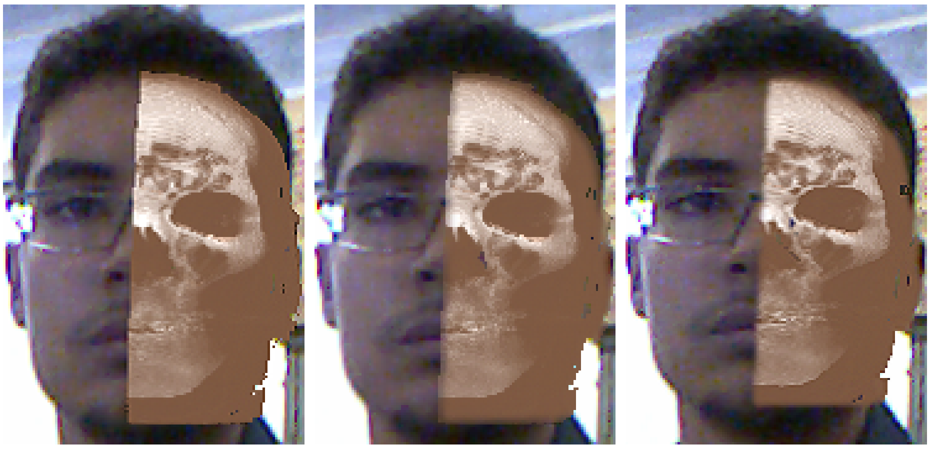


Figure 4.18 Influence of the parameter w_c in smooth contours. (a) $w_c = 0$. (b) $w_c = 2$. (c) $w_c = 4$.

As can be seen in Figure 4.15, our approach supports occlusion at shader level. However, if the occluder overlaps more than 70% of the occludee, tracking may fail.

Influence of context-preserving volume rendering parameters in our application can be seen in Figure 4.16. By increasing the values of the two parameters proposed by Bruckner et al. [Bruckner et al. 2005, Bruckner et al. 2006], which control the exploration of the dataset and the sharpness of the visualization, external structures of the volume (e.g. soft tissue) become increasingly invisible, arising internal structures of the volume (e.g. bone). Support for this kind of rendering is specially important in our application, as a physician does not necessarily want to visualize the naive volume rendering, with soft tissue, organs and bones altogether.

Figure 4.19 Influence of the parameter $w_{grayLevel}$ in visible background on CT data. (a) $w_{grayLevel} = 0$. (b) $w_{grayLevel} = 0.5$ (c) $w_{grayLevel} = 0.75$. (d) $w_{grayLevel} = 1$.

Influence of F+C visualization parameters proposed in CAM technique in our MAR environment can be seen in Figure 4.17. As the distance fall-off increases, transition between real and virtual images becomes smoother and volume less visible. With respect to the curvature term, as it increases, regions of the real scene with high curvature (i.e. nose and glasses in Figure 4.17) remain visible even if they are inside the focus region.

The influence of the parameter w_c in the F+C visualization based on smooth contours can be seen in Figure 4.18. As w_c increases, the transition between the volume and the real scene becomes smoother. At the same time, the volume contours become less visible.

The influence of the parameter $w_{grayLevel}$ in the F+C visualization based on visible background on CT data can be seen in Figure 4.19. In this figure, it is visible that by changing the parameter, the volume can be rendered almost completely invisible, with the soft tissue linearly interpolated with the background scene or with the volume rendered without the background scene.

For the F+C visualization for visible background on MRI data, a comparison between different sampling schemes for ray casting can be seen in Figure 4.20. As the artifacts become more visible during user's movement, this figure shows patient's ROI in some positions and the presence of artifacts in the scenarios as well. By using fixed sampling, the artifacts at the intersection between patient's ROI and clipping plane are more visible when the user rotates his head in front of the sensor. By using adaptive sampling, artifacts are mitigated and the results are comparable to the best scenario, where the fixed step size of the ray is too small. In this case, our method has best performance over the best visual quality scenario, as our approach runs in 28 FPS, while ray casting with small step size achieves 7 FPS only.

In Figure 4.21, we can see more examples of interactions with the F+C visualization based on visible background with MRI data. In the upper clipping, there is a good composition of the medical data in the scene. However, in the lateral clipping, it is visible the 3D reference model clipping and the background rendering because of the shadow cutting present on the final scene.

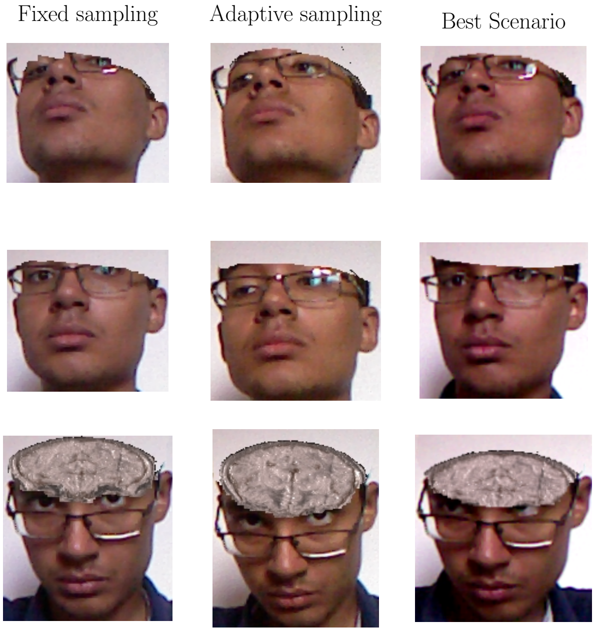


Figure 4.20 Different schemes for F+C visualization based on visible background on MRI Data.

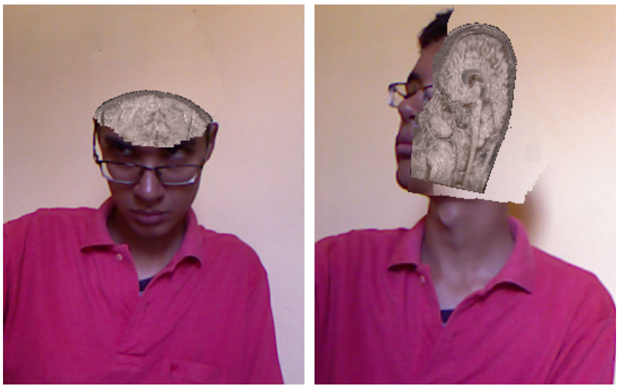


Figure 4.21 Examples of interactions with the F+C visualization based on visible background on MRI Data. Left image: Upper clipping. Right image: Lateral clipping.

4.6 SUMMARY

In this chapter we have presented a marker-free augmented reality approach for on-patient volumetric medical data visualization which supports real-time performance, occlusion and high quality for final rendering through the use of F+C visualization and volume rendering techniques developed to improve quality for final volume rendering. An extensive list of tests were conducted, validating both performance and visual quality for the MAR environment. In this sense, we can take one step further existing state-of-the-art works for on-patient medical data visualization adding support for photorealistic volume rendering based on local illumination components. Thus, we can improve even more the visual quality of the augmented scene without affecting significantly application's performance.

Chapter 5

PHOTOREALISTIC LOCAL ILLUMINATION

In this chapter we present the photorealistic volume rendering solution based on local illumination components proposed in this dissertation. This chapter covers the methods described on Figure 5.1, which require a light probe and a webcam.

