DIGITALLY RECONSTRUCTED WALL RADIOGRAPHS

EE XIANHE
(B.COMP (COMP. ENG.) HONS, NUS)

A THESIS SUBMITTED FOR THE DEGREE OF
MASTER OF SCIENCE (COMPUTING)
SCHOOL OF COMPUTING
NATIONAL UNIVERSITY OF SINGAPORE
2008

Acknowledgements

I will like to thank my supervisor, Associate Professor Dr. Leow Wee Kheng, for his many suggestions and constant support during this research. I am also thankful to Dr Howe Tet Sen from Singapore General Hospital for his valuable medical knowledge and imaging data which make this research possible. Next, I will like to show my appreciation for the invaluable insights from Dr. Borys Shuter, medical researcher, at National University Hospital.

I have the pleasure of meeting Chen Ying and Piyush Kanti Bhunre in my research laboratory. They have provided much help in overcoming some of the difficulties I have encountered. I also have to thank Chan Siwei, an Honours Year student, who makes rendering in real time possible by implementing the volume rendering method with hardware acceleration.

I am very grateful to my parents who have been showing nothing but patience and support throughout this tough period.

Finally, I want to thank my boyfriend, Dennis, for being with me through all the good and bad times.

Ee Xianhe
01 June 2008

Table of Contents

Abstract

Computer-Aided Surgery (CAS) technology enables the use of computers to generate 3D virtual environments of body parts slated for operation. In these virtual environments, surgeons enjoy the benefits of visualization, pre-operative planning, simulation and real-time navigation. All these translate into better surgical treatments, reduced complications, improved patient safety and lower health-care costs.

CAS applications, like model-based segmentation, real-time navigation, simulation and modeling of orthopedic implants, require an essential technique for medical image analysis which is the registration of an anatomical model to medical images. Existing registration approaches include geometry-based and intensity-based approaches which have some shortcomings. Intensity-based approach generates a lot of irrelevant details which can obscure relevant features. Hence, it is susceptible to getting trapped at local minima. It is also computationally expensive. On the other hand, geometry-based approach needs to extract features from the images. Feature extraction algorithms generally cannot distinguish between relevant and irrelevant features. Thus, it is difficult to automate this approach with accuracy and reliability.

There are three critical components in accurate and robust model-based medical image registration: the model, the objective function and the optimization algorithm. This thesis shall focus on the modeling of 3D data for registration. It proposes a hybrid approach that can combine the strengths of the two registration approaches while mitigating their weaknesses. It uses a 3D wall model of an anatomical part such that the wall surfaces capture surface shape, the solid wall captures intensity information, and the interior is hollow. In a digitally reconstructed radiograph of the wall model (DRWR), high-contrast features similar to edges and contours are produced only by the wall. DRWR can be registered to an x-ray image using intensity-based approach without the need for feature extraction.

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

With the advent of sophisticated hardware and software, computer technologies are increasingly being used to aid medical practitioners in their tasks. Applications of computer technologies range from the storage of medical images in digital image formats for easy storage and manipulation, to the visualization of the various body parts of patients that require treatment. Of particular interest is the area of Computer-Aided Surgery (CAS).

CAS technology enables the use of computers to generate 3D virtual environments containing body parts slated for operation (Figure 1.1). In these virtual environments, surgeons can observe the problem and perform pre-operative planning. Furthermore, tools can be provided such that surgeons can run through a simulation of the operation before actually performing it. Therefore, CAS provides surgeons with more accurate diagnostic capabilities as well as visualization via the 3D virtual environment and simulation, thus allowing for greater precision during the procedure and hence higher chances of success. CAS also provides surgeons with real-time navigation tool which is both fast and accurate [BLH07, GKBT98, LYJ03, ZGW01, LCJ+00] (Figure 1.2). Furthermore, procedures involving CAS are often minimally invasive, meaning faster patient recovery. All these advantages translate into better surgical treatments, reduced complications, improved patient safety and lower health-care costs.

An important component of CAS technology is the registration of an anatomical model to

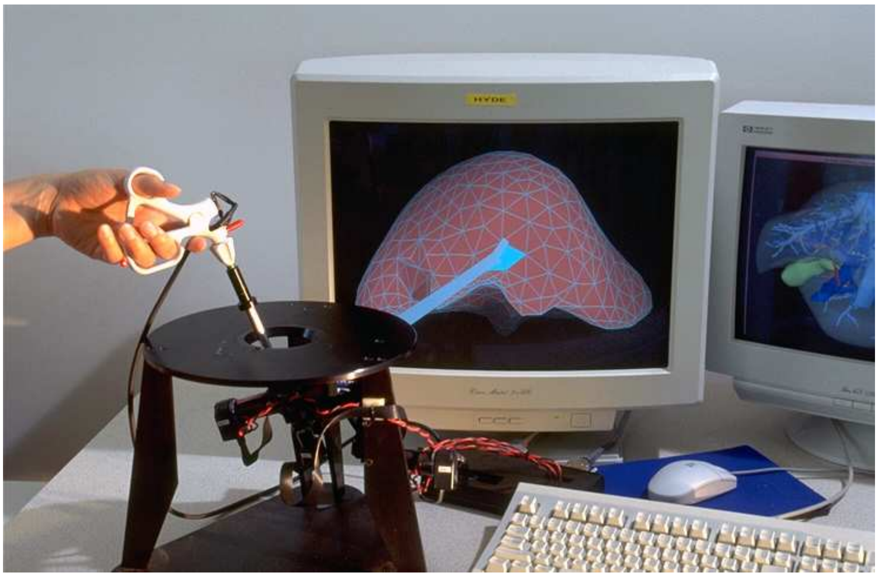
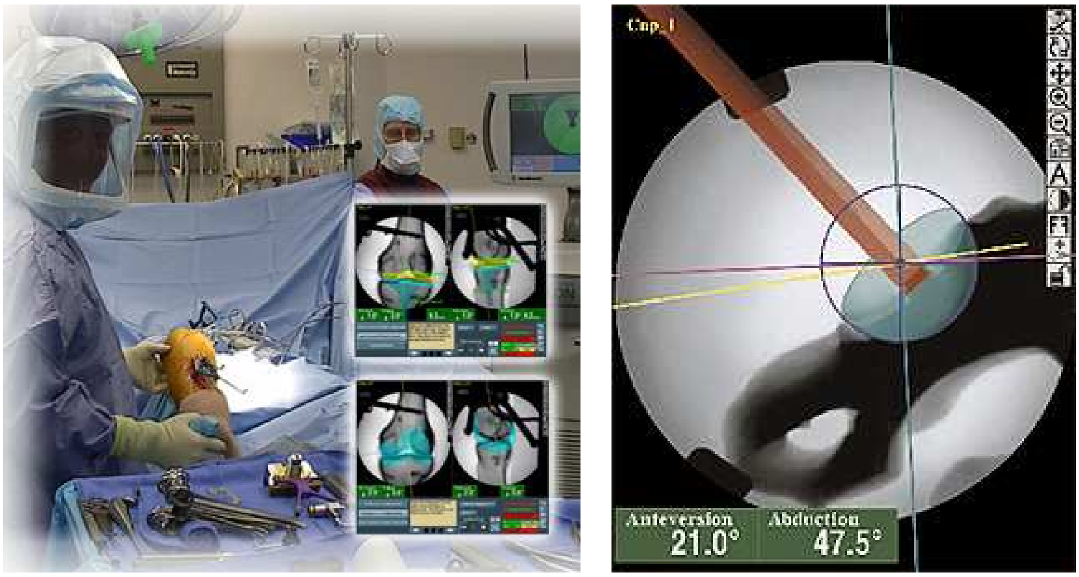


Figure 1.1: Surgical simulation in 3D virtual environment. (Adapted from Institute for information and computing sciences, Utrecht University [sim].)

medical images. Registration is the process of determining a mapping between a model in one coordinate space to an input object in another coordinate space for the purpose of aligning the model to the object. One example that needs registration is the hip replacement surgery (Figure 1.2(b)). Using flouroscopic images of the hip implant taken during the surgery, the hip implant can be registered to the hip joint to retrieve its relative orientation. This allows better visualization of the surgery, resulting in higher accuracy of the hip implant placement.

There are two approaches for solving registration problems: geometry-based and intensity-based. Both approaches have their strengths and weaknesses. Geometry-based approach relies on the presence and identification of natural landmarks or fiducial markers in the input datasets in order to determine the best alignment. Natural landmarks such as edges and contours can be used for comparison between 2D medical images and usually a 3D surface model [BLH07,GKBT98,LS95] (Figure 1.3). However, it is also possible to use a 3D volume model for registration [LS95]. The small number of features to be registered makes this approach computationally efficient. However, this approach needs to extract features from the images for matching with the model. In general, feature extraction algorithms cannot distinguish between relevant and irrelevant features. So, it is very difficult to automate



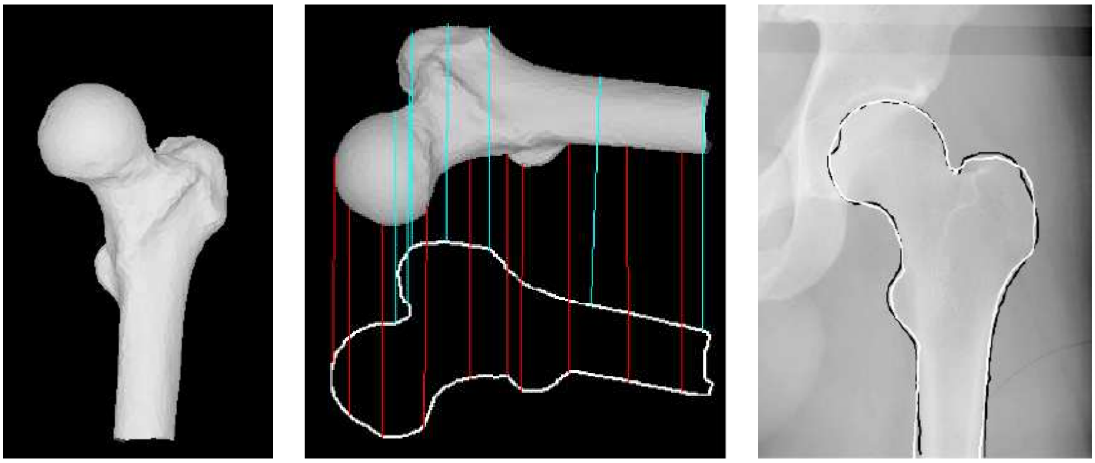
(a) (b)

Figure 1.2: More precise real-time targeting of anatomical sites for Computer-Aided Surgery. (a) Knee Arthroplasty. (b) Hip Replacement. (Adapted from Smith & Nephew Orthopaedics [CAS].)

geometry-based methods with accuracy and reliability in real applications.

On the other hand, intensity-based approach operates on the pixel or voxel intensities directly (Figure 1.4). It calculates various statistics using the raw intensity values of the inputs which are then compared to the images to be aligned. Often, 3D volume data such as 3D CT or MR volume, is used to synthesize a digitally reconstructed radiograph (DRR) [ZGW01, LCJ$^+$00, vdKPT$^+$05] and the pixel intensities of the DRR (Figure 1.5) are matched with those in the images. Although this approach is computationally expensive due to the greater number of points to be registered than in the case of the geometric-based approach, no feature extraction step is required. It also can generate more details, both relevant and irrelevant, but some relevant features may be obscured by the abundant details. Hence, it is susceptible to getting trapped at local minimal solutions because of the large amount of irrelevant features in the images.

A hybrid approach has been proposed [LYJ03, TLP02] that matches the intensity gradient images of CT volume and x-ray image. In this way, high-contrast features that correspond to edges are matched in an intensity-based approach without performing explicit feature



(a) (b) (c)

Figure 1.3: Geometry-based registration using contour. (a) 3D surface model. (b) 2D contour generated from (a). (c) Result of registration between (b) (white) and the image contour (black). (Adapted from [BLH07])

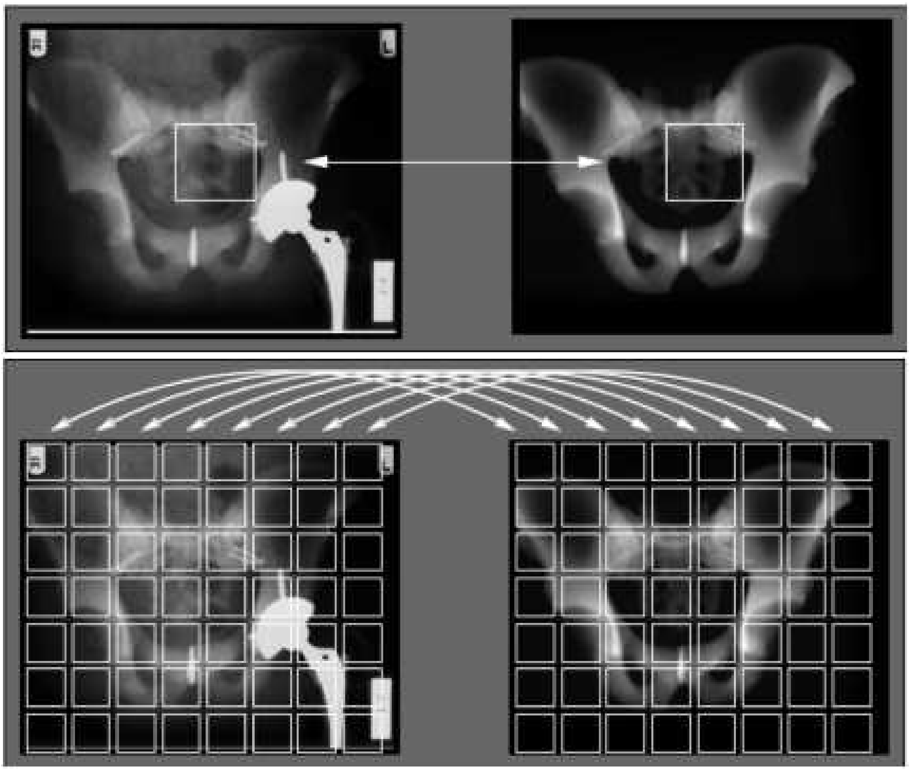


Figure 1.4: Intensity-based registration.(Adapted from [LCJ^{+}00])

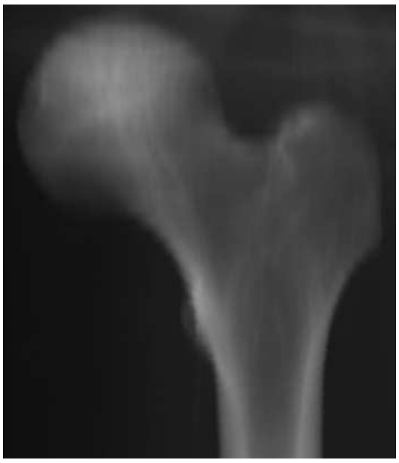


Figure 1.5: DRR generated from a 3D volume model of femur.

extraction. Actually, this approach is very close to the geometry-based approach that extracts edges because edges are just thresholded intensity gradients. So, it can still be affected by irrelevant or extraneous gradients.

There are three critical components in the registration of a model to an image, namely the model, the objective function and the optimization algorithm. The objective function and the optimization algorithm are usually studied together and is, in general, a difficult and challenging topic in medical image analysis. Hence, this thesis focuses only on the modeling of 3D data in registration. This thesis proposes a hybrid approach that can combine the strengths of the geometry- and intensity-based approaches while mitigating their weaknesses. It uses a 3D *wall model* of an anatomical part: the inner and outer wall surfaces capture surface shape, the interior between the wall surfaces captures intensity information, and the interior within the inner wall surface is hollow. When a digitally reconstructed wall radiograph (DRWR) of the wall model is synthesized, high-contrast features similar to edges and contours are produced only by the wall. Therefore, it produces clearer features and less noise than conventional DRR (Figure 1.5) and intensity gradient maps. DRWR can be registered to an x-ray image using intensity-based method without feature extraction.

In principle, the approach of DRWR can be applied to the registration of any anatomical part with thick walls such as large bones and the heart. In this thesis, the approach shall be illustrated for long bones, specifically the femur.

1.2 Research Objectives

The main objective of this thesis is to develop an algorithm for constructing a hybrid 3D volume model, known as 3D wall model, given the 3D CT volume of the anatomical part of interest. It combines the strengths of both volumetric and surface models while alleviating their weaknesses. The 3D wall model will capture both the surface shape features and relevant 3D intensity data of the wall. The interior of the inner wall surface is hollow to remove irrelevant volume data.

The DRWR of the 3D wall model is synthesized by simulating the attenuation of x-rays as they pass through the 3D wall model. It naturally contains high contrast features, corresponding to the wall, which are similar to edges and contours without irrelevant features.

1.3 Organization of Thesis

In order to provide a better understanding of the problem discussed earlier, medical knowledge about bone anatomy and characteristics of x-ray imaging process will be explained in Chapter 2. There are three parts to the application of DRWR namely, 3D model reconstruction, synthesis of DRWR and 3D-2D registration of rigid objects. Hence, related work done in the areas of DRWR synthesis and registration are discussed in Chapter 3. Based on the previous chapters, the algorithm for constructing the 3D wall model is presented in Chapter 4, followed by the algorithm for DRWR synthesis in Chapter 5. To prove the viability of DRWR and to evaluate its performance against the DRR in terms of robustness, tests determining the degree of match between each model and the x-ray image are carried out. These results are then discussed in Chapter 6. Lastly, the thesis is concluded with a summary and possible future work in Chapter 7.

Chapter 2

Background Knowledge

2.1 Bone Anatomy

A long bone such as the femur has a very unique shape. Its shaft is cylindrical but its extremities differ in shape when viewed from different viewing angles. The wall of the bone, categorized as *cortical bone*, is a dense deposit of calcium phosphate, which shows up as bright regions in an x-ray image (Fig. 2.1(a)). It is thickest along the shaft and decreases smoothly towards the extremities. The bone's interior contains a honey-comb structure called *trabeculae*, which shows up as a textured pattern of densely packed fine lines (Fig. 2.2). However, not all of the lines correspond to trabeculae. Some of them are due to the complex surface shape of the bone extremities (Fig. 2.1(b, c)).

The distribution and density of the trabeculae differ significantly among patients, especially for those suffering from osteoporosis. Moreover, the bone is a living tissue which undergoes continuous degradation and regeneration. Delicate structures like trabeculae are therefore more prone to changes induced by this process. This means that the trabecular network can even change over time for the same individual. So, the trabeculae are not useful for registration.

On the other hand, cortical bone at the shaft and the extremities produce high-contrast features, such as edges and contours (Fig. 2.1(b, c)), that are quite consistent among patients if these parts are not fractured or diseased. So, they can potentially provide more information for accurate registration.

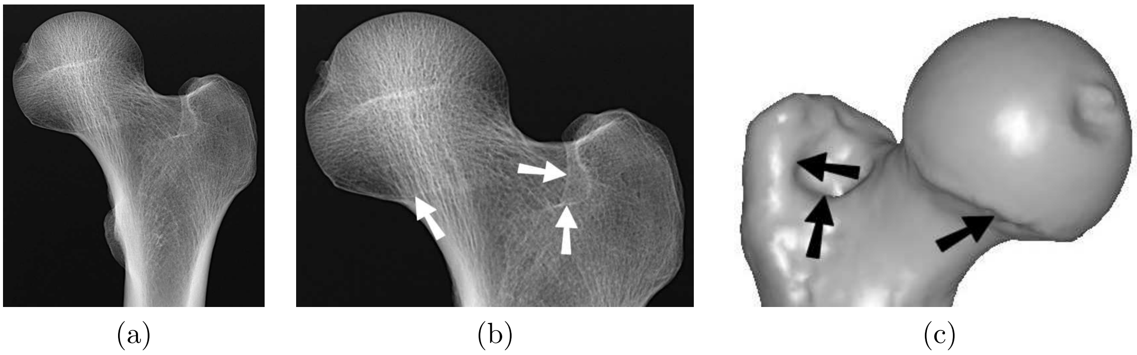


Figure 2.1: Structure of femur. (a) The shaft is cylindrical and its wall (solid white) decreases in thickness smoothly towards the upper extremity. (b) The distribution of the trabeculae (dense fine lines) differ among patients but the edges (arrows), due to the surface structure, are consistent among patients. (a, b) X-ray images. (c) 3D mesh model.

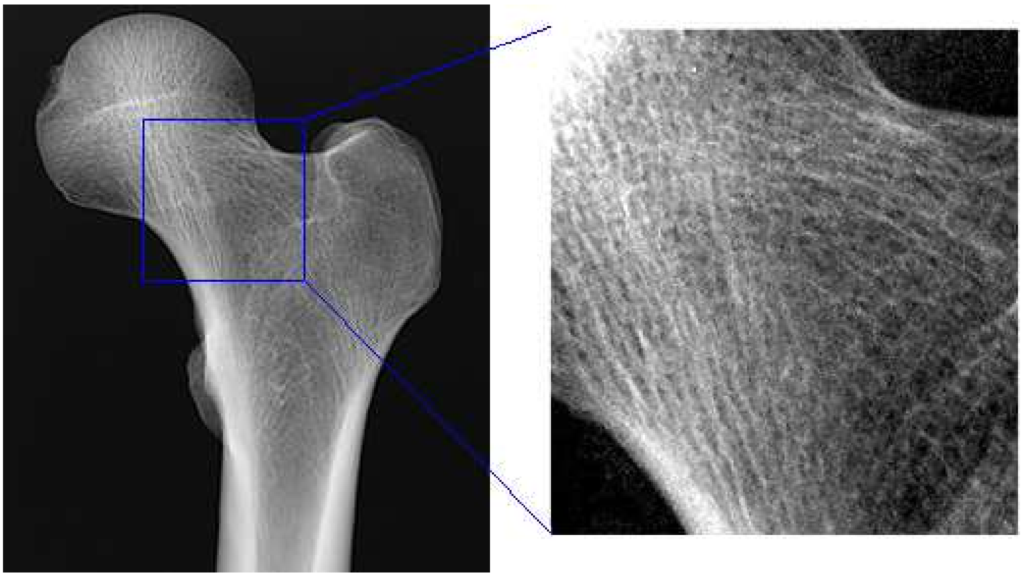
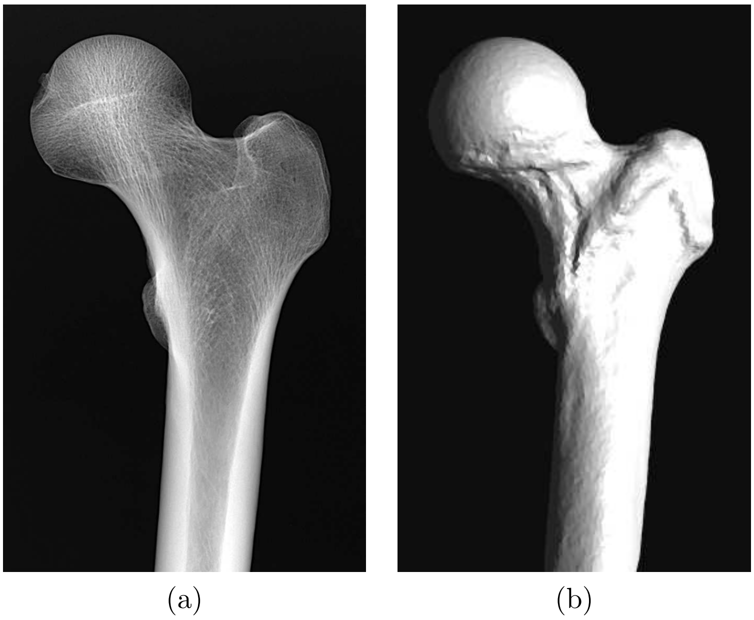


Figure 2.2: Trabecular network in a femur.