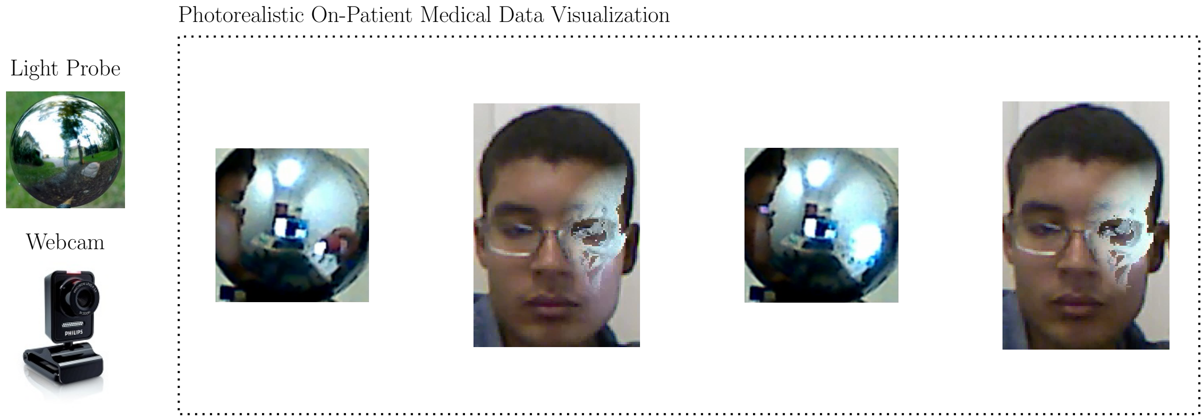


Figure 5.1 Photorealistic on-patient medical data visualization based on local illumination components. Webcam is courtesy of ©Phillips. Light probe is courtesy of www.pearsonfaces.com.

5.1 SPHERICAL HARMONICS

When dealing with photorealistic rendering based on real local illumination components, SH basis is commonly used due to its compact and efficient representation [Sloan 2009]. The SH basis functions are defined by the following equation:

$$\gamma_l^m(\omega) = K_l^m \begin{cases} \sqrt{2} P_l^{-m}(cos\theta) sin(-m\phi) & \text{if } m < 0, \\ \sqrt{2} P_l^m(cos\theta) sin(m\phi) & \text{if } m > 0, \\ P_l^0(cos\theta) & \text{if } m = 0. \end{cases} \tag{5.1}$$

where P_l^m are associated Legendre polynomials, K_l^m is a normalization constant, l is a band index, and $-l \leq m \leq l$ indexes basis functions in band-l.

A spherical function $f(\omega)$ can be projected onto the SH basis, resulting on the following SH coefficient vector:

$$\mathbf{f} = \int_{S^2} f(\omega)\mathbf{y}(\omega)d\omega \tag{5.2}$$

where $\mathbf{y}$ is a vector of SH basis functions. A discretized version of Equation 5.2 is shown as follows:

$$\mathbf{f} \approx \sum_{u,v} f(\omega_{uv})\mathbf{y}(\omega_{uv})d\omega_{uv} \tag{5.3}$$

where $f(\omega_{uv})$ is the environmental lighting captured by the real scene and $d\omega_{uv}$ is defined by:

$$d\omega_{uv} = (2\pi/w)^2 sinc(\theta) \tag{5.4}$$

where w is the resolution of the (cropped) image which contains the environmental lighting and the *sinc* function can be defined by:

$$sinc(x) = \begin{cases} 1 & \text{if } x < 0, \\ \frac{sin(x)}{x} & \text{otherwise.} \end{cases} \tag{5.5}$$

5.2 ESTIMATING LOCAL ENVIRONMENTAL LIGHTING

To lit the medical volume based on photorealistic local illumination, first we need to capture the real-world environmental lighting. Traditionally, this step has been done by using the light probe, a mirror ball which reflects the lighting from the environment [Debevec 1998]. A separate webcam is used to capture the light probe image from the real scene. We do not use the Kinect color camera for such task because it is pointed directly to the user (instead of the light probe), it is not high-definition and we cannot control focus. Then, the use of another webcam can easily overcome such issues. To segment the light probe from the rest of the scene, we constraint the user to center and capture the light probe in a sub-region of resolution w. The sphere image is cropped and the image coordinates are normalized to the interval $(u, v) = [-1, 1] \times [-1, 1]$ and mapped to spherical coordinates:

$$\omega_{uv} = (\theta, \phi) = (arctan(v/u), \pi\sqrt{u^2 + v^2}) \tag{5.6}$$

and cartesian coordinates as well:

$$\omega_{uv} = (x, y, z) = (sin\theta cos\phi, sin\theta sin\phi, cos\theta) \tag{5.7}$$

After light probe capturing, SH basis is used to compactly represent the lighting and its local illumination components (i.e. diffuse and specular terms).

Only nine SH coefficients $\mathbf{f} = (\mathrm{f}_0, \mathrm{f}_1, \ldots, \mathrm{f}_8)$ are sufficient to reconstruct the spherical function with enough accuracy [Ramamoorthi and Hanrahan 2001]. Therefore, $\mathbf{y}(\omega_{uv})$ can be efficiently computed from the cartesian coordinates:

$$\begin{aligned}
\mathrm{y}_0 &= 0.282095 & \mathrm{y}_5 &= 1.092548yz \\
\mathrm{y}_1 &= 0.488603y & \mathrm{y}_6 &= 0.315392(3z^2 - 1) \\
\mathrm{y}_2 &= 0.488603z & \mathrm{y}_7 &= 1.092548xz \\
\mathrm{y}_3 &= 0.488603x & \mathrm{y}_8 &= 0.546274(x^2 - y^2) \\
\mathrm{y}_4 &= 1.092548xy & &
\end{aligned}$$

where the single indexing scheme for the basis coefficient y_i is $i = l(l+1) + m$. As only nine SH coefficients are computed, three orders of SH basis are used, denoting a reconstruction up to band $l = 2$ [Sloan 2009]. Moreover, to optimize performance, the product $\mathbf{y}(\omega_{uv})d\omega_{uv}$ is precomputed.

From the SH coefficients computed from the real-world environmental lighting, the medical volume can be lit according to diffuse and specular terms recovered from the real scene. Diffuse lighting Dl can be computed as follows [Ramamoorthi and Hanrahan 2001]:

$$\begin{aligned}
\mathrm{Dl} = {} & 0.429043\mathrm{f}_8(n_x^2 - n_y^2) + 0.743125\mathrm{f}_6 n_z^2 \\
& + 0.886227\mathrm{f}_0 - 0.247708\mathrm{f}_6 \\
& + 0.858086(\mathrm{f}_4 n_x n_y + \mathrm{f}_7 n_x n_z + \mathrm{f}_5 n_y n_z) \\
& + 1.023328(\mathrm{f}_3 n_x + \mathrm{f}_1 n_y + \mathrm{f}_2 n_z)
\end{aligned} \tag{5.8}$$

where n refers to the normal vector computed for each voxel of the volume being traversed.

To compute the specular lighting Sl, light dominant direction L_d and color L_c must be computed from the SH coefficients [Nowrouzezahrai et al. 2011]:

$$L_d = \overline{\left(\sqrt{\sum_{i=1}^{3} \mathrm{f}_i^2}\right)(-\mathrm{f}_3, -\mathrm{f}_1, \mathrm{f}_2)} \tag{5.9}$$

$$L_c = \left(\sum_{i=1}^{3} cL_d[i]\mathrm{f}_i\right) / \left(\sum_{i=1}^{3} (cL_d[i])^2\right) \tag{5.10}$$

where $c = 1.44395$ and $L_d[i]$ is an indexing scheme for the given vector, treating each cartesian coordinate $[x, y, z]$ as a position $[0, 1, 2]$.

As $f(\omega_{uv})$ is captured from light probe image, it is in fact a trichromatic vector. Therefore, it can be conveniently redefined as $\mathbf{f}(\omega_{uv}) = (f(\omega_{uv})^r, f(\omega_{uv})^g, f(\omega_{uv})^b)$, the spherical functions for each color channel storing nine SH coefficients. By computing Equation 5.9, we will obtain a trichromatic L_d. However, the dominant light direction must be unique for the scene. So, standard grayscale conversion can be employed to result in a single light direction. In the opposite case, each color channel for L_c is computed from Equation 5.10 as the SH coefficients $\mathbf{f}$ are trichromatic. Sl is computed by using L_d and L_c in traditional Blinn-Phong shading. Final illumination for a given voxel on the volume can be easily computed from Equation 4.1 as $q(s) = q_{ea}(s) + \mathrm{Dl}(n) + \mathrm{Sl}(s, n)$

By capturing the LDR image of the light probe, we have a loss in quality computing Dl and Sl when compared to solutions which use HDR image. To minimize this problem, we normalize the range $[0, 255]$ for each color channel to $[0, 1]$ and apply the most simple reverse tone-mapping operator, linearization, over the LDR image to convert it to HDR content. The linearization technique uses inverse gamma correction over the normalized LDR image to compensate for non-linearities in the LDR representation. A visual comparison between using LDR, linearized LDR and HDR images for computing diffuse and specular real lighting components can be seen in Figure 5.2. It is visible that by using a reverse tone mapping operator, most of the original appearance lost in the HDR-to-LDR conversion process is recovered.