(a) (b)

Figure 2.3: Comparison between (a) an x-ray image and (b) an optical image of the same bone.

2.2 Characteristics of X-ray Imaging

As a beam of x-ray passes through matter, x-ray photons are removed from the x-ray beam by absorption and scattering, resulting in the *attenuation* of the incident x-ray. X-ray imaging captures the attenuation of the x-ray as it passes through the object whereas normal optical imaging captures light reflection when the ray hits the surface of the object and is reflected according to the surface property. Therefore, the images formed by these two modalities are totally different (Figure 2.3).

The diagnostic x-ray imaging process can be modeled as a linear attenuation of x-rays as they pass through the patient's body [LaR01]. Under the linear attenuation model, each type of tissue has an associated *linear attenuation coefficient* μ that describes the likelihood of a photon being attenuated as it passes through the tissue. If the tissue has uniform linear attenuation coefficient μ, the attenuation through the slab can be described as

$$N_{out} = N_{in} e^{-\mu x} \tag{2.2.1}$$

where a number of photons N_{in} enter a slab of tissue of uniform thickness x. Only some unattenuated photons N_{out} will exit the tissue and reach the x-ray sensor (Figure 2.4). This

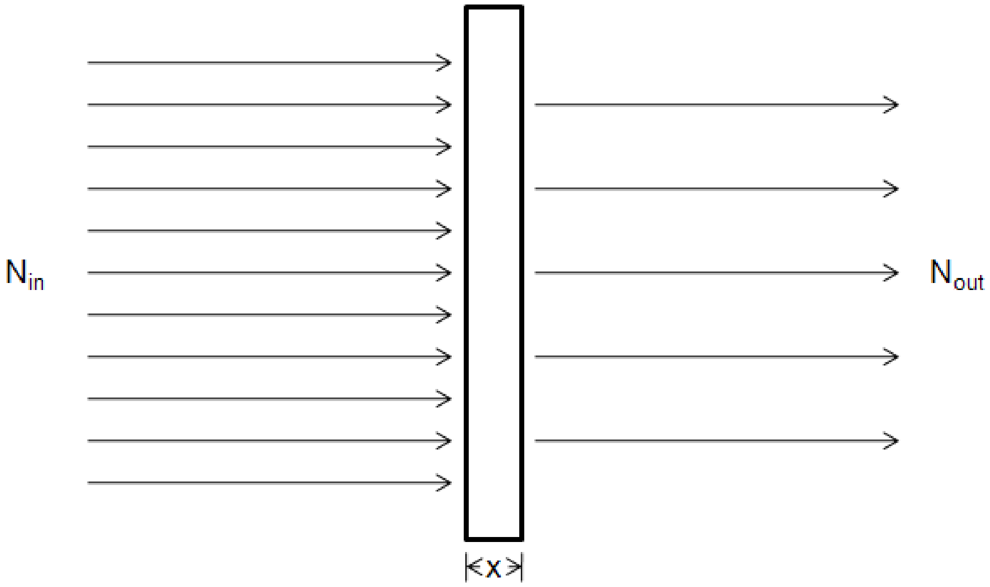


Figure 2.4: Attenuation of x-ray photons through a slab of tissue.

is also known as the Lambert-Beer's Law.

Human tissues are not of uniform thickness and an x-ray can pass through different types of tissues with different attenuation coefficients. Therefore the quantity μx in Equation 2.2.1 should be replaced by a line integral:

$$N_{out} = N_{in} e^{-\int \mu(x) dx} \tag{2.2.2}$$

where $\mu(x)$ is the attenuation coefficient at position x along the x-ray path. Taking logarithm of Equation 2.2.2 yields the *log total attenuation*:

$$\log N_{out} = \log N_{in} - \int \mu(x) dx \tag{2.2.3}$$

For more details on the derivation of the formulae, please refer to the medical handbook [BKM00].

2.3 Tomographic Imaging

Tomographic imaging is an enhancement of x-ray imaging to produce cross-sections of interest. Computed Tomography (CT) is a technique in which transmission measurements

Figure 2.5: CT scanning. (Adapted from Pauls Valley General Hospital [CTs].)

of a narrow beam of x-rays, made at constant intervals of angles around an object, are used to synthesize slices of interest within the object (Figure 2.5). The synthesis of the slices of interest is done by back-projection whereby every point on the series of x-ray images is projected back along the x-ray path to form an image. As a result, simple back-projection renders a crude central image with blurred radial spokes, known as the *star artefact* (Figure 2.6). This greatly degrades the spatial information in the CT image. However, this artefact can be solved by simply filtering each projection in the frequency domain, followed by inverse Fourier transform before the original back-projection process.

A CT volume is acquired as a sequence of CT slices in the axial (z) direction. The CT volume is rarely taken in isotropic resolution because of the health-damaging radiation that patients are exposed to. Therefore, the CT slice thickness is usually of lower resolution compared to the x and y resolution.

There are some common artefacts associated with CT imaging such as Partial Volume Effect (PVE) and cupping artefact due to beam hardening. PVE arises when a voxel contains more than one tissue type which results in averaging of the densities (Figure 2.7). PVE often causes blurring outside the surface boundary of the object which can distort the spatial information of the surface boundary. However, partial volume effect is

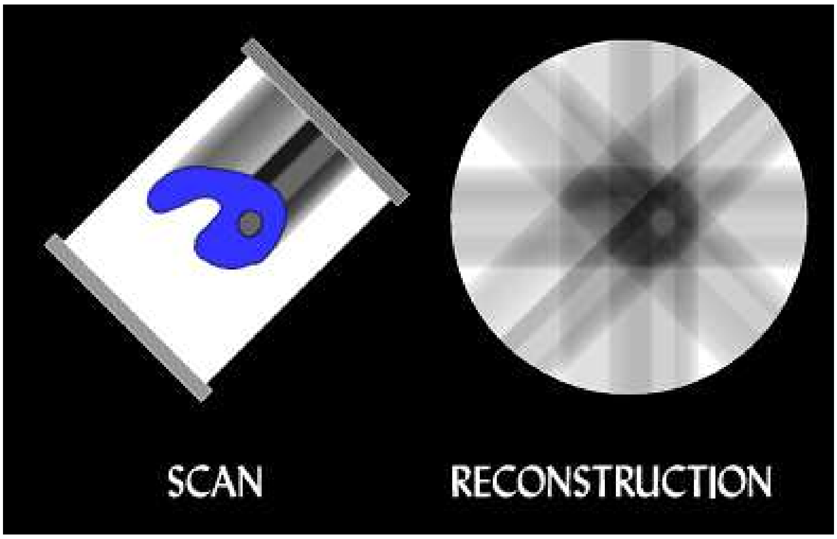


Figure 2.6: Re-synthesis of CT by simple back-projection with star artefact. (Adapted from the MRC Human Genetics Unit Edinburgh UK [bac].)

considered insignificant most of the time and can also be reduced by increasing the axial resolution. There is also a phenomenon known as *beam hardening* in x-ray imaging which can cause artefact in CT. In diagnostic x-ray imaging, the x-ray beam has a moderately broad energy spectrum. X-rays in energy ranges that are more easily attenuated are known as *soft x-rays* whereas those that are more penetrating are known as *hard x-rays.* Beam hardening is the process of the progressive removal of the soft x-rays as the x-ray beam passes through body tissues. The amount of beam hardening depends on the initial x-ray spectrum and the composition and thickness of the tissue traversed. Beam hardening can result in *cupping artefact* in CT (Figure 2.8) where the reconstructed attenuation coefficients decrease towards the centre of a large uniform object. This cupping artefact should not affect the construction of the 3D wall model because it contains only the wall of the femur. Therefore, the effects of these artefacts on 3D wall modeling are negligible as explained.

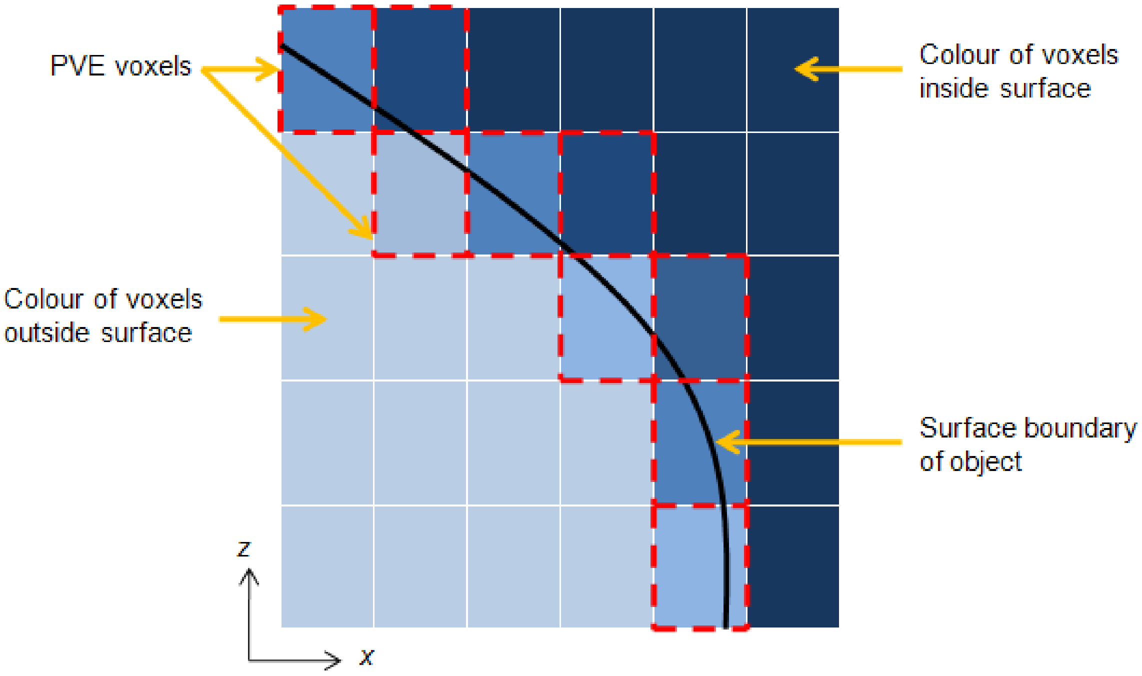

PVE voxels
Colour of voxels inside surface
Colour of voxels outside surface
Surface boundary of object
z
x

Figure 2.7: Partial volume effect in CT.

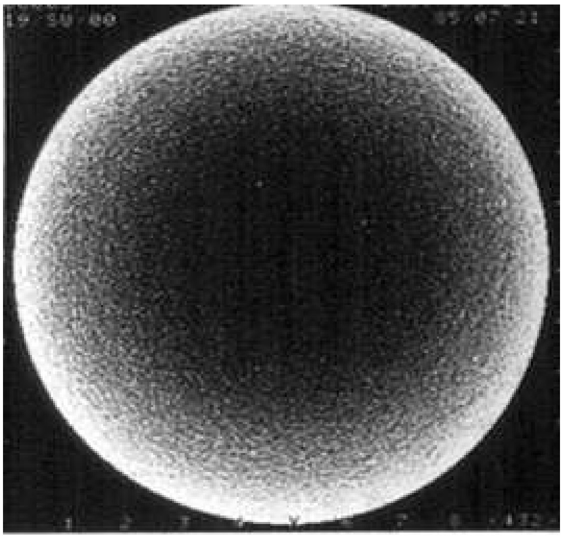


Figure 2.8: Cupping artefact in CT. (Adapted from the ImPACT group [cup].)

Chapter 3

Related Work

The problem of generating a DRWR comprises three main steps. A 3D wall model has to be first reconstructed from 3D medical images such as Computed Tomography (CT), Magnetic Resonance Imaging (MRI), 3D ultrasound, Single-Photon Emission Computed Tomography (SPECT) and other cross-sectional images. After a 3D wall model is reconstructed, the next step will be the synthesis of the DRWR. Existing methods of DRR synthesis related to synthesis of DRWR will be brought forth and discussed in Section 3.1. Lastly, registration of the projection of DRWR to input images is required for image analysis. Algorithms for 3D-2D registration of rigid objects will therefore be reviewed in Section 3.2.

3.1 DRR Synthesis

Many volume rendering techniques have been applied for the synthesis of DRR. They can be categorized into ray casting, splatting, shear-warping, Fourier volume rendering and Monte Carlo volume rendering. Ray casting is an image order algorithm which allows direct simulation of x-ray attenuation through an object while splatting is an object order technique. Both methods produce comparable results. Shear warping simplifies the projection from a viewpoint to a shear followed by a warp, which is faster than the previous two methods but image quality is compromised. Fourier volume rendering provide only direct integration along the line of projection to produce x-ray-like images at a faster speed. Monte Carlo is recommended for processing large volume data at a faster speed by sampling. In the following sections, these techniques together with their variants for optimization and improvement will be further discussed.

3.1.1 Ray Casting

Ray casting [Lev88] is a straight-forward, image-order algorithm whereby a ray is shot from the eye through each pixel of the image into the volume. Along the ray's intersection with the volume several operations can be performed to obtain the color of the pixel. Therefore, ray casting allows for the direct simulation of x-ray attenuation as described in Section 2.2. However, perspective ray casting with trilinear interpolation is a rather time consuming method of volume rendering. Moreover, the stepping and summing of the integrals may lead to inaccuracies.

Performance of ray casting algorithms can be significantly improved without affecting image quality if regions which do not contribute to the image are skipped from rendering. However, the nature of ray casting needs a traversal of the data structure once every ray which results in many redundant computations. Many methods have been proposed by exploiting coherence in the data set. These methods rely on spatial data structures that encode the presence or absence of high-opacity voxels so that computations can be omitted in transparent regions of the volume. These data structures are built during preprocessing after applying an opacity transfer function to the volume. Such spatial structures include pyramids [DH92,Lev90], k-d trees [SF90] and distance transforms [SK00,ZaAHKV92]. Although this optimization is data dependent, researchers have reported that in typical classified volumes around $70 - 90\%$ of the voxels are transparent [Lev90, SF90]. Early ray termination is also another efficient optimization [Lev90]. By keeping track of the opacity of the data encountered so far, the ray can be stopped as soon as the cumulative opacity is close to total.

Another class of optimization methods for ray casting pre-compute projection values, which can then be accessed and interpolated during the registration phase. Similar data structures have been proposed to store such projection values, like the Transgraph by Larose [LaR01], the Direction Dependent Projection Fields by Freund et al. [FHR04] and the Attenuation Field (AF) by Russakoff et al. [RRM$^+$05]. These data structures have been extended from the Light Field [LH] and the Lumigraph [GGSC] in computer graphics. A variant of AF is Progressive Attenuation Field (PAF) by Rohlfing et al. [RRD$^+$04] which computes the AF on the fly. These methods trade-off image quality for speed improvement. Except for PAF, the other methods all require a one-time pre-computation for an exchange of interactive rendering rates, and assumptions have to be made regarding expected viewing angles for memory efficiency.

Attenuation Field (AF)

Light fields are proposed by Levoy and Hanrahan [LH]. A similar idea called the Lumigraph was simultaneously and independently presented by Gortler et al. [GGSC]. Essentially, light fields provide a method of parameterizing all the rays that emanate from a static scene. Each ray is represented by its intersection with two arbitrary planes in space. These two planes form a light slab where all light rays enter and exit. If all the rays within this light slab can be calculated, almost any image with a focal point inside the light slab can be recreated simply by determining the rays involved and associating them with their corresponding pixel values. Practically, a large number of them can be generated provided that a reasonably dense coverage of the space is known. The missing values are then generated by interpolating quadrilinearly among existing discrete samples. AF implements a virtual image plane in between the volume and focal plane for the same two-plane parameterization in light field. However, the rays will extend beyond the volume to the image plane behind the volume so that the attenuation along each ray can be calculated for each pixel.

Progressive Attenuation Field (PAF)

Unlike traditional attenuation fields, the PAF [RRD$^+$04] is built on the fly as the registration proceeds. It does not require any pre-computation, nor does it make any prior assumptions of the patient's pose that would limit the permissible range of patient motion. A cylindrical attenuation field parameterization is used which is better suited for rotating imaging systems than the usual two-plane parameterization. Compared to the common two-plane parameterization, the cylindrical coordinate space is capable of describing full 360^o rotations around the cylinder axis. This cylindrical coordinate space is characterized by the 4-tuple of real numbers (u, v, s, t). Attenuation values of rays are calculated at discrete intervals of these parameters and stored in a hash table for time-efficient storage and access. All other rays are then calculated using quadrilinear interpolation of the computed rays. However, these benefits come at the cost of higher computational cost during the actual registration.

Transgraph

The Transgraph [LaR01] is very similar to the AF in concept. It has two parallel coordinate planes C_0 and C_1, similar to the focal and image planes in AF, which lie across the CT volume. The two planes are only offset by a small amount. Therefore, any trajectory through the CT volume can be parameterized by q_0 and q_1, the intersections with these coordinate planes. The Transgraph is implemented as a nested data structure, a 2D array of 2D arrays. The top-level 2D array indices correspond to C_0 coordinates. Each element in the top-level array is a 2D array and the indices in the sub-array correspond to the C_1 coordinates. Each element in a sub-array stores the total attenuation along the corresponding ray. The sub-array describes only a region of the C_1 coordinate plane because it is calculated based on an expected range of the patient poses. During DRR synthesis, quadrilinear interpolation is used.

Direction Dependent Projection Field

Direction Dependent Projection Field by Freund et al. [FHR04] uses a different parameterization from the others. It stores pre-computed values of some rays in a four-dimensional array called the projection field. Single pixel values are also generated by quadrilinear interpolation between the values of the projection field. A projection field is a 2D array that stores in each element another 2D array. It assigns each quadruple (x, y, φ, θ) the intensity integral of the ray passing through the pixel at the location (x, y) with the direction (φ, θ). The directional parameters φ and θ are simply the angles between the projections of the ray onto two orthogonal planes normal to the projection field plane. The projection field is selected to be parallel to the DRR plane.

Ray Casting in Wavelet Space

Progressive refinement is made possible by Lippert and Gross [LG95], and Gross et al. [GLDH97], using ray casting in wavelet space, allowing adaptive trade-off between image quality and speed. The initial RGB volume data set is decomposed by 3D wavelet transform separately for each channel as a one-time process. The line integral along a ray can be approximated by wavelet expansion as an accumulation of the line integrals of 3D wavelets intersected by the ray. The intensity integral in wavelet space for each viewpoint is computed by slicing the 3D wavelet, perpendicular to the ray, in the frequency domain according to the Fourier Projection-slicing Theorem (Section 3.1.4). Due to the self-similarity of the 3D wavelets, the rendering process becomes an accumulation of scaled and shifted versions of eight 2D intensity or color distributions, corresponding to the 3D wavelets of the $RGB\alpha$ channels and their scaling functions. This results in eight different 2D textures, one for each mother wavelet representing their ray intensity integrals in frequency space. Next, the inverse FFT computes the intensity distribution in spatial domain. Superimposing these distributions produces an image. The image is progressively refined by an accumulation of more selfsimilar textures based on the basis functions.

3.1.2 Splatting

Splatting [Wes90], or footprint rendering, is an object-order algorithm which is the opposite of ray casting. Object-order algorithms incrementally reconstruct the original signal by spreading each data sample's energy into space. As each voxel is projected onto the image plane, the voxel's energy is spread over the image raster using a reconstruction kernel centered at the voxels projection point. This reconstruction kernel is called a *splat*. Conceptually, the splat is a spherically symmetric 3D reconstruction kernel centered at a voxel. However, since the splat is reconstructed into a 2D image raster, it can be implemented as a 2D reconstruction kernel. The 2D kernel then becomes a circle if the volume has equal spacing in each grid directions and the viewing transform is orthographic. Otherwise, the 2D kernel becomes an ellipse. This 2D kernel, called a *footprint function*, contains the integration of the 3D kernel along one dimension. The integration is usually pre-computed and the footprint function is represented as a finely-sampled 2D lookup table. The 2D table is centered at the projection point and sampled by the pixels which lie within its extent. Each pixel composites the value it already contains with the new value from the footprint table. If the volume grid has regular spacing, the viewing transform is orthographic and the splat is symmetric, the footprint table can be computed once and used for all voxels. Otherwise, the footprint function will vary. The table will need to be re-computed for each view if the kernel is non-symmetric, and possibly for each voxel for perspective projection.

Splatting produces images of quality comparable to ray casting but at greater speed. This is achieved by pre-computing the footprint table. Object-order algorithms are also more easily made parallel. This is because each voxel only needs to know about a small neighborhood and, shading and transforming can be done in parallel for sub-sections of the volume. However, it is hard to implement early ray termination as in ray casting. Integration of the volume is only approximate in splatting because kernels must overlap in object space to ensure a smooth image.

Hierarchical Splatting

Hierarchical splatting is a progressive refinement algorithm proposed by Laur and Hanrahan [LH91], which stores an octree within the pyramidal volume. Data coherence has also been used to build an octree to skip transparent regions as in ray casting. The pyramid is a multi-resolutional representation of the volume allowing for volume rendering at different resolutions at different speedups. The algorithm builds a set of footprints at different sizes, one for each level in the pyramid. It does not just draw a reduced resolution version of the volume, but determines the number of the splats by fitting a collection of cells at different resolutions in the pyramid to the original data based on a user-supplied error criteria. A low error in a large region indicates that it is homogeneous, and homogeneous regions can be drawn efficiently using large splats. This makes real-time rendering possible with trade-off in image quality.

Ray-driven Splatting

Mueller and Yagel [MY96] merge splatting with ray casting to yield a ray-driven splatting approach to speed up perspective splatting, which complicates the mapping of the footprints. Splats are imagined as being suspended in object space, a splat at every voxel. Rays then traverse the space and intersect the splats. Hence, each pixel accumulates its color, opacity, and density sums separately. Proper compositing is ensured by dividing the volume into 2D slices parallel to the image plane. When a ray is shot, it stops at each slice and determines the range of voxel kernels within the slice that are traversed by the ray. The voxel contributions are then accessed from the respective footprint tables for composition at the pixel. Ray-driven splatting is more efficient than perspective splatting as there is no requirement to compute the expensive perspective viewing transform. Furthermore, it offers acceleration methods such as early ray termination and bounding volumes, which traditional splatting cannot benefit from. This results in better speed performance. Perspective ray casting also under-samples volume regions farther away from the eye-point due to the diverging nature of the rays. This problem is eliminated by utilizing summed area footprint tables [Cro84] and tracing the volume by pyramidal ray beams.

Wavelet Splatting

Lippert and Gross [LG95], and Gross et al. [GLDH97] both modify the splatting algorithm by using wavelets as reconstruction filters, so that data can be visualized at different levels of detail. This scheme takes advantage of the self-similarity of the wavelet basis functions and computes the required splats by Fourier projection slicing. Wavelet approximation of the ray integration is a weighted accumulation of eight basis functions, a pair of scaling and wavelet functions for each of the $RGB\alpha$ textures. The self-similarity property of wavelets enables all other textures to be derived from these eight basis functions by scaling and translation. Volume rendering is then an accumulation of scaled and translated versions of the RGB textures i.e. the splats. The computation of the basis functions can be accomplished by Fourier projection slicing. More details about Fourier projection slicing can be found in Section 3.1.4. 3D FT of the 3D wavelet is first performed, followed by a 2D inverse FT of the slice to obtain the wavelet-splat footprint. The wavelet splatting method progressively refines the image when more and more footprints are used to build up the representation, allowing real-time rendering.