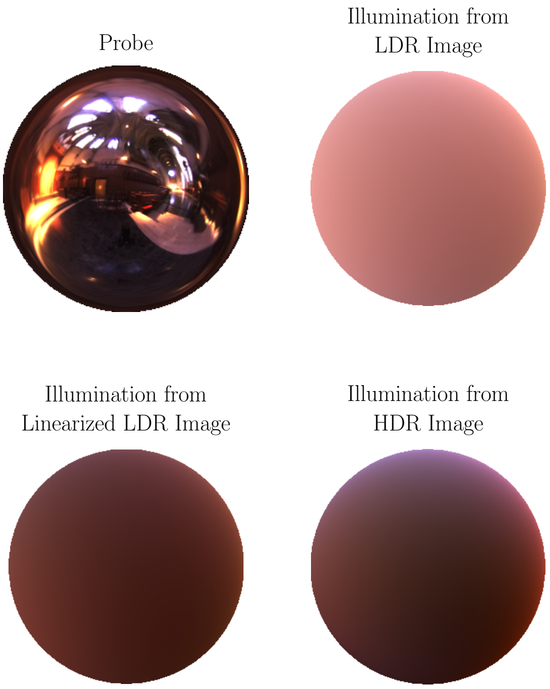


Figure 5.2 Visual comparison between using LDR, linearized LDR and HDR probe images when used for computing SH coefficients to lit a virtual sphere.

5.3 RESULTS AND DISCUSSION

In this section we analyze visual quality and performance of the whole approach with focus on the impact of using photorealistic rendering. Light probe used is a common mirrored sphere and Philips SPC530NC was used as webcam to capture light probe image. Medical volumes and PC configuration are the same as used for the other evaluations listed in this dissertation.

5.3.1 Performance Evaluation

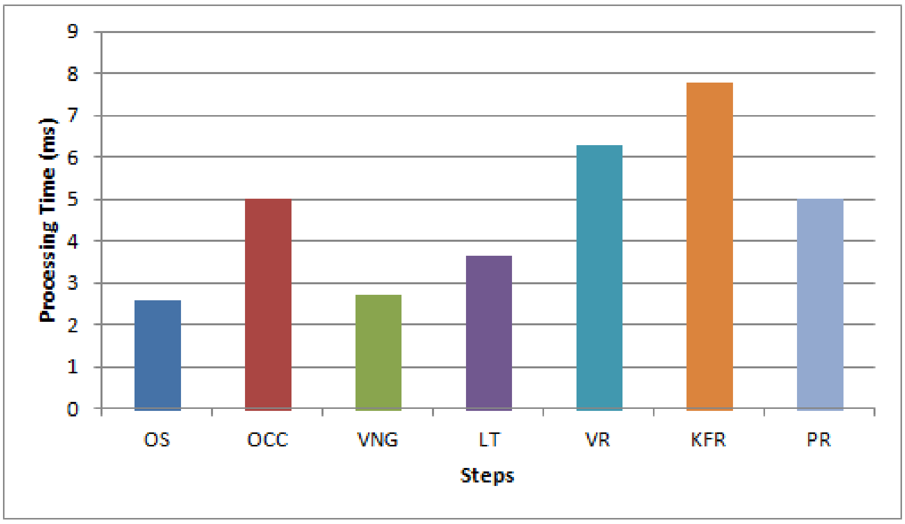


Figure 5.3 Performance results measured in average milliseconds for each step of our approach. OS - Other Steps (i.e. display timing, Kinect latency), OCC - Occlusion Computation, VNG - Vertex and Normal Map Generation, LT - Live Tracking, VR - Volume Rendering, KFR - KinectFusion's grid raycasting, PR - Photorealistic Rendering (i.e. diffuse and specular terms capturing from light probe image). Times were measured running our approach with the KinectFusion's grid in resolution 512^3 and light probe image of resolution $w = 128$.

Figure 5.3 shows the average time measured for each step of the on-patient medical data visualization application. It takes ≈ 33ms per frame, therefore running in real-time (30 FPS). Volume Rendering (VR) is one of the steps which takes most time to be processed because, for photorealistic rendering to make sense, we have to enable local illumination on the volume, which is already known to decrease application's performance. Photorealistic rendering (PR) in which the light probe image is cropped, normalized and from there it is computed the specular and diffuse lighting components based on SH coefficients takes $5ms$ for a resolution of 128^2 pixels of the cropped image.

We have also evaluated the performance impact of the cropped image resolution w. The results can be seen in Figure 5.4. In this figure, we evaluated the influence of w for only two scenarios: DVR with local illumination and transfer function. As can be seen from Figure 4.11, we have a real changing on application's performance when activating those modes, because of the number of trilinear interpolations performed. The timing for other modes (e.g. MIP, iso-surface, context-preserving rendering) can be calculated from the results given in Figures 5.4 and 4.11. From Figure 5.4 it is visible that by using resolutions such as $w = 192$ and $w = 256$ we start to decrease significantly application's performance, however it still runs in real-time (above 20 FPS), maintaning user's interactivity with the application. Moreover, from this evaluation we have found appropriate to use $w = 128$ because we have a balance in terms of performance (achieving full real-time when using only local illumination, and above 25 FPS when activating transfer function) and visual accuracy as will be shown in the next subsection.

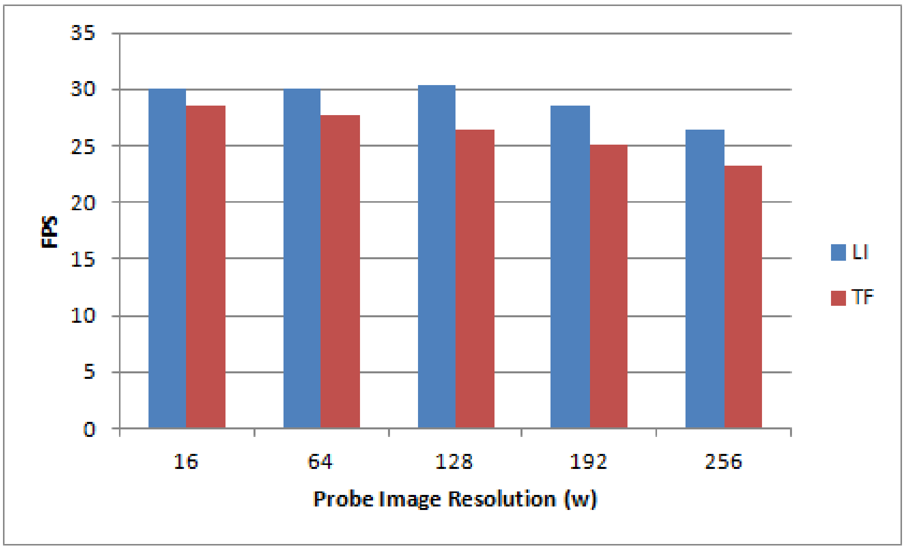


Figure 5.4 Performance results measured in frames per second for different light probe image resolutions. LI - DVR + Local Illumination (required for photorealistic volume rendering). TF - DVR + LI + Transfer Functions.

The performance of our approach for each F+C visualization technique mentioned in this paper added with the effect of photorealistic volume rendering can be seen in Figure 5.5. All of the techniques run above 25 FPS, as most of their processing is done on shader and adds negligible overhead for the application. One exception of this situation occurs for visible background on MRI data, which is already known to be slower than the other approaches (Figure 4.14).

5.3.2 Visual Quality Evaluation

SH/w	256	192	128	64	16
1	1.342	1.342 (99.99%)	1.340 (99.87%)	1.338 (99.75%)	1.328 (98.95%)
2	0.169	0.170 (99.43%)	0.172 (98.25%)	0.173 (98.01%)	0.176 (96.27%)
3	0.023	0.023 (99.51%)	0.022 (93.63%)	0.025 (93.80%)	0.049 (48.30%)
4	0.025	0.025 (99.01%)	0.025 (97.54%)	0.022 (85.56%)	0.019 (76.52%)
5	0.021	0.022 (96.75%)	0.022 (94.44%)	0.026 (82.14%)	0.037 (56.61%)
6	-0.042	-0.041 (97.88%)	-0.040 (94.12%)	-0.037 (87.83%)	-0.025 (60.03%)
7	-0.077	-0.077 (99.68%)	-0.079 (97.48%)	-0.083 (92.65%)	-0.102 (75.60%)
8	-0.036	-0.035 (97.74%)	-0.033 (92.66%)	-0.029 (81.03%)	0.008 (-23.43%)
9	-0.018	-0.017 (93.94%)	-0.015 (85.54%)	-0.009 (52.88%)	0.029 (-61.91%)
ARA		**98.22%**	**94.84%**	**85.96%**	**47.44%**

Table 5.1 Accuracy measurements for each one of the nine SH coefficients computed from a synthetic light probe of different resolutions. ARA - Average Relative Accuracy.

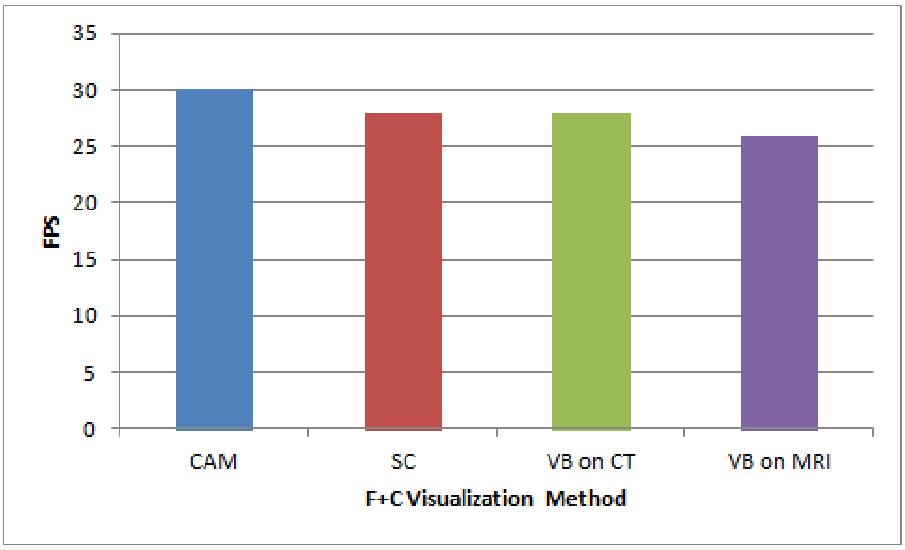


Figure 5.5 Performance results measured in FPS for each one of the F+C visualization techniques discussed in this paper. CAM refers to the technique proposed in [Bichlmeier et al. 2007]. SC - Smooth Contours, VB on CT - Visible Background on CT data, VB on MRI - Visible Background on MRI Data. Times were measured running our approach with the KinectFusion's grid in resolution 512^3 and light probe image of resolution $w = 128$.

SH/w	256	192	128	64	16
1	0.404	0.402 (99.53%)	0.404 (99.99%)	0.404 (99.93%)	0.412 (98.18%)
2	0.014	0.014 (99.66%)	0.015 (95.17%)	0.017 (84.26%)	0.042 (34.28%)
3	0.064	0.064 (99.65%)	0.064 (99.11%)	0.061 (95.04%)	0.050 (78.54%)
4	-0.006	-0.007 (84.53%)	-0.009 (65.30%)	-0.015 (40.28%)	-0.045 (13.35%)
5	0.013	0.014 (93.89%)	0.016 (82.42%)	0.021 (63.46%)	0.046 (29.00%)
6	-0.053	-0.052 (97.87%)	-0.050 (94.85%)	-0.043 (81.61%)	0.003 (-7.42%)
7	0.001	0.001 (66.36%)	0.000 (2.18%)	-0.004 (-42.18%)	-0.027 (-7.31%)
8	-0.069	-0.069 (99.25%)	-0.071 (97.09%)	-0.074 (92.90%)	-0.086 (79.89%)
9	-0.262	-0.260 (99.36%)	-0.259 (99.17%)	-0.255 (97.52%)	-0.229 (-87.72%)
ARA		**93.34%**	**81.70%**	**68.09%**	**45.13%**

Table 5.2 Accuracy measurements for each one of the nine SH coefficients computed from a real light probe of different resolutions. ARA - Average Relative Accuracy.

As can be seen in Figure 5.3, light probe resolution has a significant impact on final application's performance. Therefore, to choose the most appropriate resolution, we have conducted a visual quality evaluation between different light probe resolutions and its impact on photorealistic rendering. As shown in Figure 5.6, a virtual sphere was lit from local illumination components computed from synthetic and real light probes of several resolutions. The numerical accuracy for synthetic and real scenarios can be seen in Tables 5.1 and 5.2. Compared to the scenario of highest resolution available to enable

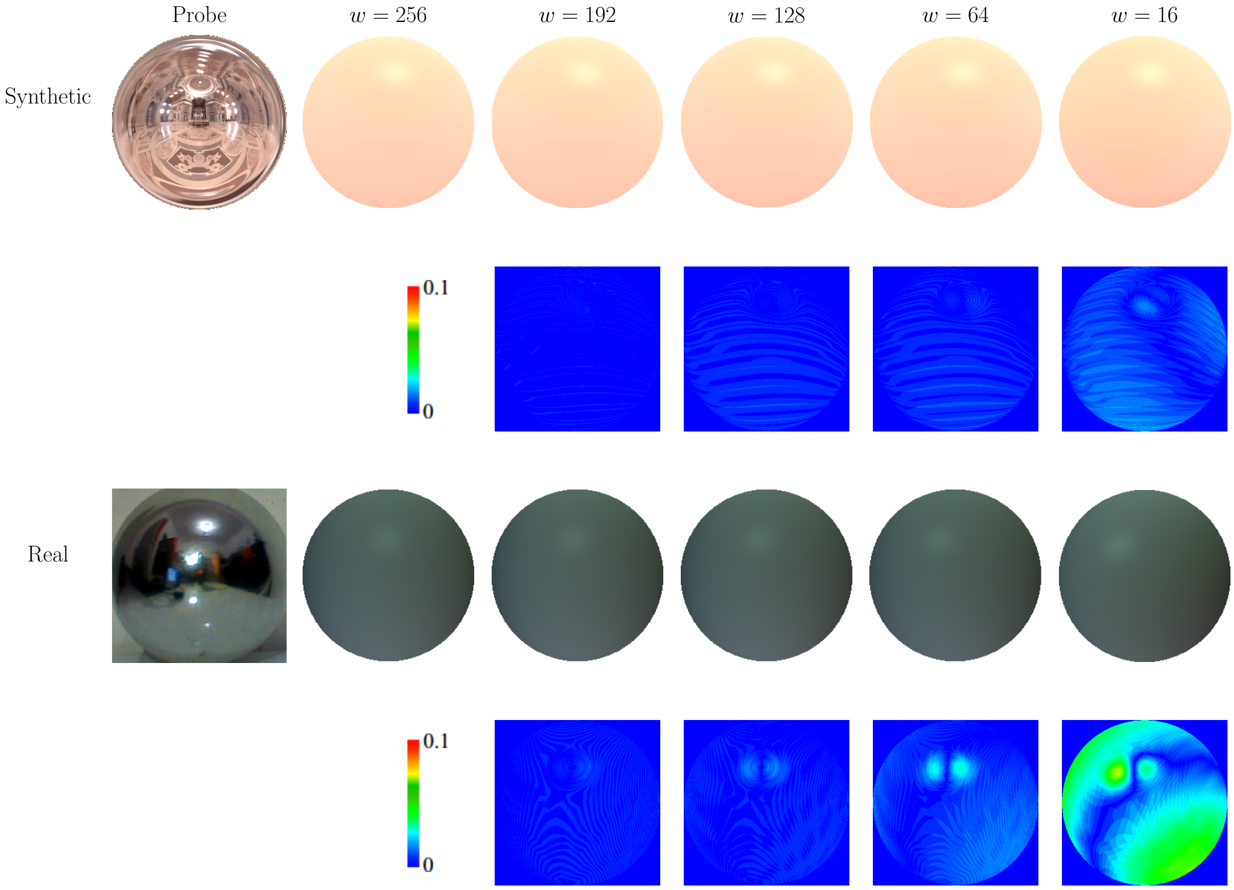


Figure 5.6 A visual comparison between different real and synthetic light probe resolutions and its impact on the illumination of a virtual sphere.