3.1.3 Shear-warping

Shear-warp factorization, by Lacroute and Levoy [LL94], combines the advantages of image-order and object-order algorithms. This method is based on three steps namely, factorization of the viewing matrix into a 3D shear parallel to the slices of the volume data, a projection to form a distorted intermediate image and a 2D warp to produce the final image. Shear-warping is faster than ray casting and splatting. However, as two re-sampling steps are required for the intermediate projection and during the warping step, images produced using this technique are more blurred as compared to the results of ray casting or splatting.

For a perspective transformation, each slice must be scaled as well as sheared. The scanlines of the voxels in the volume data are always aligned with that of the pixels in the intermediate image. Therefore, spatial data structures based on run-length encoding are used to skip runs of transparent voxels and opaque pixels. Each opaque (or occluded) intermediate image pixel stores an offset to the next non-opaque pixel in the same scan-line so as to skip runs of opaque pixels. Hence, work is only done for voxels which are both non-transparent and visible. After the intermediate image is rendered, a general-purpose affine image warper with a bilinear filter is applied to produce the final image in parallel projection. The paper also implements a data structure for encoding spatial coherence in unclassified volumes using octree and summed area table.

Splatting Shear-warp

The splatting shear-warp algorithm was originally used for Maximum Intensity Projections (MIP) [CS98]. Cai and Sakas [CS00] adapted this algorithm for use in DRR rendering. Splatting in shear-warp space uses 3D reconstruction filter and area sampling to produce better quality images than standard projective shear-warp which uses only 2D reconstruction filters and point sampling. A cubic voxel consisting of points (i, j), $(i+1, j)$, $(i+1, j+1)$ and $(i, j + 1)$ is sheared to a parallelepiped. The parallelepiped voxel is then splatted to pixel A along the principal viewing direction. The contribution of the parallelepiped voxel to pixel A is integrated by its projection area i.e. area sampling. The footprint of the parallelepiped voxel is called the *sheared footprint*. All the voxels in the sheared object space share the same general 2×2 sheared footprint.

3.1.4 Fourier Volume Rendering (FVR)

FVR, by Dunne et al. [DNR90], is based on the *Fourier Projection-slice Theorem* which states that a 2D projection of a function $f(x, y, z)$ at an arbitrary angle may be computed by taking the inverse 2D Fourier transform (FT) of a slice, of the 3D FT of f, that passes through the origin and is normal to the direction of projection. FVR allows projections of volume data to be generated in $O(n^2 \log n)$ time for a volume of size n^3 instead of $O(n^3)$ in ray casting. A one-time preprocessing step of applying a 3D FT to the volume data is first performed to obtain a 3D array of samples in the spatial-frequency space. A plane slice through the centre of the new array is re-sampled and 2D inverse Fourier transformed to produce an image. This image is a re-projection of the original 3D data in a direction normal to the extracted plane. Therefore, a re-projection at any desired angle can be computed quickly by simply extracting the appropriate slice and applying a

2D FT. FVR can generate views at interactive rates. Progressive refinement is also easily performed by limiting the sampling rate at low frequency, resulting in a smaller image that can be interpolated up to arbitrary size. This reduces the complexity of both the resampling and inverse transformation stages. However, FVR is a linear projection which is order independent along the line of projection t. Therefore, hidden surface effects are not present and limits FVR to transparent imagery which look like X-rays. Due to the complex arithmetic associated with FT, the memory cost of FVR is much higher than spatial domain algorithms.

Malzbender [Mal93] also propose a similar approach as Dunne et al., with the addition of a filter design and spatial data preprocessing techniques to produce artefact-free renderings. He also makes sure that the resampling rate used for the inverse FT is high enough, so that aliasing in the spatial domain in the form of overlapping of copies of the original data set is avoided. Resampling filter which is band-limited in both the frequency and spatial domains is employed to reduce aliasing. This is made possible by using the Projection on Convex Set technique [CN88], which allows constraints in both the frequency and spatial domains to be optimized or satisfied, if possible. The filter is iteratively transformed back and forth the frequency and spatial domains with truncation until the change between iterations is below a threshold. In addition, common filters like the triangle and sinc functions lead to an overemphasis of the central region of the data set due to their shapes. Malzbender compensates this effect by spatial pre-multiplication, which multiplies the original data set by $1/g(x, y, z)$ where $g(x, y, z)$ is the filter function in spatial domain, before the 3D FT is taken. Spatial zero-padding to the pre-multiplied data will reduce the distortion, further reducing the aliasing problem.

Ntasis et al. [NCSN99] also implement FVR with a hybrid filter and spatial zero-padding for real-time DRR rendering. The hybrid filter uses Hamming windowed sinc() function for the central area of the spectrum and trilinear interpolation for the rest to improve the quality of the result.

Fourier Wavelet Volume Rendering (FWVR)

Westenberg and Roerdink [WR00] make use of wavelets to progressively refine the FVR to allow for real-time interaction. This technique is similar to the approach by Lippert and Gross [LG95] which uses wavelets to progressively refine the ray casting and splatting

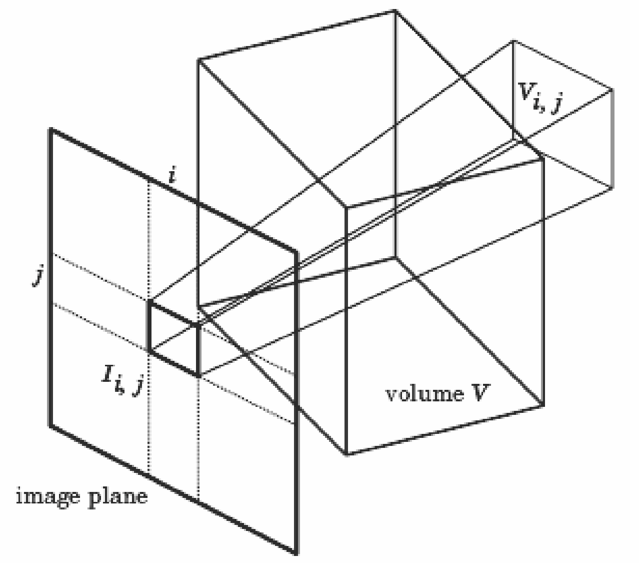


Figure 3.1: Calculation of a pixel intensity $I_{i,j}$ as a volumetric integral over the corresponding pyramidal domain $V_{i,j}$ [CSK03].

methods. However, the time complexity of FWVR is $O(n^2 \log n)$ which is faster than the time complexity of $O(n^3)$ for wavelet splatting and ray casting in wavelet space. The 3D FT of the volume data is first computed. For each direction θ, resampling of the FT is done in the slice plane normal to θ. The 2D Fourier-wavelet decomposition (FWD) of depth M is performed on the resampled slice to obtain approximation coefficients and detailed coefficients D_j respectively. A partial Fourier-wavelet reconstruction (FWR) from the approximation coefficients without the detailed signals, followed by a 2D inverse FT yields an initial approximation in the spatial domain. The image can be progressively refined by partial FWR using D_j, $0 \leqslant j \leqslant M$, followed by a 2D inverse FT.

3.1.5 Monte Carlo Volume Rendering (MCVR)

Csébfalvi and Szirmay-Kalos [CSK03] present a novel volume-rendering technique based on Monte Carlo integration. In MCVR, a pixel intensity $I_{i,j}$ is calculated as a volumetric integral of $f(x)$ over a pyramidal domain $V_{i,j}$ which is projected onto the given pixel (Figure 3.1). The calculation of such a volumetric integral can be interpreted as an evaluation of infinitely many rays piercing through the rectangular pixel area. The volume data is first convolved with a reconstruction kernel yielding a continuous 3D density function. A point cloud of random samples is then generated using this probability density function. This point cloud is projected onto the image plane, and the normalized intensity of each pixel is estimated as the number of samples projected onto the given pixel divided by the number of all the samples in the point cloud. A DRR is then rendered by quantizing the estimated normalized intensities onto L gray levels provided by the display device.

Theoretically, for a fixed image resolution, there exists M number of samples such that the average standard deviation of the estimated pixel intensities is under the level of quantization error regardless of the number of voxels. Therefore, MCVR is mainly proposed to efficiently visualize large volume data sets. Furthermore, the trade-off between image quality and speed can be controlled by progressive refinement where a subset of the point samples can be interactively rotated, and after having the appropriate viewing direction fixed, the rest of the samples are projected onto the image plane, progressively refining the estimation. The time complexity of MCVR is $O(n^2)$. The accuracy of MCVR depends on the sampling rate for point cloud generation.

3.2 3D-2D Registration of Rigid Objects

For a 3D-2D registration problem, there is a need to optimize an objective function that measures the matching error between the 3D model and the 2D image. Some well-known optimization techniques are the Iterative Closest Point (ICP) [BM92a], gradient descent, Newton method, least square method and Levenberg-Marquad method [CZ96]. The correspondence between the model and the image may be geometric- or image-based as elaborated in the following sections.

3.2.1 Geometric-based Measures

Geometric-based approaches rely on the presence and identification of natural landmarks or fiducial markers in the input datasets in order to determine the best alignment. The small number of features to be registered could provide great computational efficiency. However, major drawbacks include the need for careful planning of image acquisition, re-scanning of the patient if the images do not contain the fiducial markers, the inconvenience of planting artificial markers on the patient and possible introduction of more errors from segmentation. Moreover, these solutions may also require some level of user interaction which generally is troublesome for medical procedures.

The contour-based algorithms [Low91, LS95, FAB97, GKBT98, CBRS99] try to match contours of the 3D model with those in the image for the registration between them.

David Lowe proposed a registration algorithm for estimating the pose of a hand drill in intensity image using a 3D parametric model of the object [Low91]. After a 3D rigid transformation followed by a perspective projection of the model, this method computes and matches the contours of the model with the corresponding contours in the image. In this algorithm, the parameters of the 3D rigid transformation is computed by minimizing the perpendicular distances between the edge points of the image from the tangent lines to the 2D contour of the projected model. The minimization is performed by Newton method and Levenberg-Marquads least square method [CZ96]. This algorithm produces a good result in a few iterations. However, it requires a camera of known focal length and a good initial estimate of the parameters for convergence to the global minimum.

Feldmar et al. proposed an algorithm for registering 3D curves such as blood vessels in the 3D MR or CT image to their counter parts in the 2D x-ray images [FAB97]. The goal of this algorithm is to combine the information about the blood vessels in MRI with those in the x-ray image of the same patient. Blood vessels are visible in MRI but they are generally not visible in the x-ray image. To make them visible, an opaque liquid is injected into the vessels before taking the x-ray image. The 2D curves in the x-ray image and the 3D curves in MRI image are extracted before the registration. Then Iterative Closest Point algorithm [BM92a] is applied to register the projected 2D curves and the target 2D curves in the x-ray image.

Czof et al. proposed an algorithm to register a model of human bone (femur) to its x-ray image [CBRS99]. They focused on how to compute a good initial pose of the object for registration. They defined a set of primitive geometric features of the objects contour such as corner points, curve segments, and a combination of them. The model of the object is represented by a large number of 2D views known as aspect views. The predefined features are extracted from each view of the model. The views containing similar contour features are grouped together. Initial pose estimation is performed by comparing features extracted from the input image with the features in the aspect views of the model. This method produced descent result if the number of aspect views is large enough.

Lavallée et al. proposed a method to register a 3D MR or CT model of a patients bone to two x-ray images simultaneously [LS95]. They extracted contours of the x-ray images for registration. The two x-ray images are brought into the same 3D coordinate system to find the 3D x-ray paths from the x-ray source to the corresponding contour pixels in the x-ray images (Figure 2.1). Then, the 3D distance between these lines and the surface of the model are computed. The error function of the registration is the sum of 3D signed distances. The sign of a distance is negative if the corresponding line passes through the surface, and positive if the corresponding line is either tangent to the surface or does not intersect with the surface. The goal is to find the pose parameters such that these distances become zero, i.e., the lines become tangent to the model surface. This method used two x-ray images taken simultaneously using two x-ray sources. The error function is minimized by Levenberg-Marquad method. This method also requires a good initial estimate of the pose of the model to get desired registration.

Guéziec et al. proposed an improved algorithm by computing the 3D distances of the x-ray paths from an apparent contour of the model [GKBT98]. For 3D distance computation, there is no need to perform a search for the surface of the model and hence this algorithm requires fewer computations than the algorithm proposed by Lavallée et al..

3.2.2 Image-based Measures

Image-based measures operate on the pixel or voxel intensities directly. They calculate various statistics using the raw intensity values of the inputs which are then compared in the images to be aligned. Though the number of points to be registered is much greater than in the case of the geometric-based approaches, no feature extraction step is required.

An extensive study of intensity-based similarity measures for 2D-3D applications has evaluated the performance of six different objective functions in matching X-ray fluoroscopy and CT images [PWL$^+$98]. The imaged organ is a phantom spine, and only a user-defined small region of interest is registered at a time. The objective functions evaluated are normalized cross-correlation [LJF$^+$94], entropy of the difference image [BWFL], pattern intensity [WBLF97], mutual information [MCV$^+$97,SHH96,SHH97,VW95], gradient correlation [LJF$^+$94,BB96] and gradient difference. These measures are ranked based upon their accuracy and robustness. The best objective functions are pattern intensity and gradient difference which proved to be the most robust in the presence of soft tissue and surgical instruments. Mutual Information (MI) performed poorly in these experiments. It did not handle partial occlusions and truncations well and its performance further deteriorated when soft tissue was present. The study found two possible explanations for the failures of this similarity measure that has been very successful in the 3D-3D domain. First, MI requires a large set of samples to obtain a good probability density estimate for the underlying entropy calculations, which is lacking in the 2D-3D application. Second, the search space of MI is much larger than what the problem requires and therefore, it is more difficult to recover the required parameters in it.

David LaRose proposed an iterative intensity-based algorithm for registering CT-volume of the pelvis to the x-ray images of a patient to compute the patients pose (i.e., pose of patients hip) [LaR01]. He modeled x-ray attenuation in the imaging process with perspective projection to produce 2D synthetic images of the pelvis from CT-volume. These synthetic images are called digitally reconstructed radiograph (DRR). The patients pose is computed by maximizing the correlations between the input x-ray image and the DRRs by an optimization method known as quasi-Newton method. The correlation measures used for optimization are *Sum of Local Normalized Correlation* (SLNC) and *Variance-weighted sum of Local Normalized Correlation* (VLNC). Starting with an approximate values of the parameters (rotational angles and translations), the algorithm produces a single DRR if there is only one input image or multiple DRRs for multiple input images. Then, it compares the similarity by computing the correlation between the x-ray image and the corresponding DRR. If the similarity is small, the parameter values are updated by the optimization method to produce another DRR (or, a set of DRRs), and this process is repeated. The DRR generation is a time-consuming process. He also proposed a method to produce the DRR efficiently by programming a computer graphics card.

Another intensity-based registration algorithm was proposed by Zöllei et al [ZGW01]. They registered a CT-volume of human head and spine to two x-ray images simultaneously by optimizing mutual information between the x-ray image and the DRR generated from CT image. They used gradient descent method for maximization of the mutual information to compute the pose parameters.

Intensity-based method (Chan et al, 2003 [CCY^{+}03]) is also used for the registration of functional image modalities such as magnetic resonance angiography (MRA) and digital subtraction angiography (DSA). Note that this method does not estimate the pose, but it combines the visual information available in the two images.

Chapter 4

Construction of 3D Wall Model

The first part in the generation of DRWR is the construction of the 3D wall model of the femur from volume images. One of the most commonly used volume image is the CT volume. Even though isotropic scans are now possible with newer machines, they are not routinely performed. Typically, the doctor acquires a CT volume with large slice thickness in order to reduce the scanning time and the radiation dosage of the patient. With anisotropic volume, it is necessary to either (1) interpolate the slices to generate isotropic slices for constructing the 3D wall model, or (2) deform a known 3D mesh model to register with the slices. For the first approach to work, additional constraints based on the femur's surface shape need to be imposed during interpolation. Otherwise, the interpolation will produce incorrect surfaces. But, surface shape is inherent in the known 3D mesh model. Therefore, we adopt the second approach which directly uses the 3D mesh model.

The 3D wall model is constructed using a generic 3D surface mesh model and patient-specific CT volume image of femur. The use of a generic 3D mesh model solves the problem of anisotropy of typical CT volume images. The construction of 3D wall model consists of four stages:

1. Extraction of 2D inner and outer wall contours from the CT slices.
2. Registration of 3D surface model to 2D outer wall contours to obtain 3D outer wall surface.
3. Interpolation of the 2D inner wall contours with minimum wall thickness constraint

to obtain 3D inner wall surface.

4. Conversion of 3D wall surfaces to hybrid 3D wall model.

The following sections describe these stages in more details.

4.1 Extraction of 2D Wall Contours

At first glance, it may seem that the 3D wall model can be produced by thresholding a CT volume image with appropriate thresholds. Our investigation shows that this method is not satisfactory (Figure 4.1). If the threshold is low enough for the boundary wall contours to show up clearly, many trabeculae will remain in the CT volume, resulting in the blurring of the shape features in the DRWR (Figure 4.1a). On the other hand, if the threshold is too high, some boundary contours and shape features will be removed before all the trabeculae can be removed (Figure 4.1b). There is no single threshold that is appropriate for the entire CT volume and it is impractical to manually select different thresholds for different parts of the CT volume.

The inner and outer wall contours need to be extracted from the CT slices before registration to the 3D surface model to create the 3D wall surface. The whole process of 2D wall contours extraction can be split into 3 main steps:

1. Extraction of edges from CT slices using Canny's edge detection algorithm.

2. Extraction of outer and inner wall contours separately using the GVF snake.

3. Stacking of 2D wall contours in the 3D space.

The first step applies a modified Canny's edge detection algorithm by Tian [TCL$^+$] to the images to detect edges. Although Canny's edge detector works well in detecting the outline of the outer and inner walls, it also detects a large number of spurious edges. Such spurious edges will affect the snake's convergence onto the outline of the walls and have to be removed. Increasing smoothing effect to remove the spurious edges will cause loss of edge information at regions with low edge magnitudes such as the femoral head, while simple thresholding based on edge magnitude will also fail. Therefore, a modified Canny's edge algorithm is used.

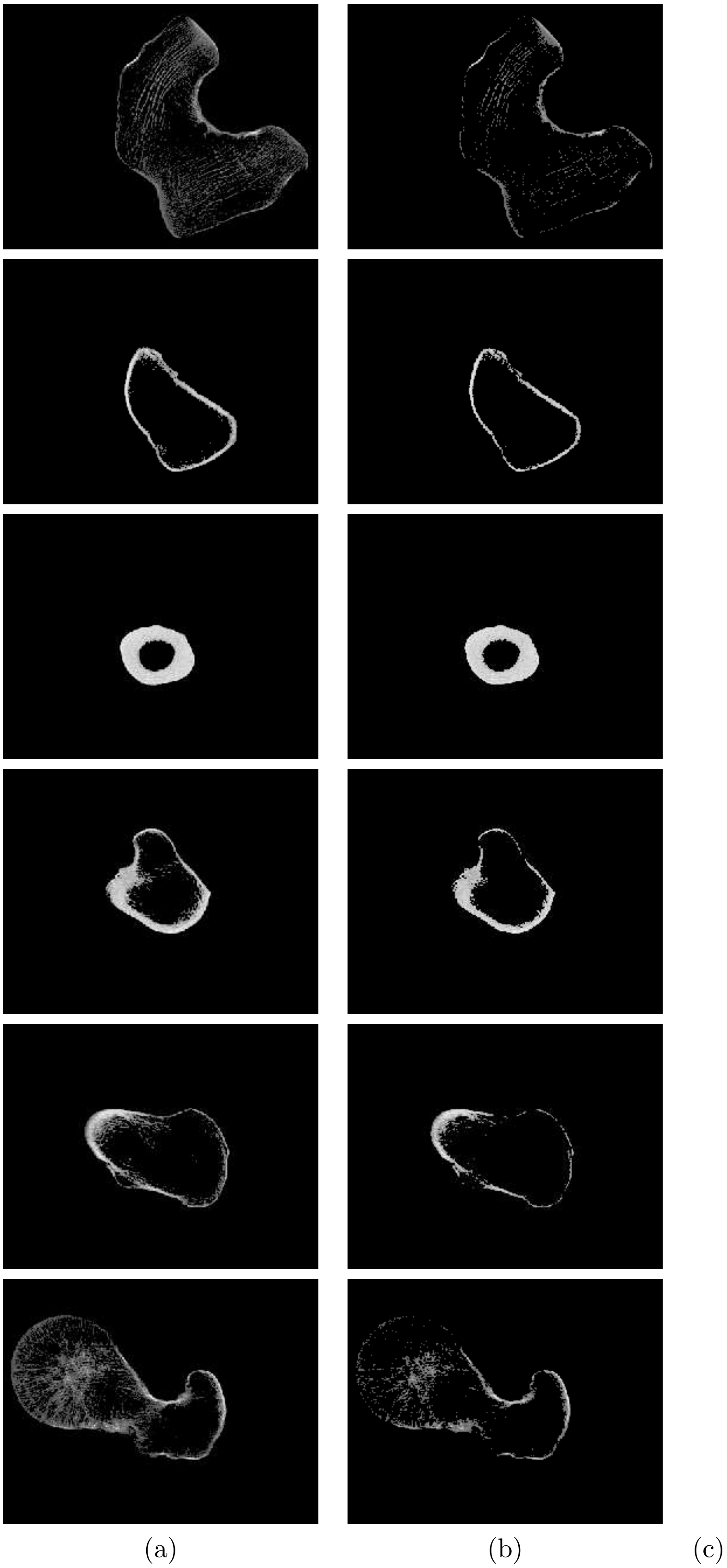


Figure 4.1: Comparison with thresholding method. (a) Low threshold results. (b) High threshold results.