Figure 5.7 Photorealistic on-patient medical data visualization based on local illumination components.

still real-time performance ($w = 256$), by using resolutions of 192 and 128, we still have an accurate solution. However, for lower resolutions of 64 and 16, accuracy decreases specially for the real situation. Thus, to optimize performance, we can use lower light probe resolutions, such as $w = 192$ or $w = 128$, without losing too much visual quality on the final rendering. This statement is assured by the visual color-coded error of Figure 5.6. While for the synthetic scenario there is little error in the final image (even for $w = 16$), for the real scenario, the difference is main present on specular lobe position for resolutions of $w = 128$ and $w = 64$, but, for the lowest resolution, diffuse component deviates considerably in relation the highest resolution scenario.

In terms of photorealistic on-patient medical data visualization, the methods used in this dissertation can compute diffuse and dominant specular terms for several challenging illumination conditions, as can be seen in Figure 5.7. Thereby, allowing a consistent, seamlessly integration of the virtual medical volume into the real scene.

5.4 SUMMARY

In this chapter we have presented a markerless augmented reality integrated solution for on-patient medical data visualization based on photorealistic local illumination. By using a light probe to capture dynamic environmental lighting, the approach supports photorealistic volume rendering based on diffuse and specular terms computed in real-time. From tests conducted, we have shown that the photorealistic volume rendering improves visual quality while application's performance is still real-time.

Chapter 6

CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

We believe that the integrated solution proposed in this dissertation highlights the maturity of several fields: markerless tracking, which is currently accurate and optimized for real-time performance; on-patient medical data visualization, where medical volumes can be rendered in real-time and with high quality based on the standard techniques proposed in the field of volume rendering. Moreover, for the augmented scene, F+C techniques have shown their usefulness by their different ways to separate focus from context regions to be visualized; photorealistic rendering, in which by using a light probe image captured from the real scene, the lightweight SH basis representation allows efficient and accurate computation of coefficients for extraction of diffuse and specular terms from the real dynamic environmental lighting.

This dissertation has proposed the first markerless augmented reality environment for on-patient medical data visualization based on photorealistic local illumination and focus + context visualization. Markerless tracking is achieved by reconstructing a 3D reference model of the patient's ROI from KinectFusion algorithm and tracking it during live stream with the ICP algorithm. Photorealistic volume rendering is performed by estimating specular and diffuse components from spherical harmonics coefficients computed from a mirrored sphere image. Finally, our approach supports four focus + context visualization techniques which improve the visual quality of the final rendered image.

From an evaluation of the proposed solution, we conclude that it runs in full real-time on typical medical datasets, provides high visual quality for the final augmented scene through the use of focus + context visualization and photorealistic rendering. Moreover, it provides accuracy enough for applications that need good ”visual” accuracy for the registration (i.e. good composition and tracking of the virtual object into the augmented scene). Experiments must still be conducted to evaluate if the accuracy achieved is enough for medical AR applications to aid surgery, for instance.

6.2 FUTURE WORK

Our approach does not support relocalization and tracking fails if patient's ROI is not visible in the scene. Therefore, it is possible to improve tracking robustness by solving these issues. Moreover, the markerless tracking proposed in this dissertation was recently extended to support interactive non-rigid tracking in the co-authored publication [Souza et al. 2014]. To achieve the best performance, it would be interesting to use two GPUs to run rigid and non-rigid tracking separately.

One of the current limitations of the proposed approach is that it was not evaluated with a real patient, which would enable us to validate our approach in terms of accuracy. Thus, one of the next steps must be the adaptation of the approach to be used and evaluated in a real medical environment.

For the augmented reality environment, we have used a conventional display to show the augmented scene. Multi-view solutions based on AR glasses, such as Google Glass, or portable solutions based on mobile devices, such as iPad, can be employed by processing the proposed approach on a server and transferring the visualization of the augmented content for those alternative hardwares, allowing a more seamlessly visualization of the virtual content onto the real scene.

Specifically for the techniques employed, F+C visualization based on visible background techniques do not support the visualization of real dynamic background scenes. A multi-view approach, in which an additional webcam is used to capture real background scene, may solve this problem. Photorealistic rendering based on a light probe has its own limitations, as the light probe is intrusive in the scene and it must be located somewhere near the volume which will be illuminated. To solve this issue, a probeless technique which computes local illumination components based on the LDR content could be employed [Karsch et al. 2011, Karsch et al. 2014, Gruber et al. 2014]. However, the current state-of-the-art works still do not support real-time performance for dynamic lighting environments.

For future work, we can extend the photorealism of the proposed approach to take into account global illumination effects into the volume rendering. Moreover, an improved LDR to HDR algorithm could be applied to improve the quality of the image captured as well as the accuracy of the diffuse and specular terms computation. In this sense, real-time LDR to HDR algorithms must be studied and evaluated in the context of an AR application to validate its applicability.

BIBLIOGRAPHY

[Abellan et al. 2008] Abellan, P., Puig, A., and Ayala, D. (2008). Focus + context rendering of structured biomedical data. In *VCBM'08*, pages 109–116.

[Akenine-Moller et al. 2002] Akenine-Moller, T., Moller, T., and Haines, E. (2002). *Real-Time Rendering.* A. K. Peters, Ltd., Natick, MA, USA, 2nd edition.

[Avery et al. 2009] Avery, B., Sandor, C., and Thomas, B. (2009). Improving spatial perception for augmented reality x-ray vision. In *Virtual Reality Conference, 2009. VR 2009. IEEE*, pages 79–82.

[Azuma et al. 2001] Azuma, R., Baillot, Y., Behringer, R., Feiner, S., Julier, S., and MacIntyre, B. (2001). Recent advances in augmented reality. *IEEE Comput. Graph. Appl.*, 21(6):34–47.

[Azuma 1995] Azuma, R. T. (1995). *Predictive Tracking for Augmented Reality.* PhD thesis, Chapel Hill, NC, USA. UMI Order No. GAX95-38370.

[Azuma 1997] Azuma, R. T. (1997). A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355–385.

[Banterle et al. 2013] Banterle, F., Chalmers, A., and Scopigno, R. (2013). Real-time high fidelity inverse tone mapping for low dynamic range content. In Miguel A. Otaduy, O. S., editor, *Eurographics 2013 Short Papers*, pages 41–44. Eurographics, Eurographics.

[Besl and McKay 1992] Besl, P. and McKay, H. (1992). A method for registration of 3-d shapes. *PAMI*, 14(2):239 –256.

[Besl 1988] Besl, P. J. (1988). Active, optical range imaging sensors. *Mach. Vision Appl.*, 1:127–152.

[Bichlmeier 2010] Bichlmeier, C. (2010). *Immersive, Interactive and Contextual In-Situ Visualization for Medical Applications.* Dissertation, Technische Universität München, München.

[Bichlmeier et al. 2007] Bichlmeier, C., Wimmer, F., Heining, S. M., and Navab, N. (2007). Contextual anatomic mimesis hybrid in-situ visualization method for improving multi-sensory depth perception in medical augmented reality. In *International Symposium on Mixed and Augmented Reality*, ISMAR '07, pages 1–10, Washington, DC, USA. IEEE.

[Blinn 1977] Blinn, J. F. (1977). Models of light reflection for computer synthesized pictures. SIGGRAPH '77, pages 192–198, New York, NY, USA. ACM.

[Blinn 1994] Blinn, J. F. (1994). Jim Blinn's corner: Compositing. 1. Theory. *IEEE Computer Graphics and Applications*, 14(5):83–87.

[Blum et al. 2012] Blum, T., Kleeberger, V., Bichlmeier, C., and Navab, N. (2012). mirracle: Augmented reality in-situ visualization of human anatomy using a magic mirror. In *VR, 2012 IEEE*, pages 169–170.

[Bouaziz et al. 2013] Bouaziz, S., Tagliasacchi, A., and Pauly, M. (2013). Sparse iterative closest point. *Computer Graphics Forum (Symposium on Geometry Processing)*, 32(5):1–11.

[Breiman 2001] Breiman, L. (2001). Random forests. In *Machine Learning*, pages 5–32.

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

[Bruckner et al. 2005] Bruckner, S., Grimm, S., Kanitsar, A., and Gröller, M. E. (2005). Illustrative context-preserving volume rendering. In *Proceedings of the Seventh Joint Eurographics / IEEE VGTC Conference on Visualization*, EUROVIS'05, pages 69–76, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

[Burrus 2014] Burrus, N. (2014). Kinect calibration. `http://nicolas.burrus.name/index.php/Research/KinectCalibration`. Accessed 02 November 2014.

[Card et al. 1999] Card, S. K., Mackinlay, J. D., and Shneiderman, B., editors (1999). *Readings in Information Visualization: Using Vision to Think.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[Chen et al. 2010] Chen, J., Granier, X., Lin, N., and Peng, Q. (2010). On-line visualization of underground structures using context features. In *Proceedings of the 17th ACM Symposium on Virtual Reality Software and Technology*, VRST '10, pages 167–170, New York, NY, USA. ACM.

[Chen and Medioni 1992] Chen, Y. and Medioni, G. (1992). Object modelling by registration of multiple range images. *Image and Vision Computing*, 10(3):145 – 155.

[Cruz et al. 2012] Cruz, L., Lucio, D., and Velho, L. (2012). Kinect and rgbd images: Challenges and applications. In *Graphics, Patterns and Images Tutorials (SIBGRAPI-T), 2012 25th SIBGRAPI Conference on*, pages 36–49.

[Curless and Levoy 1996] Curless, B. and Levoy, M. (1996). A volumetric method for building complex models from range images. SIGGRAPH '96, pages 303–312, New York, NY, USA. ACM.

[Debarba et al. 2012] Debarba, H. G., Grandi, J., Maciel, A., and Zanchet, D. (2012). Anatomic hepatectomy planning through mobile display visualization and interaction. In *MMVR*, volume 173, pages 111–115. IOS Press.

[Debevec 1998] Debevec, P. (1998). Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, pages 189–198, New York, NY, USA. ACM.

[Debevec and Malik 1997] Debevec, P. E. and Malik, J. (1997). Recovering high dynamic range radiance maps from photographs. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '97, pages 369–378, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.

[dos Santos et al. 2010] dos Santos, T. R., Seitel, A., Meinzer, H.-P., and Maier-Hein, L. (2010). Correspondences search for surface-based intra-operative registration. MICCAI'10, pages 660–667, Berlin, Heidelberg. Springer-Verlag.

[Engel et al. 2001] Engel, K., Kraus, M., and Ertl, T. (2001). High-quality pre-integrated volume rendering using hardware-accelerated pixel shading. HWWS '01, pages 9–16, New York, NY, USA. ACM.

[Fanelli et al. 2011] Fanelli, G., Weise, T., Gall, J., and Gool, L. V. (2011). Real time head pose estimation from consumer depth cameras. In *Proceedings of the 33rd International Conference on Pattern Recognition*, DAGM'11.

[Freund and Schapire 1995] Freund, Y. and Schapire, R. E. (1995). A decision-theoretic generalization of on-line learning and an application to boosting. In *Proceedings of the Second European Conference on Computational Learning Theory*, EuroCOLT '95, pages 23–37, London, UK, UK. Springer-Verlag.

[Gasteiger et al. 2011] Gasteiger, R., Neugebauer, M., Beuing, O., and Preim, B. (2011). The flowlens: A focus-and-context visualization approach for exploration of blood flow in cerebral aneurysms. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2183–2192.

[Gruber et al. 2014] Gruber, L., Langlotz, T., Sen, P., Hoherer, T., and Schmalstieg, D. (2014). Efficient and robust radiance transfer for probeless photorealistic augmented reality. In *IEEE Virtual Reality, VR*, pages 15–20.

[Hadwiger et al. 2006] Hadwiger, M., Kniss, J. M., Rezk-salama, C., Weiskopf, D., and Engel, K. (2006). *Real-time Volume Graphics*. A. K. Peters, Ltd., USA.

[Hadwiger et al. 2009] Hadwiger, M., Ljung, P., Salama, C. R., and Ropinski, T. (2009). Advanced illumination techniques for gpu-based volume raycasting. SIGGRAPH '09, pages 2:1–2:166, New York, NY, USA. ACM.

[Hansard et al. 2013] Hansard, M. E., Lee, S., Choi, O., and Horaud, R. (2013). *Time-of-Flight Cameras - Principles, Methods and Applications.* Springer Briefs in Computer Science. Springer.

[Holzer et al. 2012] Holzer, S., Rusu, R., Dixon, M., Gedikli, S., and Navab, N. (2012). Adaptive neighborhood selection for real-time surface normal estimation from organized point cloud data using integral images. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 2684–2689.

[Horn and Schunck 1981] Horn, B. K. and Schunck, B. G. (1981). Determining optical flow. *Artificial Intelligence*, 17(1–3):185 – 203.

[Izadi et al. 2011] Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., and Fitzgibbon, A. (2011). Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. UIST '11, pages 559–568, USA. ACM.

[Kalkofen et al. 2013] Kalkofen, D., Veas, E., Zollmann, S., Steinberger, M., and Schmalstieg, D. (2013). Adaptive ghosted views for augmented reality. In *ISMAR*, Adelaide, SA, Australia. IEEE Computer Society.

[Karsch et al. 2011] Karsch, K., Hedau, V., Forsyth, D., and Hoiem, D. (2011). Rendering synthetic objects into legacy photographs. *ACM Trans. Graph.*, 30(6):157:1–157:12.

[Karsch et al. 2014] Karsch, K., Sunkavalli, K., Hadap, S., Carr, N., Jin, H., Fonte, R., Sittig, M., and Forsyth, D. (2014). Automatic scene inference for 3d object compositing. *ACM Trans. Graph.*, 33(3):32:1–32:15.

[Kato and Billinghurst 1999] Kato, H. and Billinghurst, M. (1999). Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *Augmented Reality, 1999. (IWAR '99) Proceedings. 2nd IEEE and ACM International Workshop on*, pages 85 –94.

[Kilgus et al. 2014] Kilgus, T., Heim, E., Haase, S., Prüfer, S., Müller, M., Seitel, A., Fangerau, M., Wiebe, T., Iszatt, J., Schlemmer, H.-P., Hornegger, J., Yen, K., and Maier-Hein, L. (2014). Mobile markerless augmented reality and its application in forensic medicine. *International Journal of Computer Assisted Radiology and Surgery*, pages 1–14.

[Kirmizibayrak et al. 2014] Kirmizibayrak, C., Wakid, M., Yim, Y., Hristov, D., and Hahn, J. K. (2014). Interactive focus + context medical data exploration and editing. *Computer Animation and Virtual Worlds*, 25(2):129–141.