Modified Canny's edge algorithm uses information from the intensity image to help distinguish between spurious edges and the edges at the femoral head. Areas in an x-ray image that contain bones have higher intensity than non-bone regions. Hence, a pixel p is marked as a non-edge pixel if

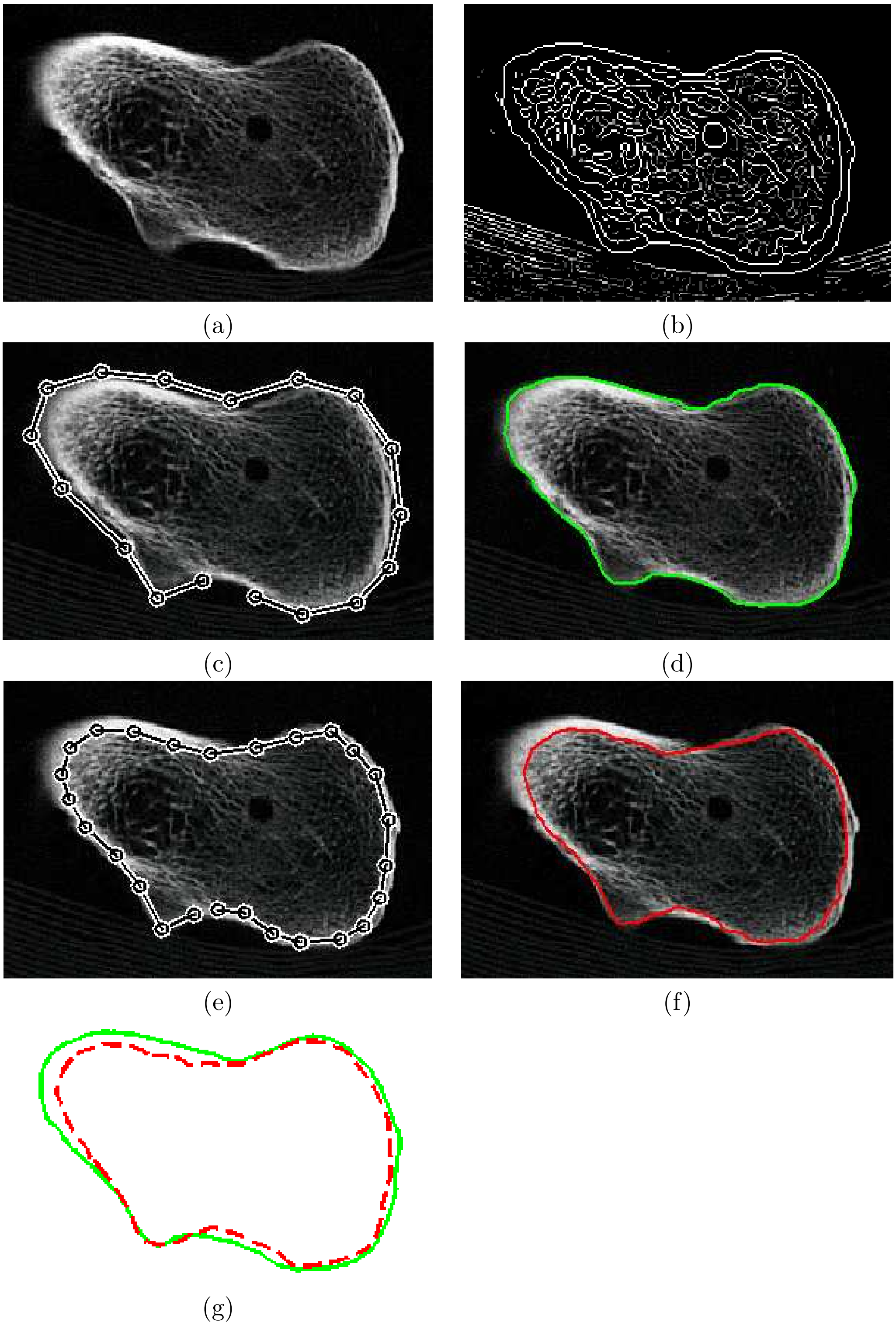
1. p is detected by Canny's edge detector
2. p has an intensity $<$ threshold Γ_1
3. p has an edge magnitude $<$ threshold Γ_2

Further details of the algorithm can be found in [TCL$^+$].

The second step uses GVF snakes [XP98] to extract the outer and inner wall contours. There are a few common segmentation algorithms used for medical imaging like active shapes, active contours (snakes), region-growing and level set. Active shape [SFE00], which is usually used for shape recognition, can also be used for extracting contours. However, it requires a lot of training samples to work well. Region-growing method is easily distorted by the dense trabeculae network in CT images whereas level set [CZ05] adds unnecessary complexity to the problem. On the other hand, active contours perform generally well for medical imaging as long as the initialization is close to the desired object. The original snake formulation has poor convergence to concave regions which is overcome by the incorporation of Gradient Vector Flow.

Active contour (i.e., snake) [KWT87] with Gradient Vector Flow (GVF) [XP98] is manually initialized to extract the inner and outer wall contours separately (Figure 4.2). The GVF field creates forces to attract the snake towards the edges and concave areas, making it less sensitive to initialization. Details about the snake formulation can be found in Sections 4.1.1 and 4.1.2. Thus, sparse initialization points are enough for the contour extraction of the complex shapes of the femur cross-sections. The initial GVF snakes converge to the respective wall boundaries after a few iterations, at interactive speeds. The GVF snake is able to extract the contours of both the outer and inner walls despite the presence of noise and the dense trabeculae (Figure 4.3). However, the GVF extracted inner wall contours may overlap with the outer wall contours at areas where the walls are thin (Figure 4.4).

The last step stacks up these extracted 2D wall contours in the 3D space along the vertical z-axis according to the CT slice thickness (Figure 4.5). As can be seen, shape information in


(a)

(b)

(c)

(d)

(e)

(f)

(g)

Figure 4.2: Extraction of the outer and inner wall contours using the GVF snake. (a) A CT slice. (b) Edge-map of (a). (c) Initialization of the GVF snake for the outer contour (17 points). (d) GVF snake converged to the outer wall boundary (denoted in green). (e) Initialization of the GVF snake for the inner contour (26 points). (f) GVF snake converged to the inner wall boundary (denoted in red). (g) Extracted inner and outer wall contours overlap at some places.

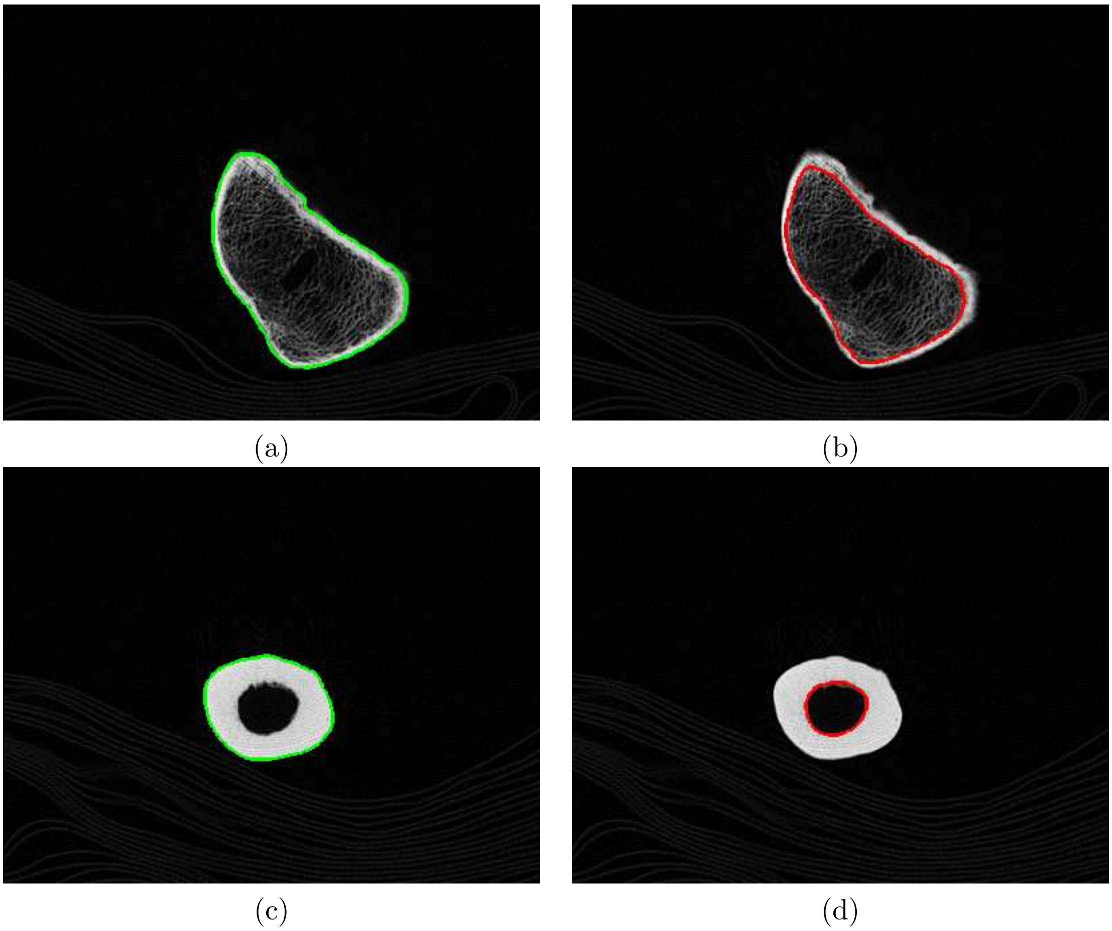
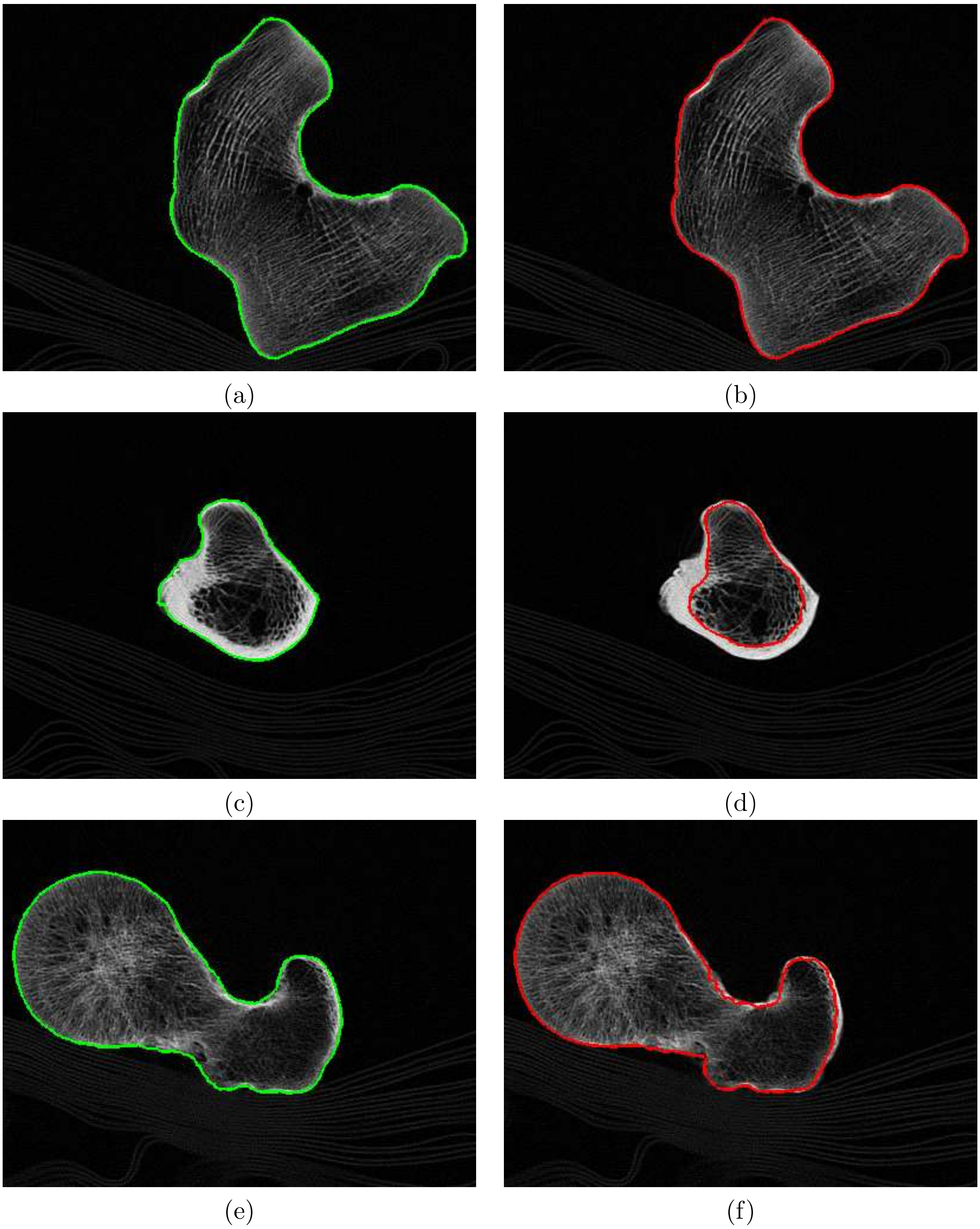


Figure 4.3: Part 1 - Outer and inner wall contours of CT slices extracted by GVF snake. (a,c) Outer wall contours (green). (b,d) Inner wall contours (red).



(a) (b) (c) (d) (e) (f)

Figure 4.4: Part 2 - Outer and inner wall contours of CT slices extracted by GVF snake. (a,c,e) Outer wall contours (green). (b,d,f) Inner wall contours (red). Note that the inner wall contours overlap with the outer wall contours at areas where the walls are thin.

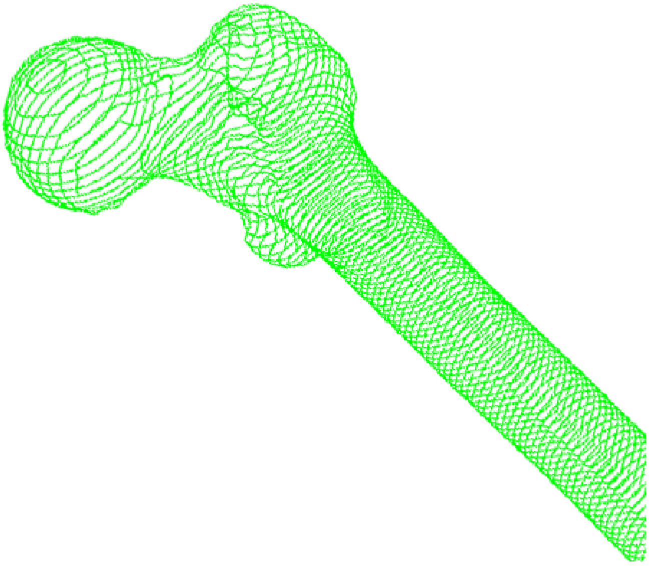


Figure 4.5: A stack of 2D outer wall contours.

between adjacent CT slices is lost. Hence, the next stage in the 3D wall model construction is the construction of the 3D wall surfaces by making use of a 3D generic surface mesh model.

4.1.1 Active Contours (Snakes)

The original snake was first introduced by Kass, Witkin and Terzopoulos in 1987 [KWT87]. The snake model represents a contour v parametrically as

$$v = v(s) = (x(s), y(s)), \quad 0 \leq s \leq 1 \tag{4.1.1}$$

The snake is formulated as an energy-minimizing contour controlled by two types of energies:

1. Internal energy which enforces smoothness constraint.
2. Image energy which attract the contour to the desired features like the edges.

The internal energy, E_{int}, composes of a first-order term controlled by $\alpha(s)$ and a second-order term controlled by $\beta(s)$ (Equation 4.1.2). $\alpha(s)$ characterizes the stretching while $\beta(s)$

characterizes the bending of the contour.

$$E_{int} = \frac{1}{2}(\alpha(s)|v'(s)|^2 + \beta(s)|v''(s)|^2) \tag{4.1.2}$$

The image forces that attract the snake are the edges in the input images. The resultant energy from the edges are denoted by E_{edge}. Therefore, the image energy, E_{image}, corresponds to E_{edge} which will be weighted appropriately by a negative weight, $-w_{edge}$ to attract the snake towards the edges.

$$E_{image} = -w_{edge}E_{edge} \tag{4.1.3}$$

Therefore, the total energy of the snake, E_{snake} is the integral of the sum of the internal energy, E_{int}, and the image energy, E_{image}, in Equations 4.1.2 and 4.1.3.

$$E_{snake} = \int_0^1 E_{int}(v(s)) + E_{image}(v(s))\ ds \tag{4.1.4}$$

When E_{snake} is minimized [KWT87], the snake has snapped onto the desired feature.

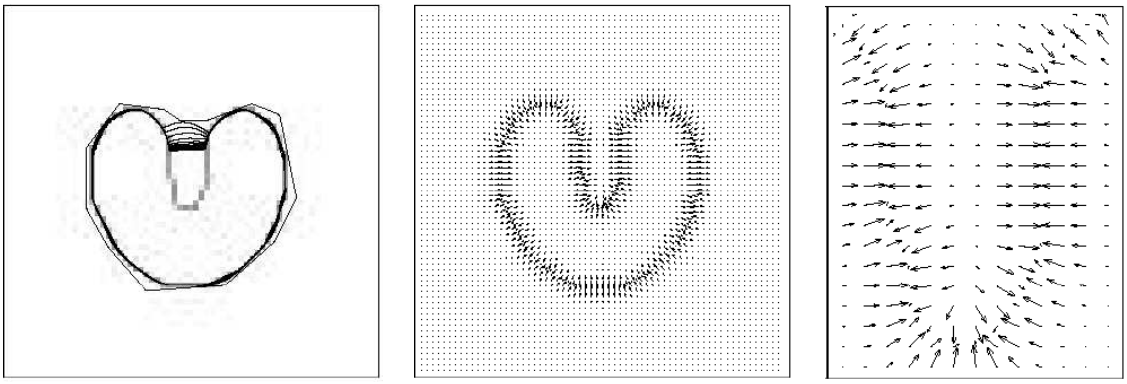
4.1.2 Gradient Vector Flow (GVF)

GVF [XP98] is created to overcome two key shortcomings of the original snake formulation namely, poor convergence to concave boundaries (Figure 4.6) and sensitivity to initialization respectively. GVF is computed as a diffusion of the gradient vectors, ∇E, normal to the edges of a gray-level edge-map, $E(x, y)$, derived from the image, $I(x, y)$. The GVF creates forces to attract the snake towards the edges of concave areas (Figure 4.7).

Fewer initialization points are needed to run the GVF snake as compared to the original snake (Figure 4.8). An accurate outline of the femur can be obtained using the GVF snake (Figure 4.8(c)) whereas it is difficult for the traditional snake to snap onto the concave structure at the femoral neck (Figure 4.8(b)).

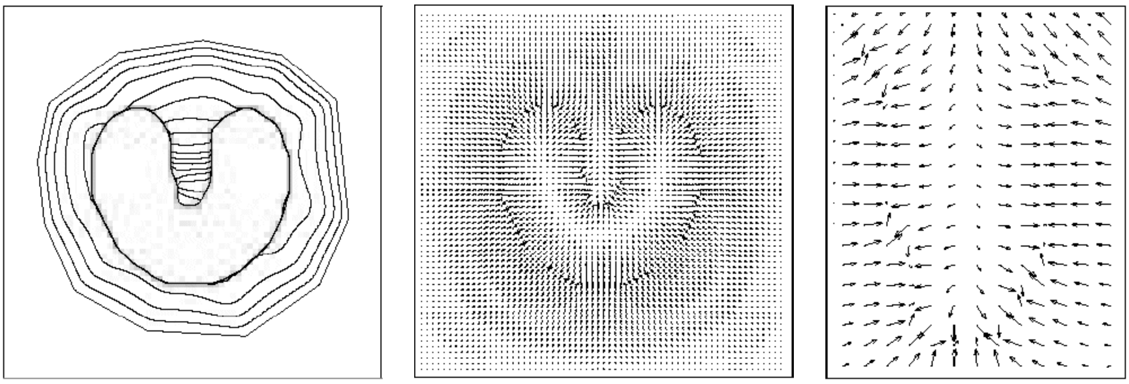
4.2 Construction of 3D Outer Wall Surface

After extracting the 2D outer wall contours, this stage makes use of a generic 3D surface mesh model to construct the 3D outer wall surface. The 3D outer wall surface is constructed as the 3D envelop of the stack of 2D outer wall contours extracted by registering the



(a) (b) (c)

Figure 4.6: Behaviour of traditional snake. (a) Convergence of contour. (b) Traditional potential force. (c) Close-up at concavity: no force to attract the snake towards the bottom. (Adapted from [XP98].)



(a) (b) (c)

Figure 4.7: Behaviour of the GVF snake. (a) Convergence of contour. (b) GVF external force. (c) Close-up at concavity: forces exist to attract the snake towards the bottom. (Adapted from [XP98].)

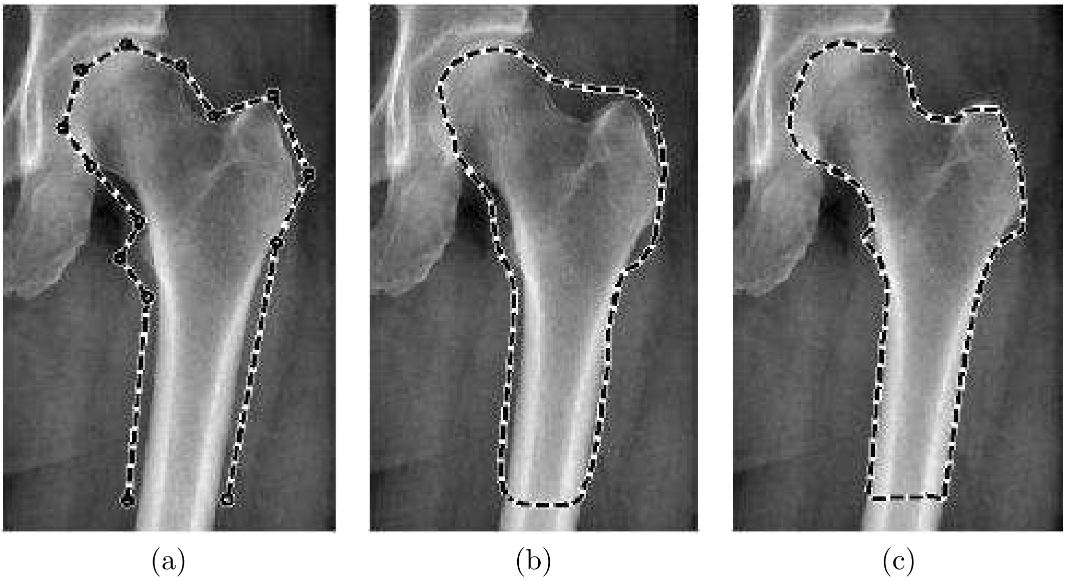


Figure 4.8: Comparison of the performance of the traditional and GVF snakes. (a) Initialization of the snake. (b) The traditional snake result. (c) The GVF snake result. (Adapted from [TCL$^+$].)

generic 3D surface mesh model to the stack of 2D contours. First, an optimal similarity transformation is computed using Iterative Closest Point (ICP) [BM92b] to scale, translate, and rotate the 3D surface model to globally align to the stack of outer wall contours. The main ideas behind ICP are:

Given two sets of 3D points M and O

1. For every point p_i in M, find the closest point p'_k in O using k-d tree.
2. Find the similarity transformation that best maps p_i to p'_k.
3. Apply that transformation to all points in M.
4. Iterate above three steps until convergence.

Although simple, the algorithm works quite effectively when given a good initial estimate. The distance function used in the first step is the Euclidean distance. Given a point p_i in M, the closest point p'_k in O to p_i satisfies

$$d(p_i, p'_k) = \min_{p'_j \in O} d(p_i, p'_j) \tag{4.2.1}$$

where d is the Euclidean distance function.

Let $f(p_i)$ denote the closest point in O to p_i where f is the closest point function. Therefore, each point p_i in M is matched to a point $f(p_i)$ in O. Registration is performed with point correspondence to find the best similarity transformation that register M with O. The error function is given by

$$E(t) = \sum_{p_i(t) \in M(t)} [s(t)R(t)p_i(t) + T(t) - f(p_i(t))]^2 \tag{4.2.2}$$

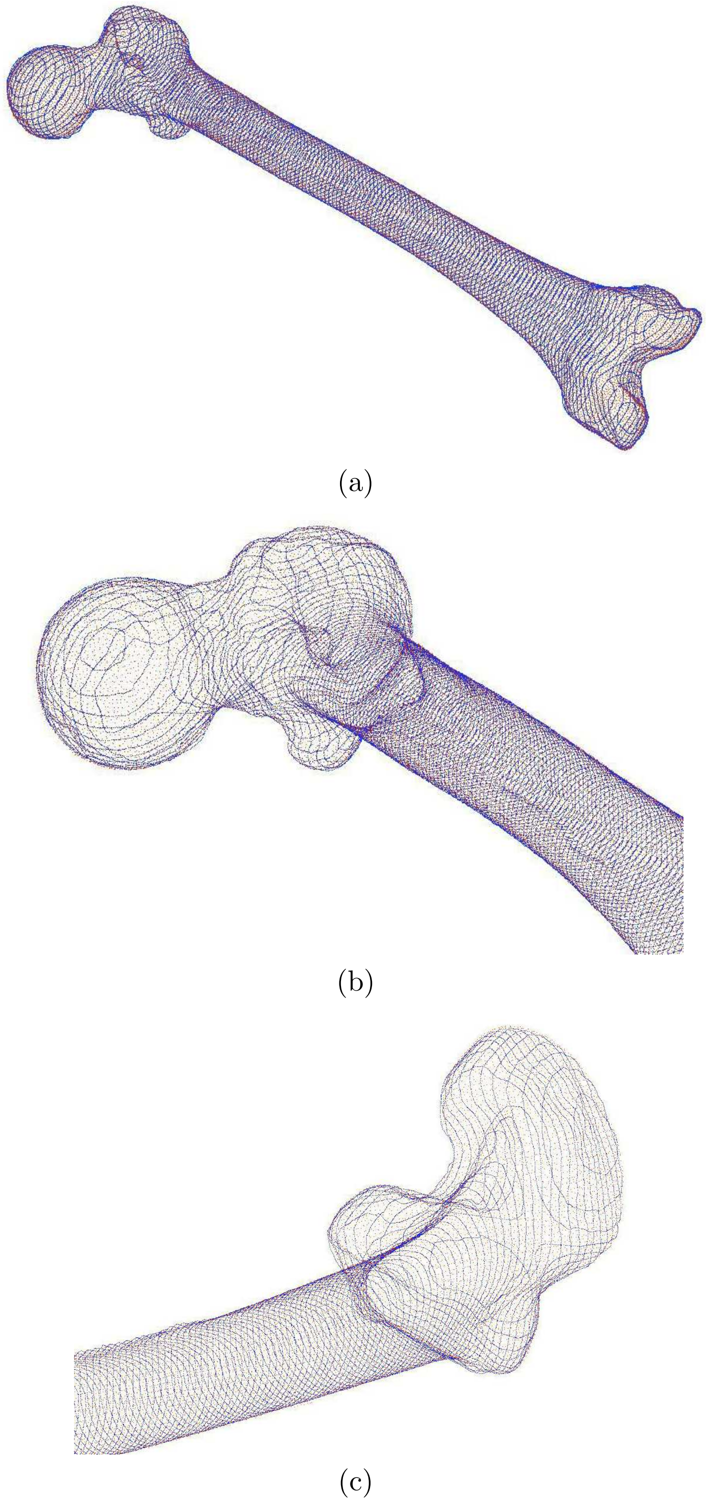
where s, R and T are the scaling, rotation and translation respectively. These steps are repeated until $E(t)$ or $E(t) - E(t-1)$ is small enough. The registered deformed 3D model then becomes the 3D outer wall surface.

In principle, the envelop of the outer wall contours can also be computed using existing algorithms such as "balloon" (3D version of snake) and level set method [Set96]. However, in practice, these algorithms are not suitable for this problem. The balloon algorithm requires the inversion of an $n \times n$ matrix where n is the number of points in the mesh, which is computationally very expensive for large n. On the other hand, the 3D surface deformed by the level set method tends to be too flexible and can easily fold into the gaps between slices, especially when the slice thickness is large. Therefore, it is computationally more efficient and robust to deform the 3D mesh model using ICP as discussed above. In this way, the deformed model still retains the overall shape of the original model, implicitly imposing shape constraint on the entire model.

Both correspondences from the contours to the mesh model (Figure 4.9) and from the mesh model to the contours (Figure 4.10) are used in the ICP registration. The registration results from using these two correspondences are comparable, and hence either result can be used to form the outer wall surface and to refine the extracted inner wall contours.

4.3 Construction of 3D Inner Wall Surface

3D inner wall surface is constructed as the 3D envelop of the stack of 2D inner wall contours. The algorithm is similar to that used for constructing 3D outer wall surface except that the *minimum wall thickness* constraint is imposed. This is required because the inner and outer wall contours are extracted separately without any constraint between them. As a result, when the wall of the bone is very thin as at the upper and lower extremities of the

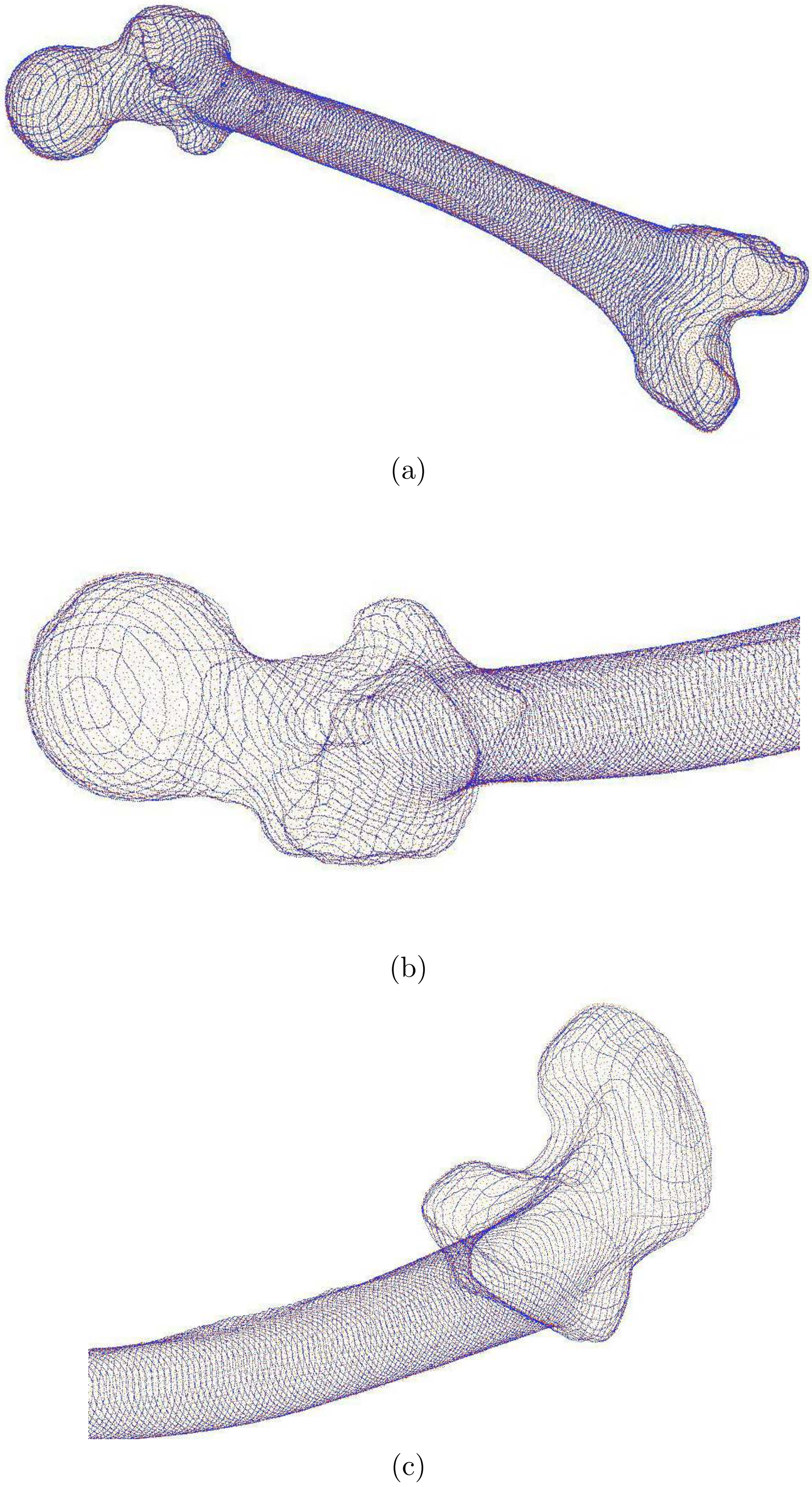


(a)

(b)

(c)

Figure 4.9: ICP registration of extracted outer wall contours (blue) to 3D mesh model (red).



(a)

(b)

(c)

Figure 4.10: ICP registration of 3D mesh model (red) to extracted outer wall contours (blue).

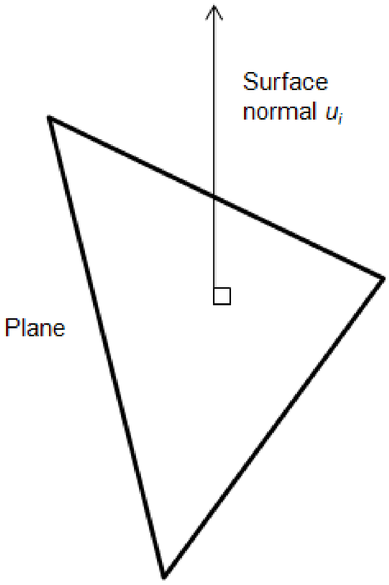


Figure 4.11: A polygon and one of its two normal vectors.

femur, the two contours may overlap and cross over (Figures 4.2(g) and 4.4). The constraint ensures that the inner wall surface always lie inside the outer wall surface.

The algorithm for constructing 3D inner wall surface is as follows:

1. Deform the outer wall surface by moving each mesh point inward by a constant amount equal to the minimum wall thickness w. Call the deformed surface the *initial inner wall surface.*

2. For each CT slice, determine the inner wall contour points that satisfy the minimum wall thickness constraint.

3. Interpolate the inner wall contours to form the inner wall surface.

Step 1 is performed by deforming the 3D outer wall surface obtained in Section 4.2 such that each point p_i is moved to p'_i defined as the point on the surface normal u_i in the inward direction (Figure 4.11) and at a distance w from p_i. The outer wall surface deformed in this manner becomes the initial inner wall surface.

The initial inner wall surface lies strictly inside the outer wall surface. But, parts of the inner wall contour extracted using snake algorithm may lie outside the initial inner wall

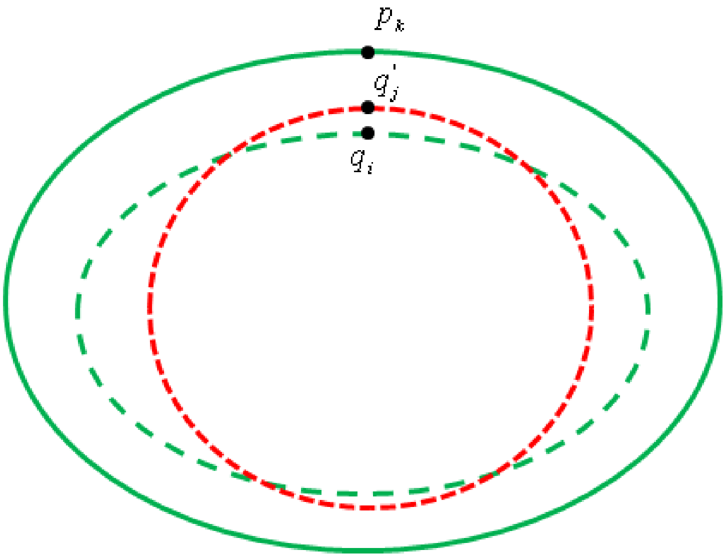


Figure 4.12: Refined inner wall contour that lies strictly within the outer wall contour. For each inner wall contour point q_j' on the extracted snake contour (dashed red), the closest corresponding points p_k and q_i on the outer wall surface (solid green) and initial inner wall surface (dashed green) are found respectively. The point further from p_k is chosen to be the refined inner wall contour point q_j^*.

surface. Step 2 resolves this problem by choosing the points that are further from the outer wall as the refined inner wall contour points (Figure 4.12):

For each inner wall contour point q_j' of each CT slice,

a. Find the closest point q_i on the initial inner wall surface and the closest point p_k on the outer wall surface.

b. Determine whether q_j' or q_i is further from p_k.

c. Use the point further from p_k as the refined inner wall contour point q_j^*.

Results of this refinement stage show that a minimum wall thickness of the 3D wall model is maintained (Figure 4.13). More examples at of other parts of the femur bone are shown (Figures 4.14 and 4.15).

After determining the refined inner wall contour points q_j^*, the final wall contours are then stacked up accordingly (Figure 4.16). Step 3 linearly interpolates the refined inner wall

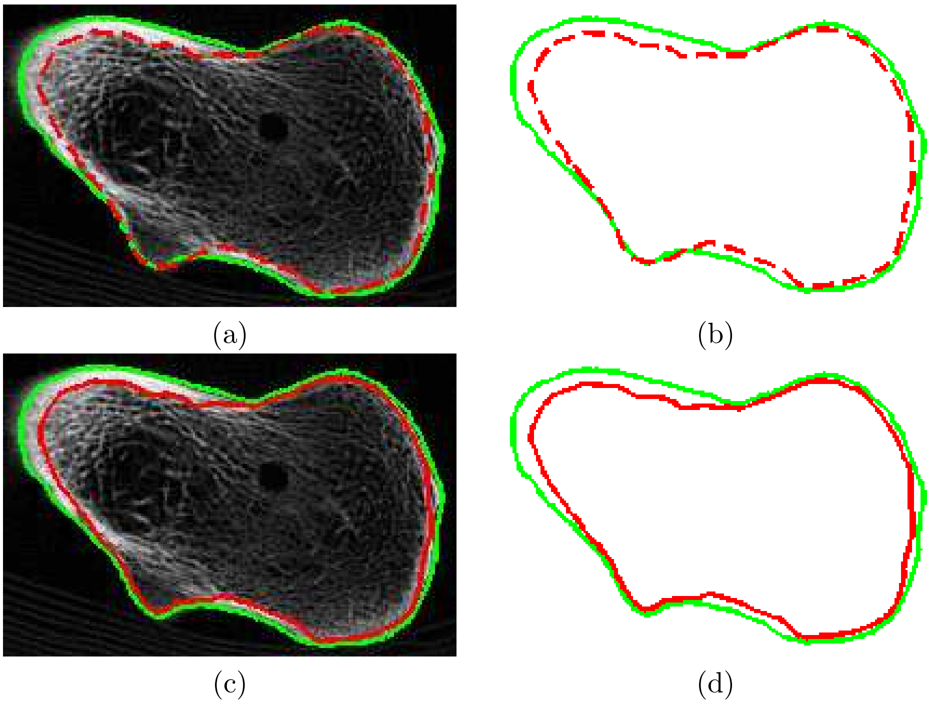


Figure 4.13: Refinement of the inner wall contours. (a, b) Extracted wall contours. (c,d) Refined wall contours.

contours to form the inner wall surface.

4.4 Conversion to Hybrid 3D Model

The final stage converts the 3D wall surfaces to a hybrid 3D wall model. First, a minimum bounding volume of the 3D wall surfaces is determined. The volume is discretized into square voxels, and values representing x-ray attenuation coefficients are assigned. The voxels outside the outer wall surface and inside the inner wall surface are assigned zero values. The voxels between the inner and outer wall surfaces are assigned a constant non-zero value by flood-filling. 4-way flood-fill is used because the 8-way flood-fill will leak through slopes of one pixel thick (Figure 4.17). The cross-sections of the 3D wall model are shown (Figure 4.18).

Due to the thick shaft wall, shaft region in the DRWR is very bright and obscures the shape features near it (Figure 4.19). So, a refinement process can be applied as follows. If the sum of the voxel values in a small local volume centered at voxel **v** exceeds a pre-defined

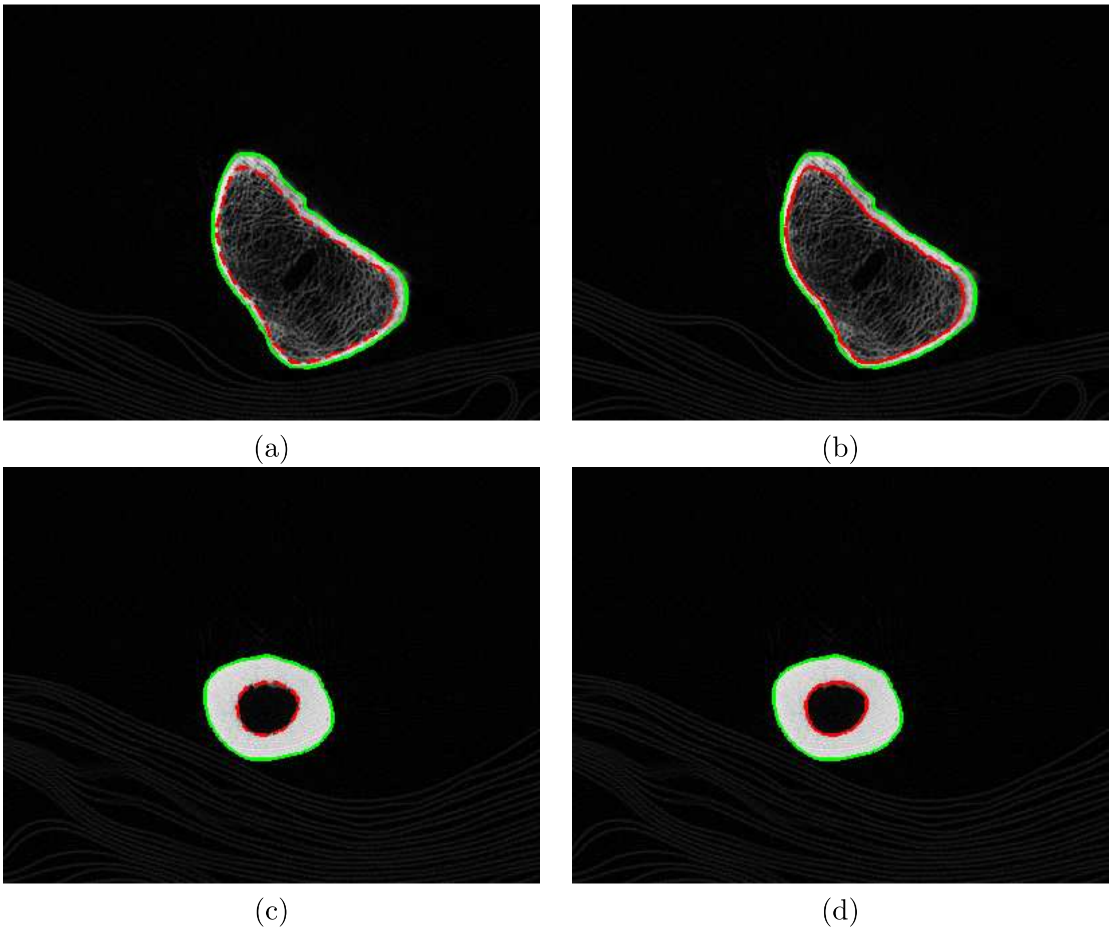


Figure 4.14: Part 1 - Comparison of GVF extracted and refined contours. (a,c) GVF extracted contours (green solid and red dashed). (b,d) Refined inner wall contours (green and red solid).

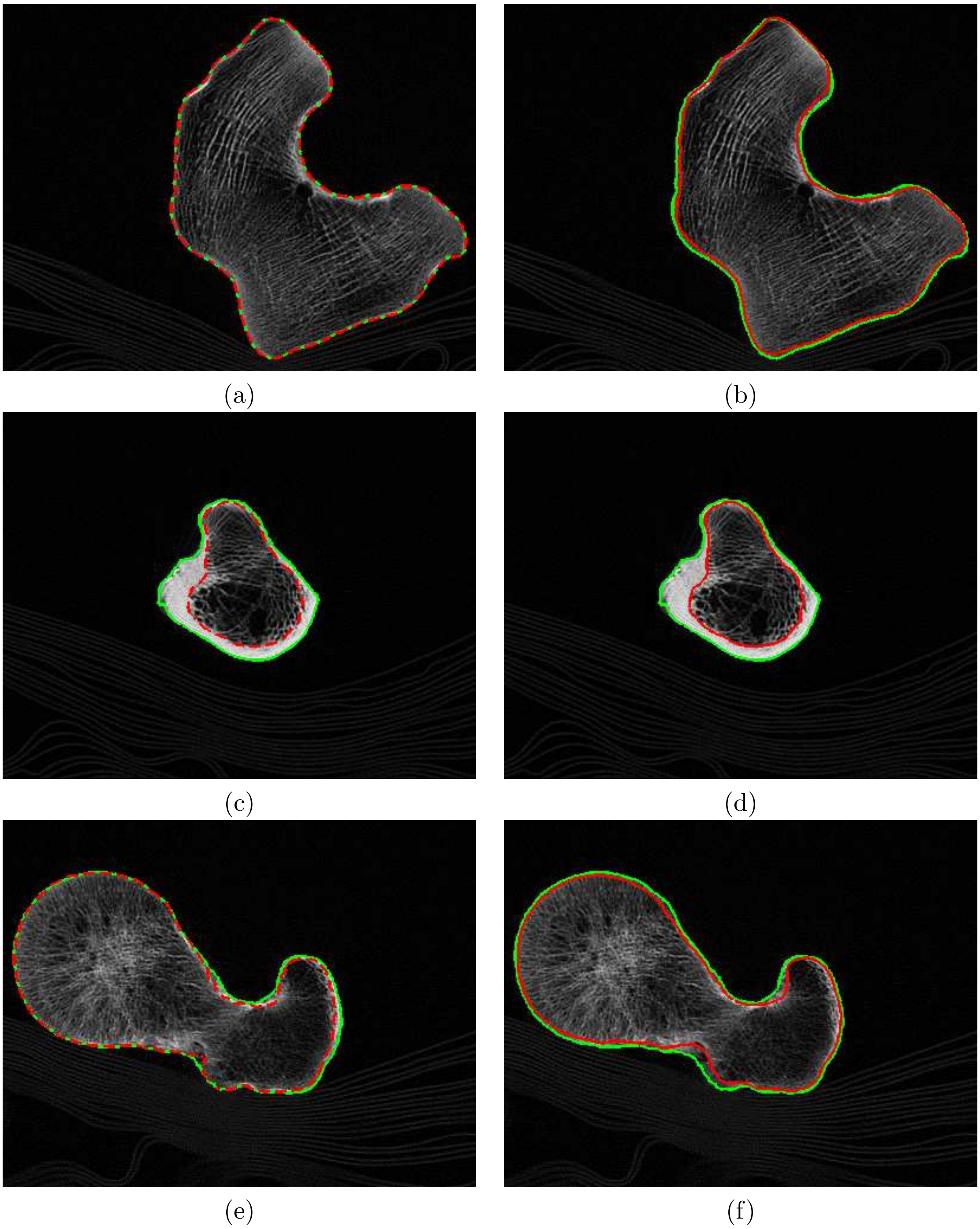
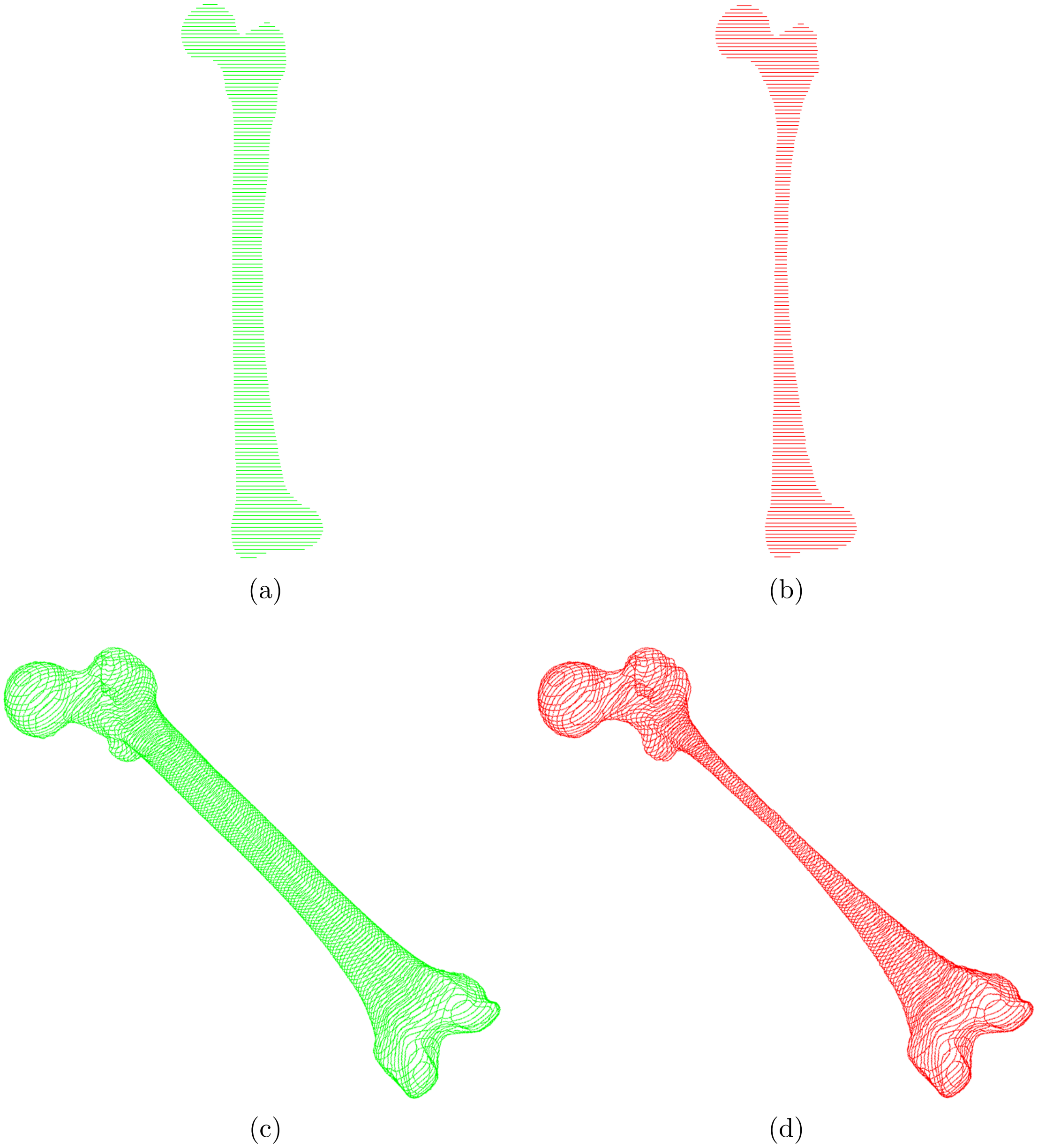


Figure 4.15: Part 2 - Comparison of GVF extracted and refined contours. (a,c,e) GVF extracted contours (green solid and red dashed). (b,d,f) Refined inner wall contours (green and red solid). The refinement has removed the overlapping of the outer and inner wall contours.