[Knecht et al. 2012] Knecht, M., Traxler, C., Mattausch, O., and Wimmer, M. (2012). Augmented reality: Reciprocal shading for mixed reality. *Comput. Graph.*, 36(7):846–856.

[Kovaleski and Oliveira 2014] Kovaleski, R. and Oliveira, M. (2014). High-Quality Reverse Tone Mapping for a Wide Range of Exposures. In *Conference on Graphics, Patterns and Images*, SIBGRAPI, Brazil.

[Koyama 2009] Koyama, T. (2009). Introduction to flartoolkit. *Adobe System Incorporated.*

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

[Kruger and Westermann 2003] Kruger, J. and Westermann, R. (2003). Acceleration techniques for gpu-based volume rendering. In *Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, VIS '03, pages 38–, Washington, DC, USA. IEEE Computer Society.

[Kutter et al. 2008] Kutter, O., Aichert, A., Bichlmeier, C., Traub, J., Heining, S. M., Ockert, B., Euler, E., and Navab, N. (2008). Real-time Volume Rendering for High Quality Visualization in Augmented Reality. In *AMI-ARCS 2008*, New York, USA. MICCAI Society.

[Lee et al. 2005] Lee, C. H., Varshney, A., and Jacobs, D. W. (2005). Mesh saliency. *ACM Trans. Graph.*, 24(3):659–666.

[Lee et al. 2012] Lee, J.-D., Huang, C.-H., Huang, T.-C., Hsieh, H.-Y., and Lee, S.-T. (2012). Medical augment reality using a markerless registration framework. *Expert Syst. Appl.*, 39(5):5286–5294.

[Levoy 1988] Levoy, M. (1988). Display of surfaces from volume data. *IEEE Comput. Graph. Appl.*, 8(3):29–37.

[Li et al. 2003] Li, W., Mueller, K., and Kaufman, A. (2003). Empty space skipping and occlusion clipping for texture-based volume rendering. In *VIS 2003*, pages 317–324. IEEE Computer Society.

[Lucas and Kanade 1981] Lucas, B. D. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'81, pages 674–679, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Macedo et al. 2014a] Macedo, M., Almeida, C., Souza, A., Silva, J., Apolinario, A., and Giraldi, G. (2014a). A Markerless Augmented Reality Environment for Medical Data Visualization. In *Workshop de Informatica Medica*, WIM, Brazil.

[Macedo and Apolinario 2014a] Macedo, M. and Apolinario, A. (2014a). Focus + Context Visualization based on Volume Clipping for Markerless On-Patient Medical Data Visualization. *Computers & Graphics*. Manuscript submitted for publication.

[Macedo et al. 2013a] Macedo, M., Apolinario, A., and Souza, A. (2013a). A robust real-time face tracking using head pose estimation for a markerless ar system. In *Virtual and Augmented Reality (SVR), 2013 XV Symposium on*, pages 224–227.

[Macedo et al. 2014b] Macedo, M., Apolinario, A., Souza, A. C., and Giraldi, G. A. (2014b). High-Quality On-Patient Medical Data Visualization in a Markerless Augmented Reality Environment. *Journal on 3D Interactive Systems*. (in press).

[Macedo et al. 2013b] Macedo, M., Apolinario, A. L., and Souza, A. C. (2013b). A Markerless Augmented Reality Approach Based on Real-Time 3D Reconstruction using Kinect. In *Workshop of Works in Progress (WIP) in SIBGRAPI*, Arequipa, Peru.

[Macedo et al. 2013c] Macedo, M., Apolinario, A. L., and Souza, A. C. (2013c). Kinect-Fusion for Faces: Real-Time 3D Face Tracking and Modeling Using a Kinect Camera for a Markerless AR System. *Journal on 3D Interactive Systems*, 4:2–7.

[Macedo et al. 2014c] Macedo, M. C., Apolinario, A. L., Souza, A. C., and Giraldi, G. A. (2014c). A semi-automatic markerless augmented reality approach for on-patient volumetric medical data visualization. In *Virtual and Augmented Reality (SVR), 2014 XVI Symposium on*, pages 63–70.

[Macedo and Apolinario 2014b] Macedo, M. C. d. F. and Apolinario, A. L. (2014b). Improving on-patient medical data visualization in a markerless augmented reality environment by volume clipping. In *Graphics, Patterns and Images (SIBGRAPI), 2014 27th SIBGRAPI Conference on*, pages 149–156.

[Maier-Hein et al. 2011] Maier-Hein, L., Franz, A. M., Fangerau, M., Schmidt, M., Seitel, A., Mersmann, S., Kilgus, T., Groch, A., Yung, K., dos Santos, T. R., and Meinzer, H.-P. (2011). Towards mobile augmented reality for on-patient visualization of medical images. In *Bildverarbeitung für die Medizin*, Informatik Aktuell, pages 389–393. Springer.

[Maier-Hein et al. 2010] Maier-Hein, L., Schmidt, M., Franz, A. M., dos Santos, T. R., Seitel, A., Jähne, B., Fitzpatrick, J. M., and Meinzer, H. P. (2010). Accounting for anisotropic noise in fine registration of time-of-flight range data with high-resolution surface data. In *Proceedings of the 13th international conference on Medical image computing and computer-assisted intervention: Part I*, MICCAI'10, pages 251–258, Berlin, Heidelberg. Springer-Verlag.

[Marr and Poggio 1979] Marr, D. and Poggio, T. (1979). A Computational Theory of Human Stereo Vision. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 204(1156):301–328.

[Meilland et al. 2013] Meilland, M., Barat, C., and Comport, A. (2013). 3d high dynamic range dense visual slam and its application to real-time object re-lighting. In *Mixed and Augmented Reality (ISMAR), 2013 IEEE International Symposium on*, pages 143–152.

[Meister et al. 2012] Meister, S., Izadi, S., Kohli, P., Hämmerle, M., Rother, C., and Kondermann, D. (2012). When can we use kinectfusion for ground truth acquisition? In *IROS*. IEEE Computer Society.

[Mendez et al. 2010] Mendez, E., Feiner, S., and Schmalstieg, D. (2010). Focus and context in mixed reality by modulating first order salient features. SG'10, pages 232–243, Berlin, Heidelberg. Springer-Verlag.

[Mendez and Schmalstieg 2009] Mendez, E. and Schmalstieg, D. (2009). Importance masks for revealing occluded objects in augmented reality. In *Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology*, VRST '09, pages 247–248, New York, NY, USA. ACM.

[Meng et al. 2013] Meng, M., Fallavollita, P., Blum, T., Eck, U., Sandor, C., Weidert, S., Waschke, J., and Navab, N. (2013). Kinect for interactive ar anatomy learning. In *Mixed and Augmented Reality (ISMAR), 2013 IEEE International Symposium on*, pages 277–278.

[Mercier-Ganady et al. 2014] Mercier-Ganady, J., Lotte, F., Loup-escande, E., Marchal, M., and Lecuyer, A. (2014). The mind-mirror: See your brain in action in your head using eeg and augmented reality. In *Virtual Reality (VR), 2014 iEEE*, pages 33–38.

[Microsoft 2014] Microsoft (2014). Kinect face tracking sdk. `http://msdn.microsoft.com/en-US/enus/library/jj130970.aspx`. Accessed 01 November 2014.

[National Library of Medicine 2014] National Library of Medicine (2014). Visible human data. `http://www.nlm.nih.gov/research/visible/`. Accessed 22 September 2014.

[Nowrouzezahrai et al. 2011] Nowrouzezahrai, D., Geiger, S., Mitchell, K., Sumner, R., Jarosz, W., and Gross, M. (2011). Light factorization for mixed-frequency shadows in augmented reality. In *Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality*, ISMAR '11, pages 173–179, Washington, DC, USA. IEEE Computer Society.

[Osher and Fedkiw 2002] Osher, S. J. and Fedkiw, R. P. (2002). *Level Set Methods and Dynamic Implicit Surfaces*. Springer, 2003 edition.

[Padilha et al. 2013] Padilha, A., Rolim, C., and Teichrieb, V. (2013). The ghosting technique applied to augmented reality visualization. In *Virtual and Augmented Reality (SVR), 2013 XV Symposium on*, pages 159–166.