(a)

(b)

(c)

(d)

Figure 4.16: Stacks of final outer and inner wall contours. (a, c) Stack of outer wall contours (green). (b, d) Stack of refined inner wall contours (red).

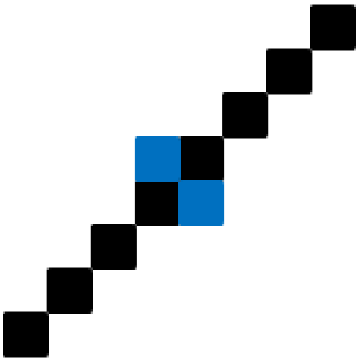


Figure 4.17: Both blue pixels are neighbours in the 8-way flood-fill, resulting in leakage out of the one-pixel thick slope (denoted in black).

threshold, voxel $\mathbf{v}$ is assigned a very small value. This normalization process lowers the voxel values of the shaft region without changing the voxel values at the other parts.

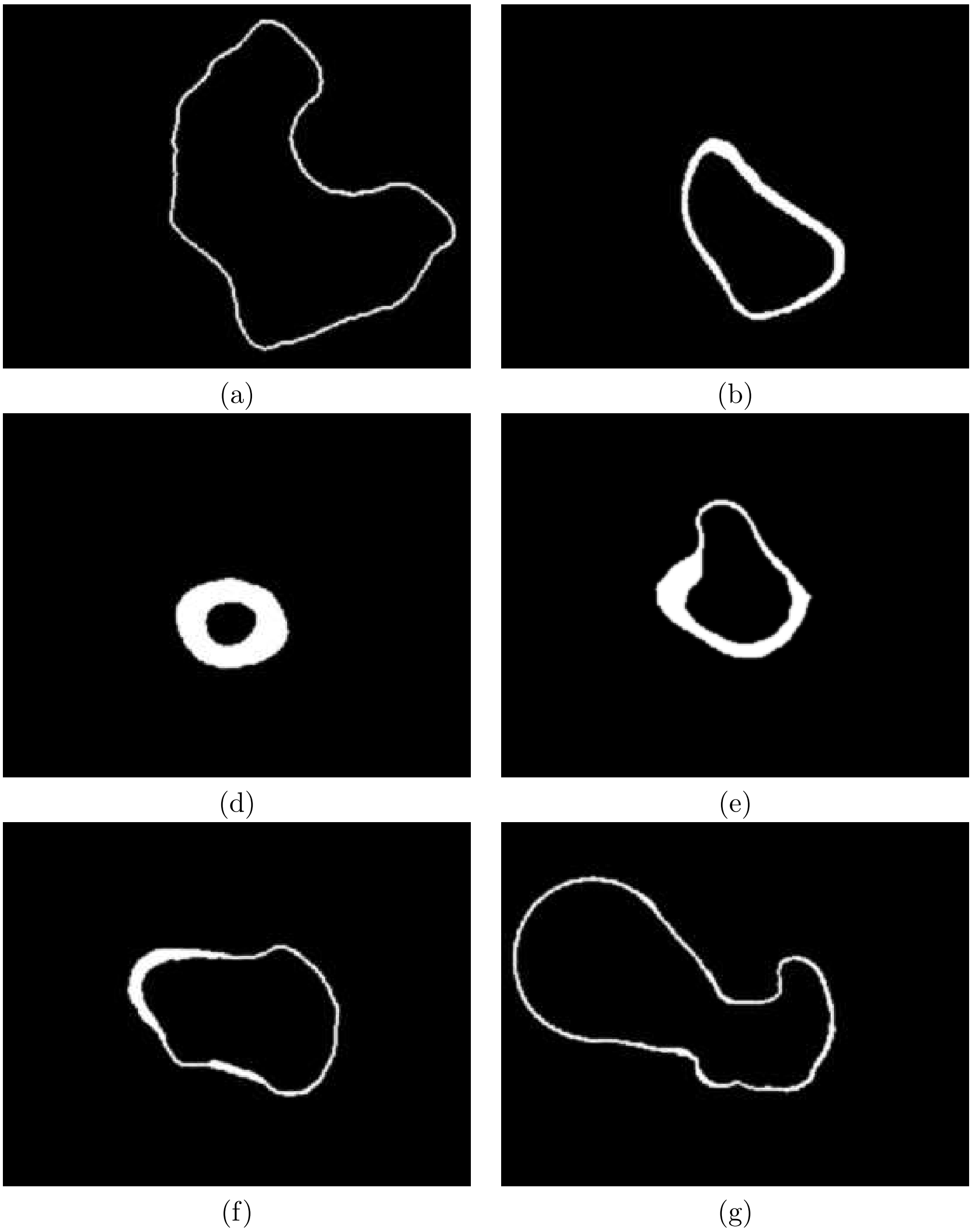
To generate DRWR with smooth edges, anti-aliasing should be applied to the voxels at the wall surfaces. In the current implementation, a simple approximation is adopted by Gaussian smoothing the voxel values in 2D (Figure 4.18). Gaussian smoothing is a filter that uses a normal distribution, also called Gaussian distribution, for calculating the transformation to apply to each pixel in an image (Figure 4.20). The Gaussian distribution in 1D is given by

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-x^2/(2\sigma^2)} \tag{4.4.1}$$

The 2D Gaussian distribution is given by

$$G(x, y, z) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/(2\sigma^2)} \tag{4.4.2}$$

Each voxel value is a weighted average of its neighborhood. The voxel at the center of the Gaussian receives the heaviest weight, and the further the distance of the neighboring voxels to the center voxel, the smaller the weights assigned to them. This results in a smoothing that preserves boundaries and edges better than other more uniform smoothing filters.



(a) (b) (d) (e) (f) (g)

Figure 4.18: Cross-sections of the 3D wall model.

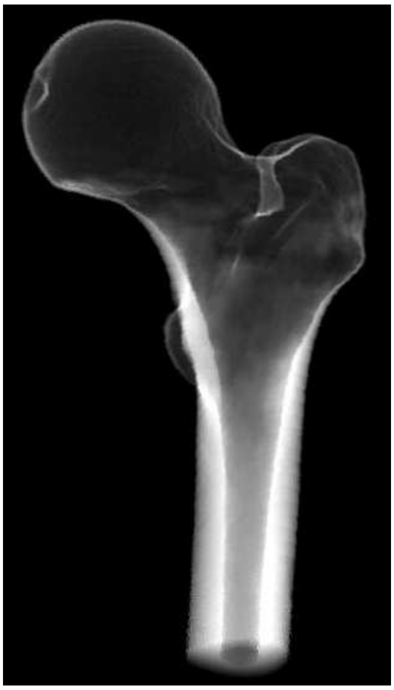


Figure 4.19: Shape features obscured by overly bright shaft region in DRWR.

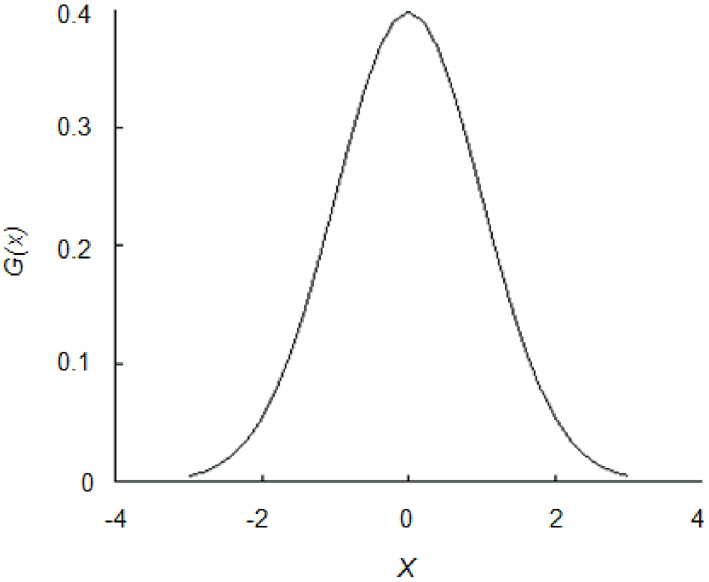


Figure 4.20: 1D Gaussian distribution with $mean = 0$ and $\sigma = 1$.

Chapter 5

Rendering of DRWR

The standard rendering technique used to generate DRRs involve conventional ray casting which simulates the actual x-ray imaging process. Time efficiency is not an issue in this case because hardware acceleration of the rendering algorithm is applied. The accuracy of the algorithm is of more importance. Therefore, the conventional ray casting technique without any optimization is the choice here.

5.1 The Ray Casting Technique

The x-ray interactions involved in diagnostic x-ray imaging are as mentioned earlier in Section 2.2. Each ray starts at the radiation source P_0, and passes through space to a particular point on the imager P_3 (Figure 5.1). P_1 and P_2 represent the points at which the ray enters and exits the volumetric model. The total attenuation of the radiation incident on each point on the imager can be calculated by summing up the ray attenuation along the line joining that point and the radiation source.

Assume that the position and orientation of the volume is given by a vector γ. The synthetic x-ray image is basically produced by considering each pixel independently and computing the log total attenuation along the ray which traces from the corresponding point on the surface of the imager back to the x-ray source. This process comprises four steps as follows:

1. Select a point on the imager P_3 which corresponds to the center of the current pixel.

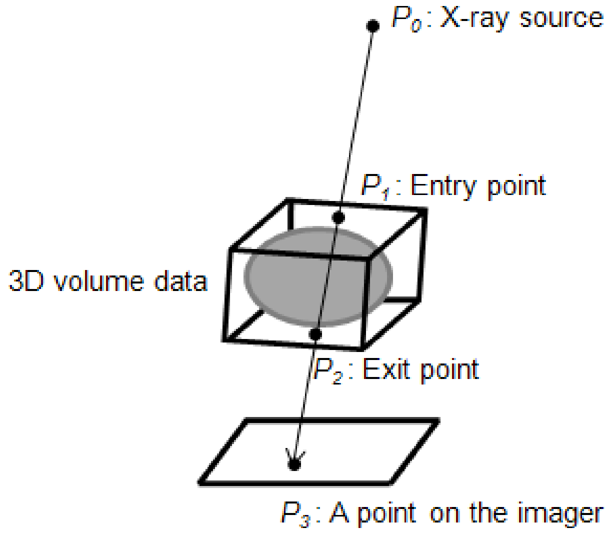


Figure 5.1: Path of x-ray from the source to the imager.

2. Compute the ray which connects P_3 to the x-ray source P_0.

3. Find points P_1 and P_2 at which the ray enters and exits the CT volume. This can be done by first creating a bounding box around the CT volume. P_1 and P_2 can then be determined as the intersection points of the ray equation with the planes of the bounding box. Note that these points depend on γ, P_0 and P_3.

4. Perform summation along the length of the computed ray $R(P_0, P_3)$. The quantity integrated is the linear attenuation coefficient at each point.

When performing the last step, the linear attenuation coefficient of air should be used for points outside the volume. Therefore, the log total attenuation U_T along the ray $R(P_0, P_3)$ from P_0 to P_3 will be

$$U_T(R(P_0, P_3)) = \|P_1 - P_0\|\mu_0 + \|P_3 - P_2\|\mu_0 + U(P_1, P_2) \tag{5.1.1}$$

$$U(P_1, P_2) = \sum_{P \epsilon R(P_1, P_2)} \mu(P) \tag{5.1.2}$$

The constants μ_0 and $\mu(P)$ are the linear attenuation coefficient of air and linear attenuation coefficient derived from the 3D wall model's intensity value at point P respectively.

For nearly all practical cases, μ_0 is equal to zero [LaR01]. Therefore, Equation 5.1.1 becomes

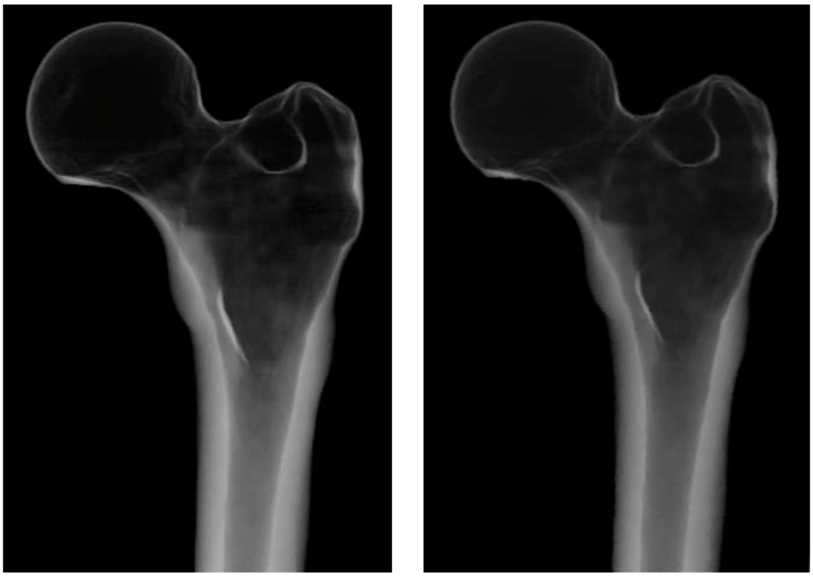
$$U_T(R(P_0, P_3)) = U(P_1, P_2) = \sum_{P \epsilon R(P_1, P_2)} \mu(P) \tag{5.1.3}$$

The 3D wall model is fixed while the x-ray source and image plane are rotated and translated for generating DRWRs from varying viewpoints, using the above-mentioned algorithm. This is different from Larose's volume rendering algorithm whereby he has implemented an additional 4D data structure called the *Transgraph* for software speedup [LaR01]. Details are already discussed in related work (Section 3.1). Software speedup is not implemented in this thesis because the ray-casting technique can be accelerated via hardware as follows, which will greatly improve the performance of the algorithm.

5.2 Hardware Acceleration by GPU

The ray-casting method described in the previous section can be hardware-accelerated by the Graphics Processing Unit (GPU) to generate DRWR. In the current implementation by our project team member [Cha06], an AMD Athlon(tm) 64 Processor, CPU 2.04 GHz with 1GB of RAM and an Nvidia GeForce 7600GS Sonic graphics card with G73 GPU is used. The Cg language is used to code the GPU shader programs while C++ and OpenGL are used to organize the data pipeline in the CPU. The whole ray-casting process is encapsulated in the GPU. This implementation takes advantage of the new characteristics of the G73 GPU, such as reading texture data in vertex programs and supporting longer lengths for shader programs, to implement volume rendering. With GPU implementation, the generation of DRWRs at varying viewpoints can be achieved in real-time.

Using the above system, a 512 × 512 DRWR is rendered via software ray-casting implemented in Java. Another DRWR is hardware-rendered using the GPU-based volume rendering. The DRWRs rendered by both methods are comparable in terms of image quality (Figure 5.2). The respective execution times are also shown below in Table 5.1. The time taken to render a DRWR from one viewpoint using the software approach needs an average of 130.6s which is very slow if used for registration purposes. On the other hand, the GPU-based volume rendering renders each pose at interactive rate of 0.324s on the average which is about 400 times faster. Hence, the GPU-based volume rendering is chosen



(a) (b)

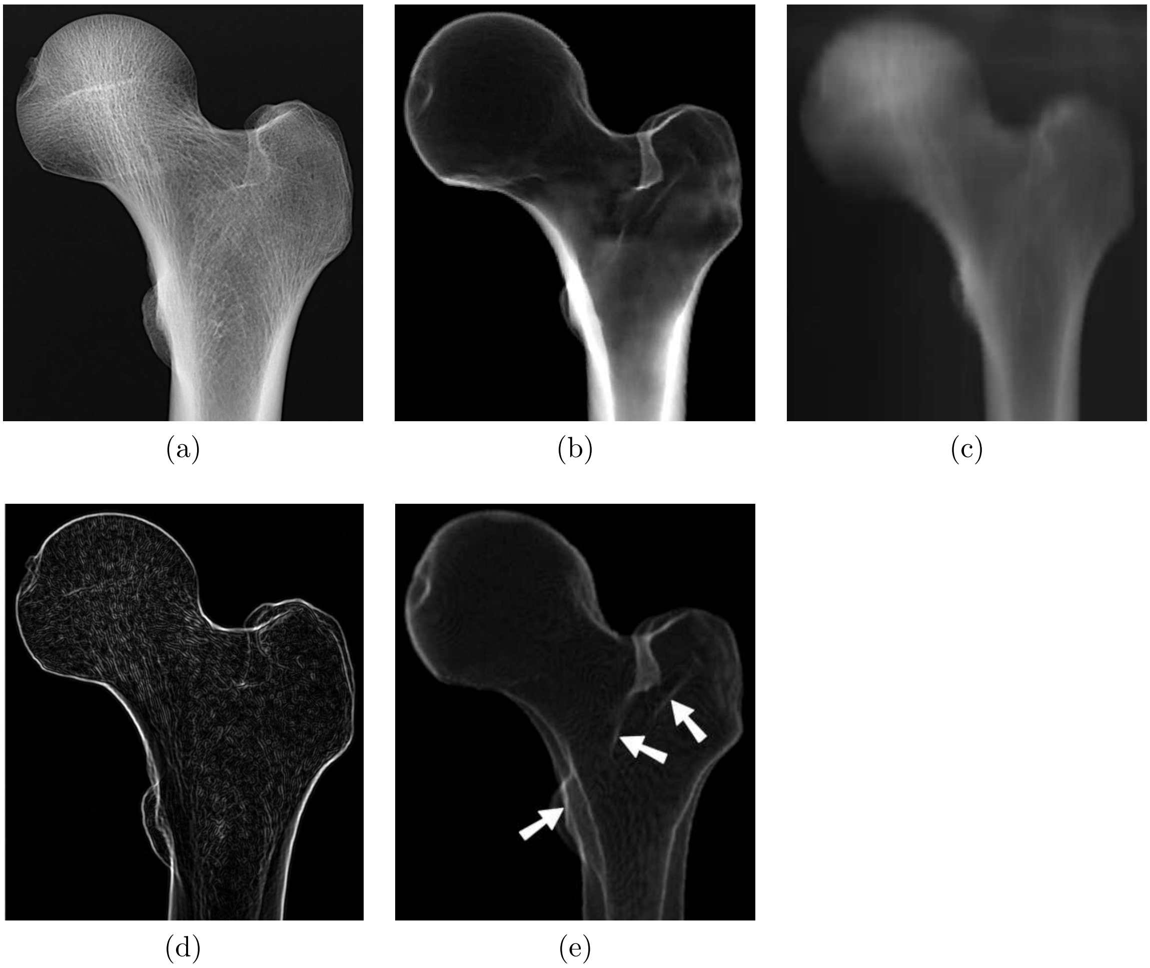
Figure 5.2: Comparison of (a) software-rendered DRWR and (b) GPU-rendered DRWR.

Table 5.1: Comparison of execution times of both software and GPU-based ray-casting

	Execution times t_i (sec)					Mean	Standard Deviation
	t_1	t_2	t_3	t_4	t_5	$\bar{t}$	σ
Software	123	126	133	146	125	130.60	8.41
GPU	0.316	0.333	0.299	0.252	0.418	0.32	0.05

to generate the DRWRs of the 3D wall model because of the remarkable improvement in time efficiency.

For comparison, the trilinearly-interpolated isotropic CT volume was obtained and DRR of the volume was also synthesized. Figure 5.3(c) shows that the shape features in the conventional DRR are blurred and obscured by the trabeculae. The solid-wall DRWR (Figure 5.3(b)) shows clearer shape features than the DRR but the bright shaft region still obscures some features around it. The hollow-wall DRWR (Figure 5.3(e)) obtained after normalization, as described in Section 4.4, shows the clearest and most distinctive shape features. The DRWRs are accurate in the sense that the position and structures of the shape features are accurately reproduced. The absolute intensities are not important because the aim is to highlight only the shape features in the DRWR without the other irrelevant details as in x-ray or DRR.


(a) (b) (c)

(d) (e)

Figure 5.3: Comparison of results. (a) X-ray image. (b) Solid-wall DRWR. (c) Conventional DRR. (d) Intensity gradient image of (a). (e) Hollow-wall DRWR.

In practice, the solid-wall DRWR can be directly registered to an x-ray image (Figure 5.3(a)). On the other hand, the hollow-wall DRWR should be registered to an x-ray image's intensity gradient image (Figure 5.3(d)) because they are similar in image characteristics.

Weighting is often used in image comparison to emphasize areas of importance. For example, Larose et al. [LCJ$^+$00] used a local semi-normalized correlation measure for image comparison. In contrast, weighting is not necessary for DRWR because the shape features are already emphasized in the DRWR, especially the hollow-wall DRWR.

Chapter 6

Evaluation and Comparison

This chapter focuses on the experiments carried out to address the two main objectives of this thesis. The first objective is to evaluate the viability of using DRWRs in registration. The next objective is to compare the degree of match of the solid-wall DRWR to x-ray image with that of both the conventional DRR and the hollow-wall DRWR.

6.1 Viability of using DRWR in registration

The data used in this experiment includes the dry femur from Singapore General Hospital (SGH), its 3D mesh model and its CT images (also provided by SGH). The 3D surface mesh model (Figure 6.1) was captured by a project team member by scanning a dry femur using the Minolta 3D laser scanner. Twelve different views of the femur were captured by the scanner, and merged using the software, Polygon Editing Tool, provided by Minolta. The 3D mesh model of the femur contains approximately 142,000 points and 286,000 triangles. For algorithmic efficiency, the 3D mesh model was down-sampled by reducing the number of points without significant loss of geometry of the model. The dry femur was also scanned in SGH to produce a CT volume with a slice thickness of about ten pixels ($3mm$) in the image plane. When stacked in an isotropic manner, only one-tenth of the slices were present.

Given the x-ray image of the dry femur taken at a particular pose, a DRWR at a similar pose is rendered (Figure 6.2). The pose parameters of the DRWR are 132°, 2° and 96° about the x, y and z axes respectively. More DRWRs are taken at varying poses, changing only one

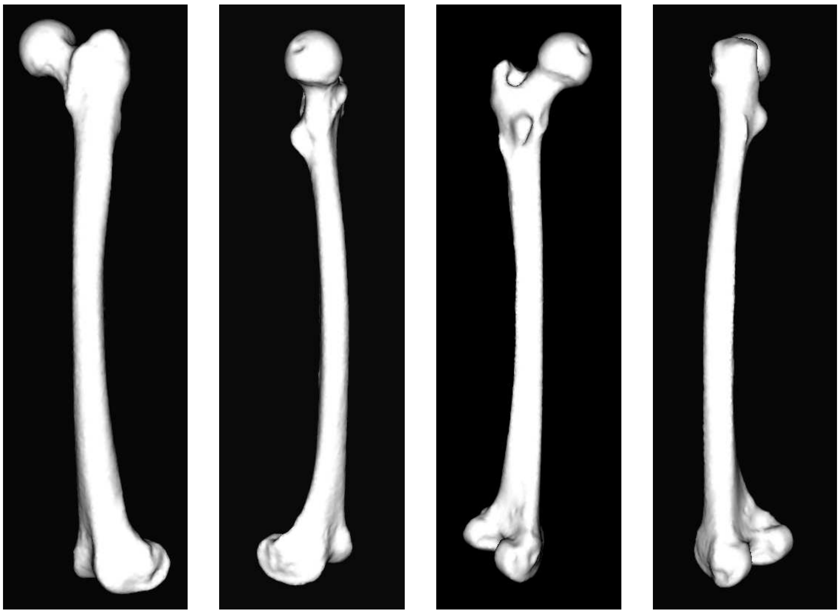
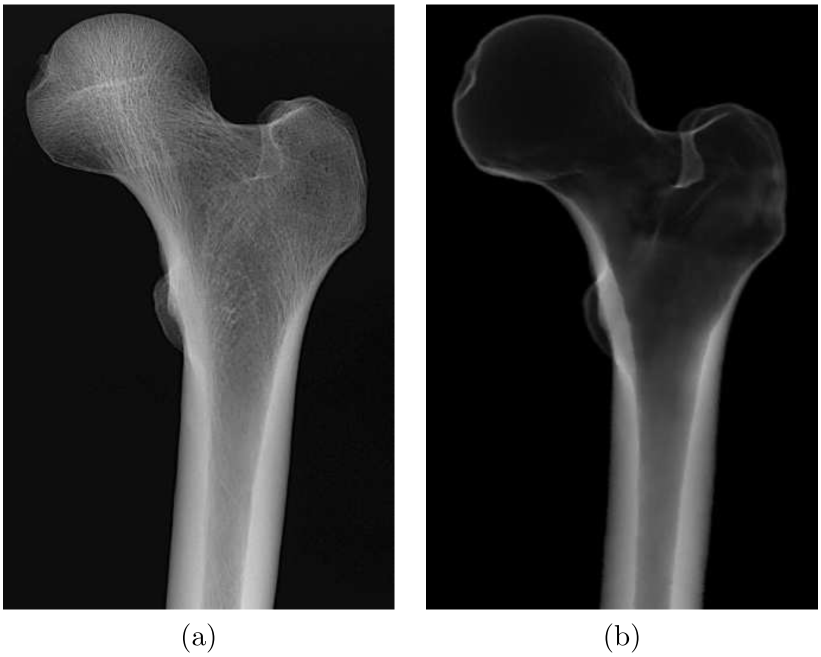


Figure 6.1: 3D surface mesh model of a dry femur at varying viewpoints.



(a)

(b)

Figure 6.2: X-ray and solid-wall DRWR images of the dry femur. (a) X-ray image. (b) Solid-wall DRWR.

parameter at a time. These DRWRs are then compared to the x-ray image shown in Figure 6.2(a) using three metrics written in Matlab code for determining the degree of match: the Root Mean Square (RMS) of the difference (Equation 6.1.1), the Mutual Information (MI) (Equation 6.1.2) and the Variance-weighted Local Normalized Correlation (VLNC) (Equation 6.1.3) used by Larose [LCJ$^+$00]. Given two 2D images I_0 and I_1 where I_1 is the control image, the RMS value is given by:

$$RMS(I_0, I_1) = \sqrt{\frac{1}{m \times n} \sum_{i=1}^{m} \sum_{j=1}^{n} (I_0(i,j) - I_1(i,j))^2} \tag{6.1.1}$$

Similarly, the MI value [PMV03] will be:

$$MI(I_0, I_1) = H(I_1) - H(I_1|I_0) \tag{6.1.2}$$

where $H(I)$ measures the entropy of image I.

Lastly, the VLNC value is computed as:

$$VLNC(I_0, I_1) = \frac{\sum_{p\epsilon Q} C(I_1, I_1, P(p)) * NC(I_0, I_1, P(p))}{\sum_{p\epsilon Q} C(I_1, I_1, P(p))} \tag{6.1.3}$$

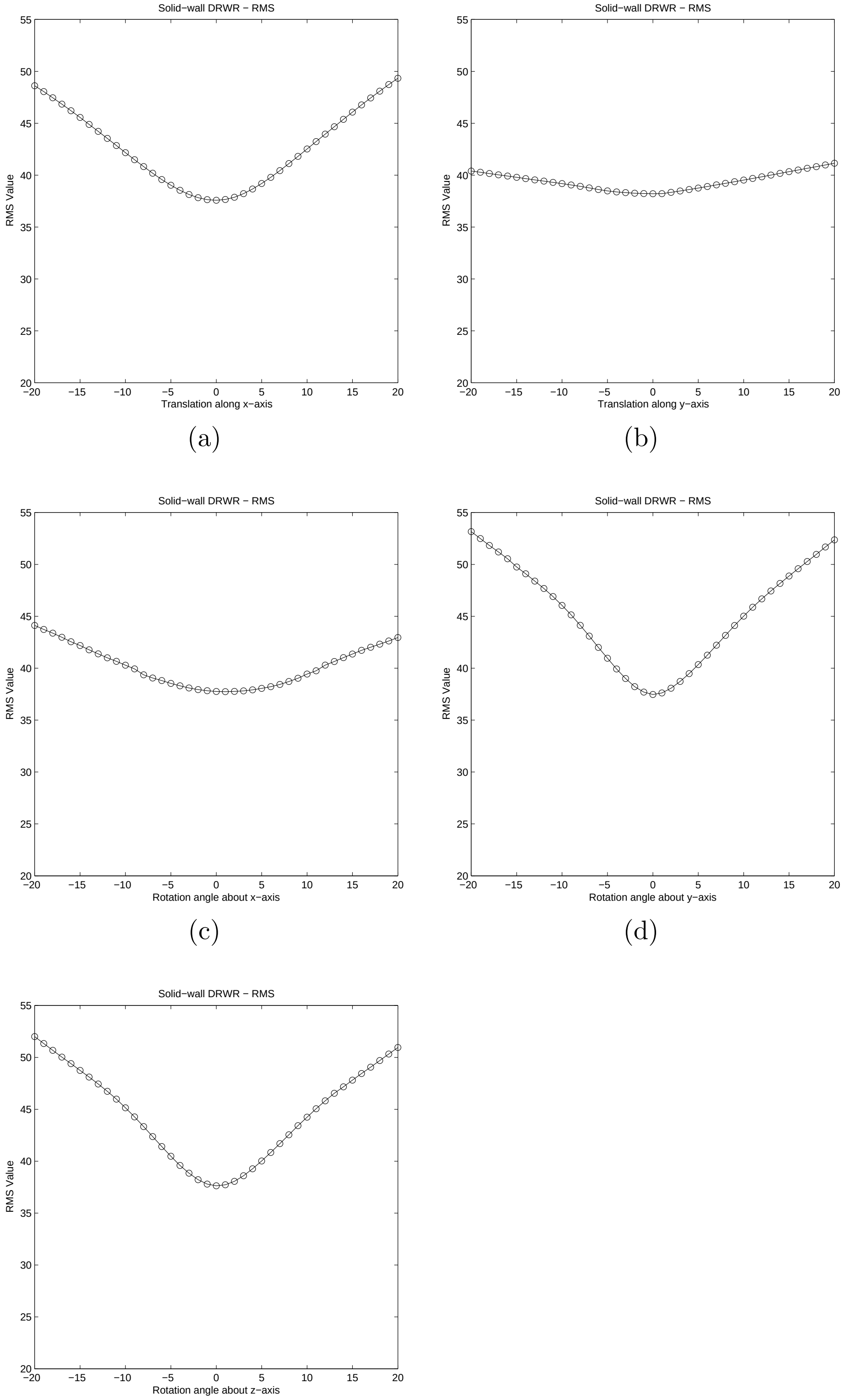
where the function $C(I_1, I_1, P(p))$ measures the variance of the control image within the neighbourhood $P(p)$.

$$C(I_i, I_j, P(p)) = \frac{1}{|P(p)|} \sum_{q\epsilon P(p)} I_i(q) I_j(q) - \frac{1}{|P(p)|^2} \sum_{q\epsilon P(p)} I_i(q) \sum_{q\epsilon P(p)} I_j(q) \tag{6.1.4}$$

Function $NC(I_0, I_1, P)$ calculates the normalized correlation coefficient between the region P in the two images.

$$NC(I_0, I_1, P) = \frac{\sum_{p\epsilon P} I_0(p) I_1(p) - \frac{1}{|P|} \sum_{p\epsilon P} I_0(p) \sum_{p\epsilon P} I_1(p)}{\sqrt{(\sum_{p\epsilon P} I_0(p)^2 - \frac{1}{|P|}(\sum_{p\epsilon P} I_0(p))^2)}(\sum_{p\epsilon P} I_1(p)^2 - \frac{1}{|P|}(\sum_{p\epsilon P} I_1(p))^2)} \tag{6.1.5}$$

From all three sets of graphs (Figures 6.3 - 6.5), it is clear that the registration will converge and there is a clear maxima or minima, despite using three very different functions to measure the degree of match. This shows the viability of using DRWRs for registration. Local maxima are observed in Figure 6.5(b), but the values are low compared to the global maxima, and can be handled with a robust optimization algorithm.



(a)

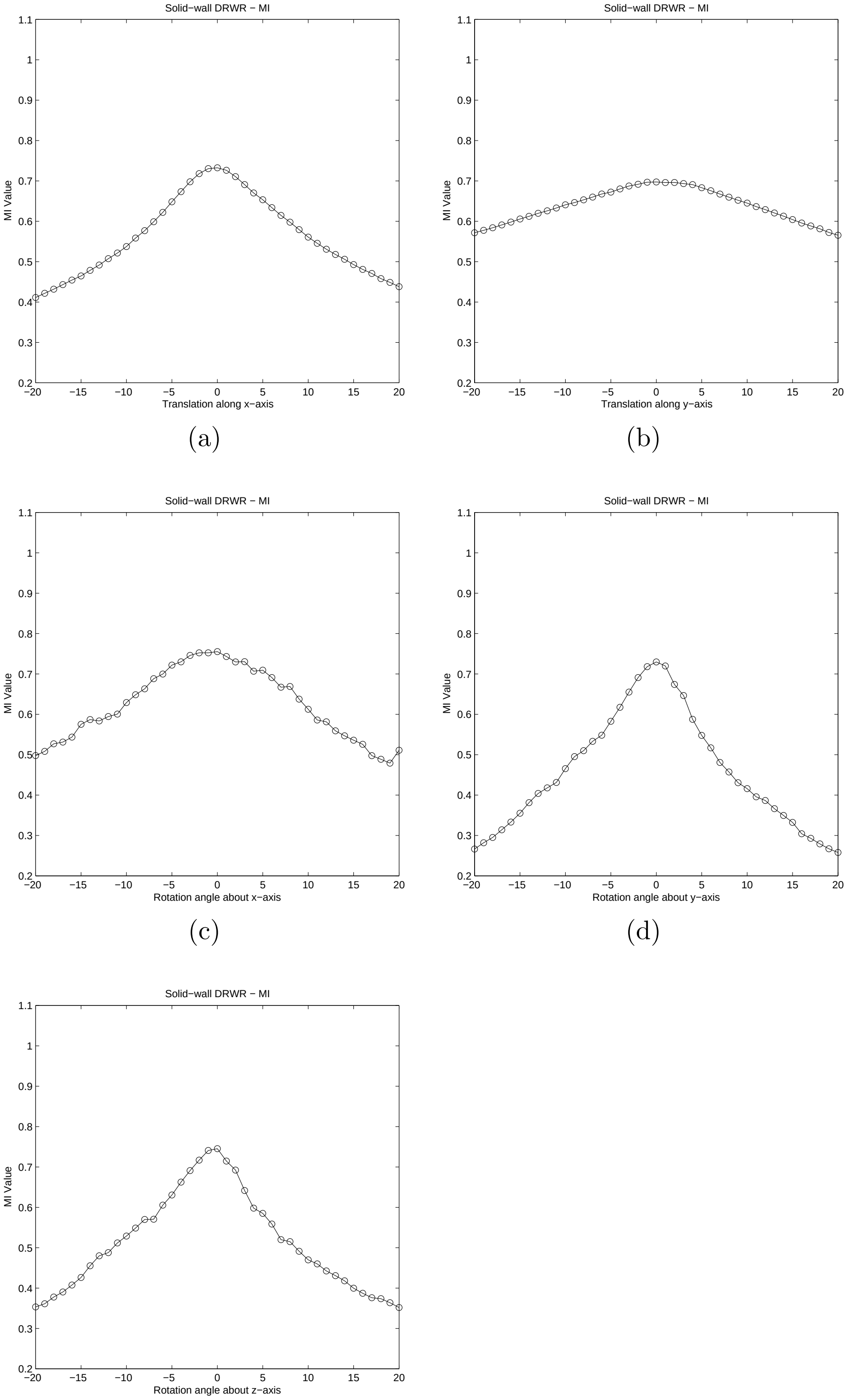
(b)

(c)

(d)

(e)

Figure 6.3: RMS degree of match curves. (a) Translation along x-axis. (b) Translation along y-axis. (c) Rotation about x-axis. (d) Rotation about y-axis. (e) Rotation about z-axis.



(a)

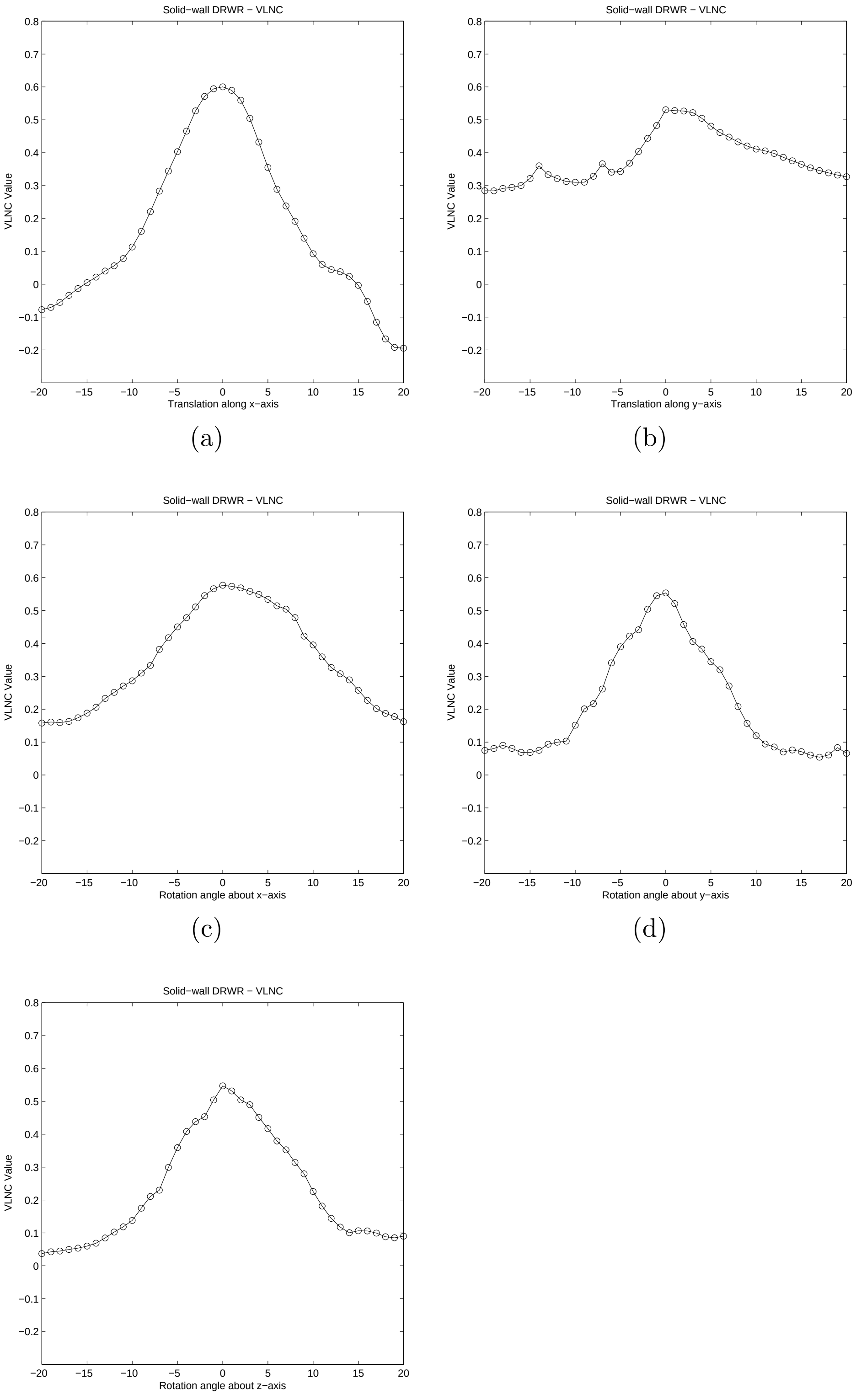
(b)

(c)

(d)

(e)

Figure 6.4: MI degree of match curves. (a) Translation along x-axis. (b) Translation along y-axis. (c) Rotation about x-axis. (d) Rotation about y-axis. (e) Rotation about z-axis.



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

Figure 6.5: VLNC degree of match curves. (a) Translation along x-axis. (b) Translation along y-axis. (c) Rotation about x-axis. (d) Rotation about y-axis. (e) Rotation about z-axis.

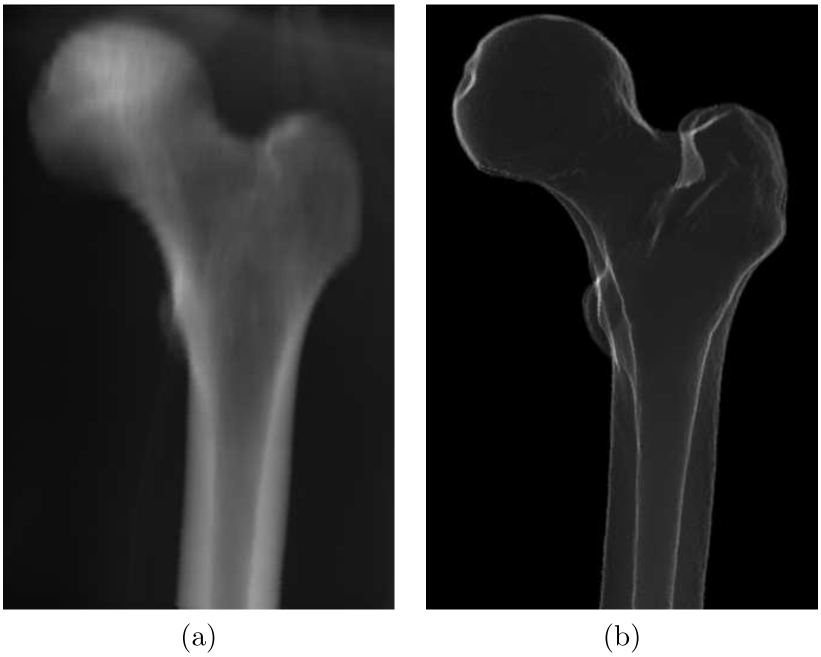
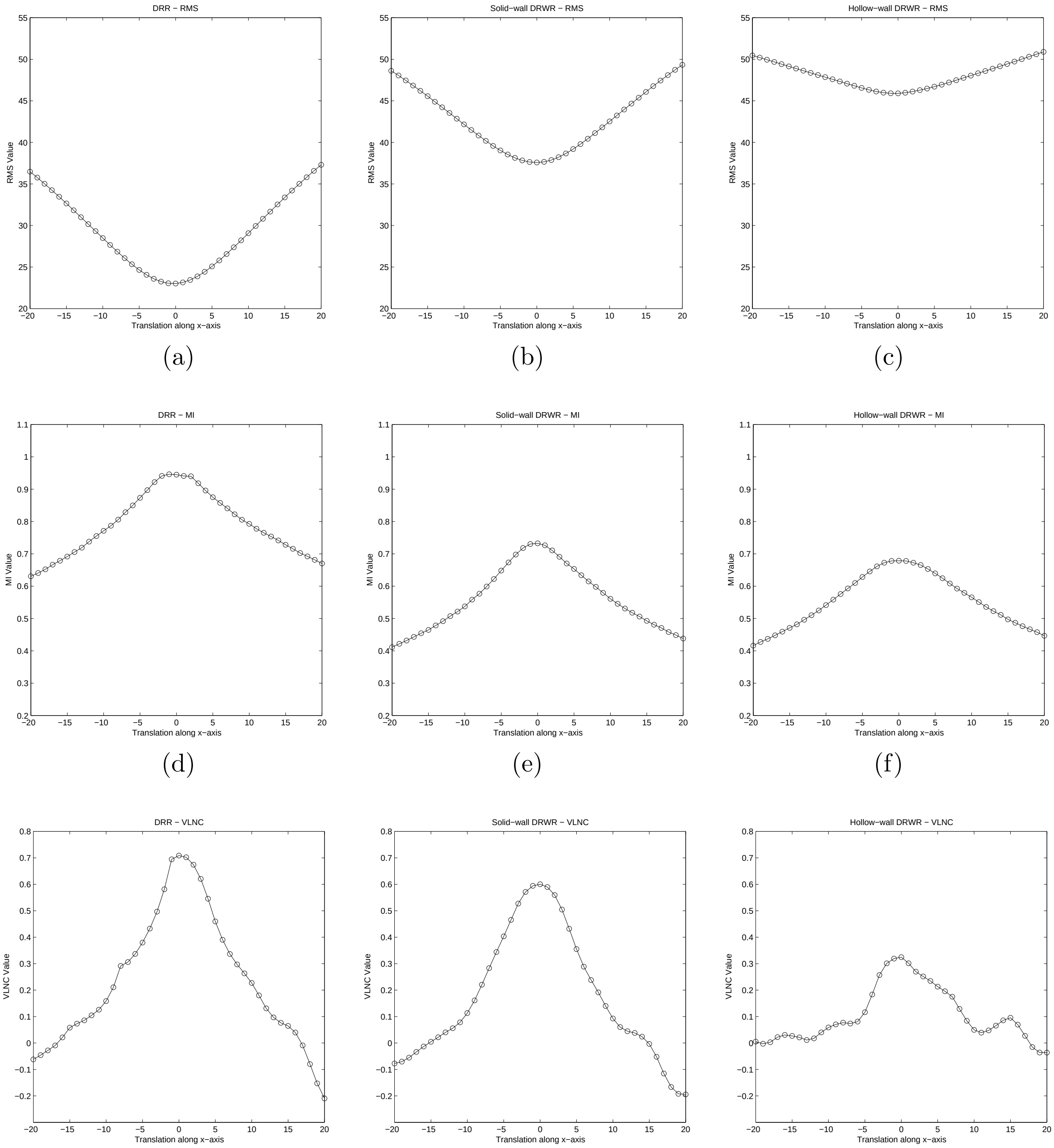


Figure 6.6: DRR and hollow-wall DRWR images of the dry femur. (a) Conventional DRR. (b)Hollow-wall DRWR.