[Peasley and Birchfield 2013] Peasley, B. and Birchfield, S. (2013). Replacing projective data association with lucas-kanade for kinectfusion. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 638–645.

[Pessoa et al. 2010] Pessoa, S., Moura, G., Lima, J., Teichrieb, V., and Kelner, J. (2010). Photorealistic rendering for augmented reality: A global illumination and brdf solution. In *Virtual Reality Conference (VR), 2010 IEEE*, pages 3–10.

[Pessoa et al. 2012] Pessoa, S. A., de S. Moura, G., Lima, J. P. S. M., Teichrieb, V., and Kelner, J. (2012). Rpr-sors: Real-time photorealistic rendering of synthetic objects into real scenes. *Computers and Graphics*, 36(2):50–69.

[Posdamer and Altschuler 1982] Posdamer, J. L. and Altschuler, M. D. (1982). Surface measurement by space-encoded projected beam systems. *Computer Graphics and Image Processing*, 18(1):1–17.

[Prokop et al. 1997] Prokop, M., Shion, H. O., Schanz, A., and Andschaefer-Prokop, M. (1997). Use of maximum intensity projections in ct angiography. In *Radiographics*, volume 17, pages 433–451.

[Ramamoorthi and Hanrahan 2001] Ramamoorthi, R. and Hanrahan, P. (2001). An efficient representation for irradiance environment maps. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pages 497–500, New York, NY, USA. ACM.

[Reinhard et al. 2005] Reinhard, E., Ward, G., Pattanaik, S., and Debevec, P. (2005). *High Dynamic Range Imaging: Acquisition, Display, and Image-Based Lighting (The Morgan Kaufmann Series in Computer Graphics)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[Roettger et al. 2003] Roettger, S., Guthe, S., Weiskopf, D., Ertl, T., and Strasser, W. (2003). Smart hardware-accelerated volume rendering. In *Proceedings of the symposium on Data visualisation 2003*, VISSYM '03, pages 231–238, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

[Röttger et al. 2012] Röttger, D., Merhof, D., and Müller, S. (2012). The BundleExplorer: A Focus and Context Rendering Framework for Complex Fiber Distributions. In *Proceedings of the Eurographics Conference on Visual Computing for Biology and Medicine*, EG VCBM'12, pages 1–8.

[Ruijters et al. 2008] Ruijters, D., ter Haar Romeny, B. M., and Suetens, P. (2008). Efficient gpu-based texture interpolation using uniform b-splines. *J. Graphics Tools*, 13(4):61–69.

[Ruijters and Thévenaz 2012] Ruijters, D. and Thévenaz, P. (2012). Gpu prefilter for accurate cubic b-spline interpolation. *Comput. J.*, 55(1):15–20.

[Rusinkiewicz and Levoy 2001] Rusinkiewicz, S. and Levoy, M. (2001). Efficient variants of the icp algorithm. In *3DIM*, pages 145–152.

[Rusu and Cousins 2011] Rusu, R. and Cousins, S. (2011). 3d is here: Point cloud library (pcl). In *ICRA*, pages 1 –4. IEEE Computer Society.

[Sandor et al. 2010] Sandor, C., Cunningham, A., Dey, A., and Mattila, V.-V. (2010). An augmented reality x-ray system based on visual saliency. In *ISMAR*, pages 27–36. IEEE Computer Society.

[Santin 2008] Santin, R. (2008). Sistema de autoria em ambiente colaborativo com realidade aumentada. Master's thesis, Universidade Metodista de Piracicaba.

[Schall et al. 2009] Schall, G., Mendez, E., Kruijff, E., Veas, E., Junghanns, S., Reitinger, B., and Schmalstieg, D. (2009). Handheld augmented reality for underground infrastructure visualization. *Personal Ubiquitous Comput.*, 13(4):281–291.

[Schulte zu Berge et al. 2014] Schulte zu Berge, C., Baust, M., Kapoor, A., and Navab, N. (2014). Predicate-based focus-and-context visualization for 3d ultrasound. *Visualization and Computer Graphics, IEEE Transactions on*, PP(99):1–1.

[Sigg and Hadwiger 2005] Sigg, C. and Hadwiger, M. (2005). Fast third-order texture filtering. In Pharr, M., editor, *GPU Gems 2*, pages 313–329. Addison-Wesley.

[Sikachev et al. 2010] Sikachev, P., Rautek, P., Bruckner, S., and Gröller, M. E. (2010). Dynamic focus+context for volume rendering. In *Proceedings of Vision, Modeling and Visualization 2010*, pages 331–338, University of Siegen, Siegen, Germany.

[Sloan 2009] Sloan, P.-P. (2009). Stupid spherical harmonics (sh) tricks. In *Game Developers Conference.*

[Souza et al. 2014] Souza, A. C., Macedo, M., and Apolinario, A. (2014). Multi-Frame Adaptive Non-Rigid Registration for Markerless Augmented Reality. In *VRCAI*, China.

[Souza et al. 2012] Souza, R. C., Moreira, H. D. F., and Kirner, C. (2012). Flaras 1.0–flash augmented reality authoring system.

[Stefan et al. 2014] Stefan, P., Wucherer, P., Oyamada, Y., Ma, M., Schoch, A., Kanegae, M., Shimizu, N., Kodera, T., Cahier, S., Weigl, M., Sugimoto, M., Fallavollita, P., Saito, H., and Navab, N. (2014). An ar edutainment system supporting bone anatomy learning. In *Virtual Reality (VR), 2014 iEEE*, pages 113–114.

[Suenaga et al. 2013] Suenaga, H., Hoang Tran, H., Liao, H., Masamune, K., Dohi, T., Hoshi, K., Mori, Y., and Takato, T. (2013). Real-time in situ three-dimensional integral videography and surgical navigation using augmented reality: a pilot study. *International Journal of Oral Science*, (2):98–102.

[Teichrieb et al. 2007] Teichrieb, V., Paulo, J., Lima, S. M., Apolinário, E. L., Souto, T., Cordeiro, M., Márcio, F., Bueno, A. S., Kelner, J., and Santos, I. H. F. (2007). A survey of online monocular markerless augmented reality. *International Journal of Modeling and Simulation for the Petroleum Industry*, 1(1):1–7.

[Tomasi and Manduchi 1998] Tomasi, C. and Manduchi, R. (1998). Bilateral filtering for gray and color images. In *ICCV*, pages 839 –846.

[Viola and Jones 2004] Viola, P. and Jones, M. J. (2004). Robust real-time face detection. *Int. J. Comput. Vision*, 57(2):137–154.

[Volume Library 2014] Volume Library (2014). Volume library. `http://www9.informatik.uni-erlangen.de/External/vollib/`. Accessed 22 September 2014.

[Whelan et al. 2012] Whelan, T., Kaess, M., Fallon, M., Johannsson, H., Leonard, J., and McDonald, J. (2012). Kintinuous: Spatially extended KinectFusion. In *RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras*, Sydney, Australia.

[Wieczorek et al. 2010] Wieczorek, M., Aichert, A., Kutter, O., Bichlmeier, C., Landes, J., Heining, S. M., Euler, E., and Navab, N. (2010). GPU-accelerated Rendering for Medical Augmented Reality in Minimally-Invasive Procedures. In *BVM 2010*. Springer.

[Wittenbrink et al. 1998] Wittenbrink, C., Malzbender, T., and Goss, M. (1998). Opacity-weighted color interpolation for volume sampling. In *Volume Visualization, 1998. IEEE Symposium on*, pages 135–142.

[Wu et al. 2013] Wu, S.-T., Yauri Vidalon, J., Souza Loos, W., and Coan, A. (2013). Query tools for interactive exploration of 3d neuroimages: Cropping, probe and lens. In *Graphics, Patterns and Images (SIBGRAPI), 2013 26th SIBGRAPI - Conference on*, pages 250–257.

[Young 1994] Young, T. (1994). *Handbook of Pattern Recognition and Image Processing: Computer Vision*. Academic Press.