6.2 Comparison of solid-wall DRWR, conventional DRR and hollow-wall DRWR

To make comparison among the solid-wall DRWR, the DRR and the hollow-wall DRWR, the DRR and the hollow-wall DRWR at the same pose are also rendered (Figure 6.6). Similarly, more images are taken at varying poses and the degree of match between each model and the x-ray image are measured using the three functions (Figures 6.7 - 6.11).

Of the three models, local maxima are observed in only one of the degree of match graphs of the solid-wall DRWR (Figure 6.8(h)) whereas local maxima are observed more often in these graphs of the other two models. This shows that the solid-wall DRWR is more robust than the conventional DRR and hollow-wall DRWR. Therefore, the solid-wall DRWR is the best performer. The hollow-wall DRWR performs below expectation despite being a variant of the solid-wall DRWR. However, it had already been suggested in Section 5.2 that the hollow-wall DRWR may be more suited to an edge-based similarity function.

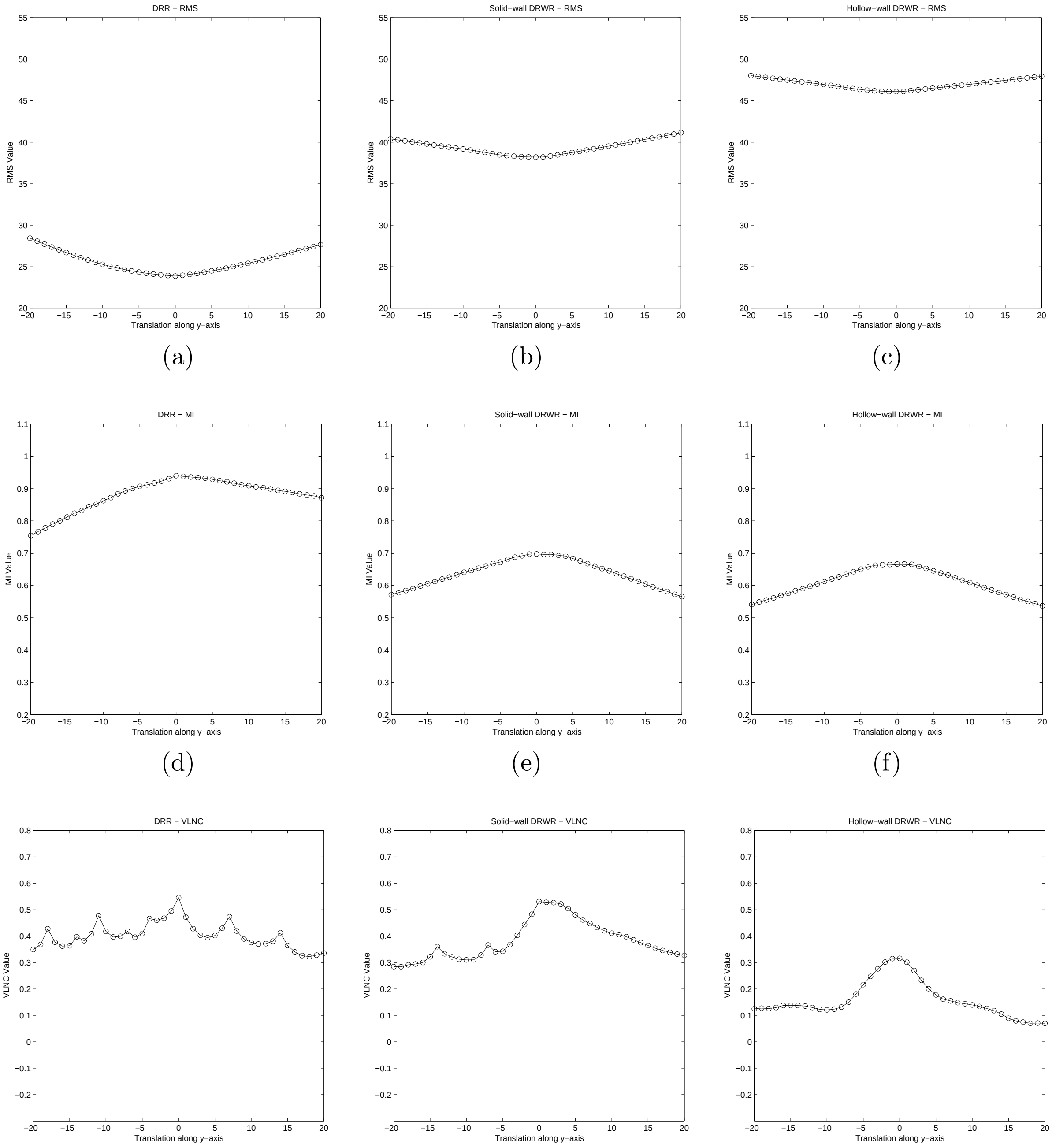


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 6.7: Comparison of the degree of match curves for translation along x-axis.

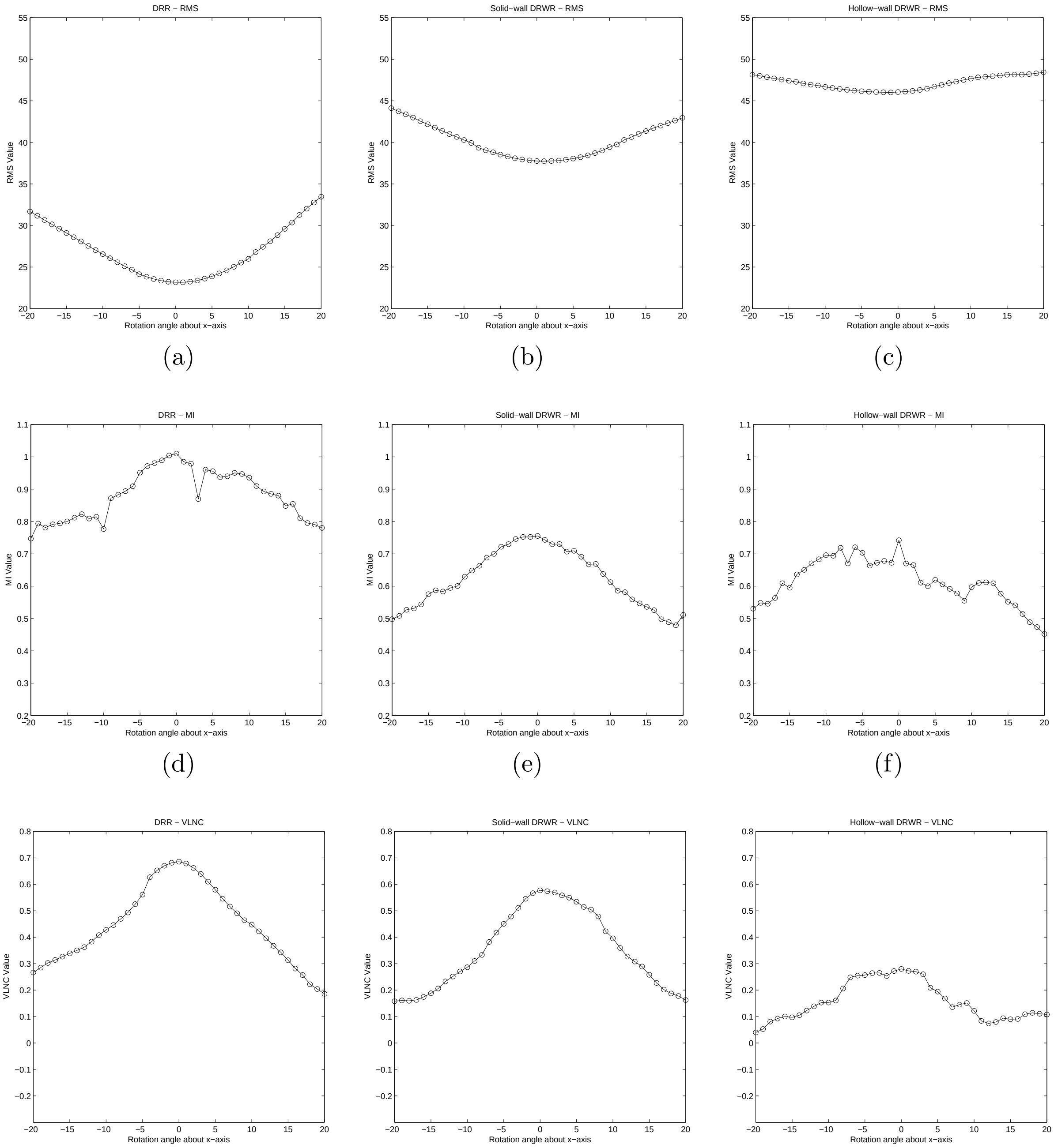


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 6.8: Comparison of the degree of match curves for translation along y-axis.

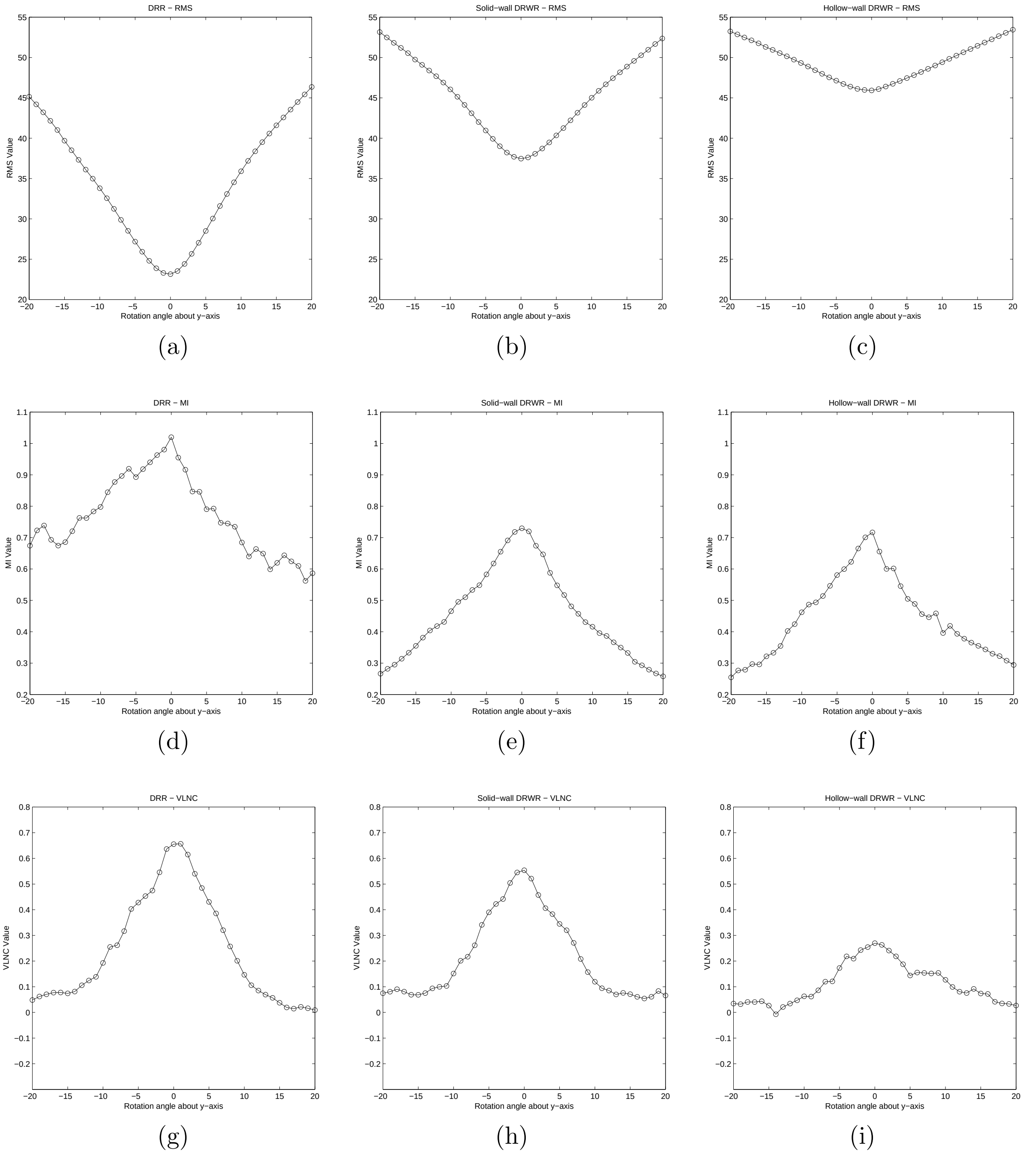


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 6.9: Comparison of the degree of match curves for rotation around x-axis.

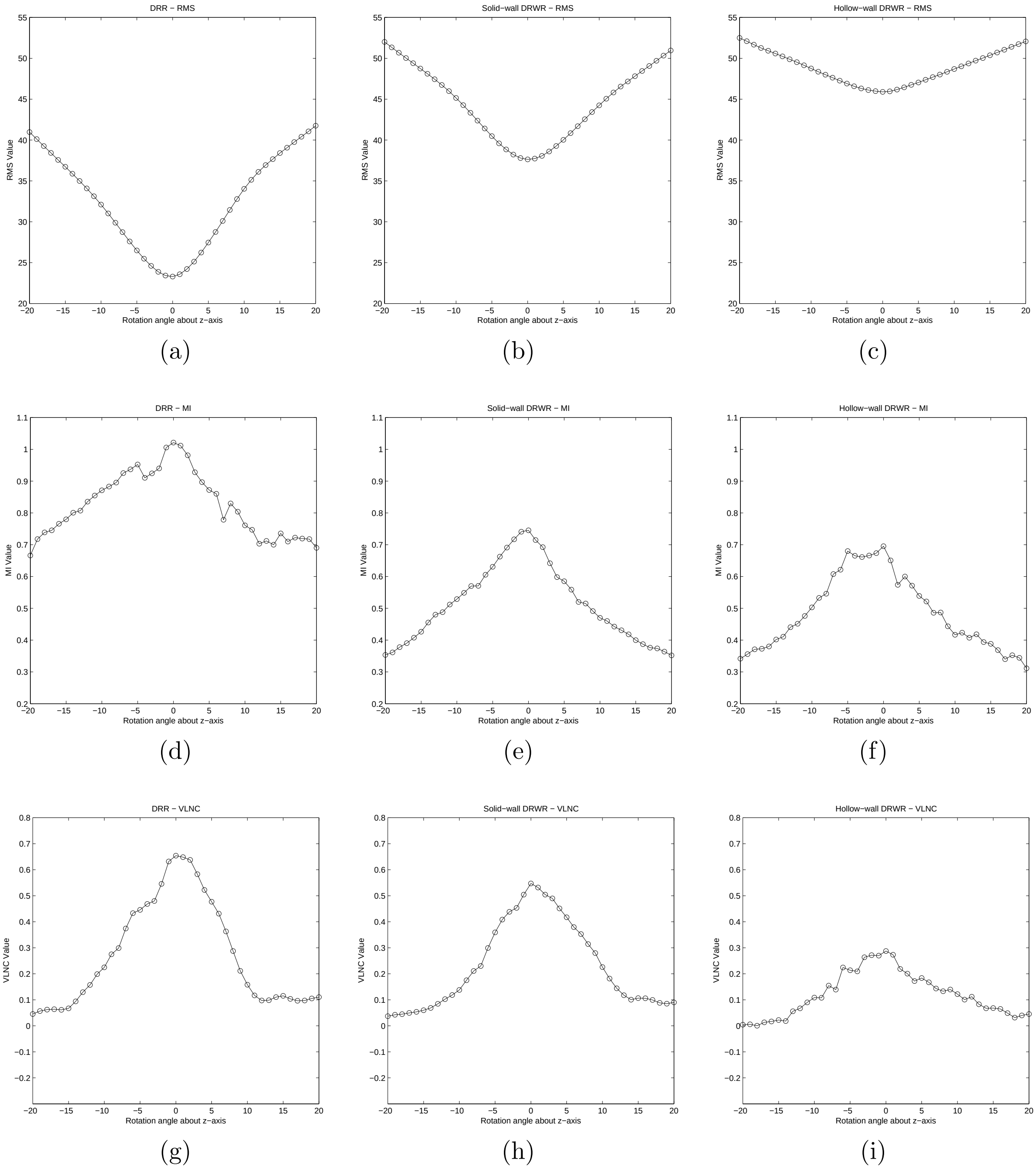


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 6.10: Comparison of the degree of match curves for rotation around y-axis.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 6.11: Comparison of the degree of match curves for rotation around z-axis.

Chapter 7

Conclusion

Registration is an essential component of CAS. In particular, 3D-2D registration is rapidly gaining in importance as more and more medical applications make use of it. However, the two approaches to 3D-2D registration, intensity-based and geometry-based approaches, have their shortcomings. A hybrid approach would be to combine the strengths of these two approaches while alleviating their weaknesses. The first step would be the design of a hybrid 3D model that incorporates both intensity and shape information. This is the DRWR, a hybrid 3D wall model that has been proposed for intensity-based registration to x-ray images in this thesis. The inner and outer wall surfaces capture the surface shape whereas the interior between the wall surfaces captures intensity information. Noises and irrelevant features such as the trabeculae are removed so that relevant features will not be obscured.

The 3D wall model is constructed using a 3D surface mesh model and CT volume. The generation of DRWR is done by simulating the attenuation of X-rays through the 3D wall model. Real-time rendering is made possible by hardware acceleration. When the DRWR of the wall model is synthesized, high-contrast features similar to edges and contours are produced only by the wall without the noise and trabeculae. Therefore, it produces clearer features and less noise than conventional DRR and intensity gradient maps. The relevant features are naturally highlighted without the need for feature extraction. The DRWR can be registered to an x-ray image or its intensity gradient image using intensity-based method without feature extraction. The results show that the DRWR is indeed a viable model for registration and can outperform the standard volumetric model in terms of robustness. This will translate to higher accuracy in registration, and thus improve the chances of success in medical procedures.

7.1 Future Work

Intensity-based similarity functions are used to evaluate the performance of these three models in this thesis. The next step would be to evaluate these models' performance using geometric-based functions. Another possible future work is to test the DRWR on real femur x-ray images. Concurrently, there is a need to derive a DRWR-specific similarity function which can be used to improve registration results.

Bibliography

[bac] Medical Research Council, Human Genetics Unit, Edinburgh UK, `http://genex.hgu.mrc.ac.uk/OPT_Microscopy/optwebsite/how_it_works/hiwtheory.htm`.

[BB96] L.M. Gottesfeld Brown and T.E. Boult. Registration of planar film radiographs with computed tomography. In *Proceedings of the 1996 Workshop on Mathematical Methods in Biomedical Image Analysis (MMBIA '96)*, page 4251, 1996.

[BKM00] Jacob Beutel, Harold L. Kundel, and Richard L. Van Metter. *Handbook of Medical Imaging, Volume 1. Physics and psychophysics*, volume PM79. SPIE Press Monograph, 2000.

[BLH07] P. K. Bhunre, W. K. Leow, and T. S. Howe. Recovery of 3D pose of bones in single 2D x-ray images. In *Proceedings of IEEE Workshop on Applications of Computer Vision*, page 48, 2007.

[BM92a] P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239256, 1992.

[BM92b] P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[BWFL] T.M. Buzug, J. Weese, C. Fassnacht, and C. Lorenz. Image registration: Convex weighting functions for histogram-based similarity measures.

[CAS] Smith & Nephew Orthopaedics, `http://www.smithnephew.com/uk/Standard.asp?NodeId=3396`.

[CBRS99] A. Czopf, C Brack, M. Roth, and A. Schweikard. 3D-2D registration of curved objects. *Periodica Polytechnica Series Electrical Engineering*, 43(1):1941, 1999.

[CCY$^+$03] H. M. Chan, A. C. S. Chung, S. C. H. Yu, A. Norbash, and W. M. Wells III. Multimodal image registration by minimizing kullback-leibler distance between expected and observed joint class histograms. *In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2:570576, 2003.

[Cha06] Si Wei Chan. GPU-based volume rendering of medical images. Honors Year Project Report, Department of Computer Science, School of Computing, National University of Singapore, 2006.

[CN88] M. R. Civanlar and R. A. Nobakht. Optimal pulse shape design using projections onto convex sets. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP) '88*, volume 3, pages 1874–1877, April 1988.

[Cro84] Franklin C. Crow. Summed-area tables for texture mapping. In *SIGGRAPH '84: Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, volume 18, pages 207–212, July 1984.

[CS98] Wenli Cai and Georgios Sakas. Maximum intensity projection using splatting in sheared object space. *Computer Graphics Forum (Eurogruphics'98)*, pages 113–124, 1998.

[CS00] Wenli Cai and Georgios Sakas. DRR volume rendering using splatting in shear-warp context. *In IEEE Nuclear Science Symposium Conference Record 2000*, 3:12–17, 2000.

[CSK03] Balázs Csébfalvi and László Szirmay-Kalos. Monte carlo volume rendering. In *Proceedings of the 14th IEEE Visualization Conference (VIS'03)*, pages 449–456, 19-24 October 2003.

[CTs] Pauls Valley General Hospital, `http://www.pvgh.net/patient.html`.

[cup] The ImPACT Group, http://www.impactscan.org/slides/impactcourse/artefacts/img7.html.

[CZ96] E. K. P. Chong and S. H. Zak. *An Introduction to Optimization*. John Wiley and Sons Inc., New York, 1996.

[CZ05] T.F. Chan and W. Zhu. Level set based shape prior segmentation. *Computer Vision and Pattern Recognition*, 2:1164–1170, 2005.

[DH92] John Danskin and Pat Hanrahan. Fast algorithms for volume ray tracing. *In 1992 Workshop on Volume Visualization, Boston, MA*, pages 91–98, October 1992.

[DNR90] S. Dunne, S. Napel, and B. Rutt. Fast reprojection of volume data. In *Proceedings of the 1st Conference on Visualization in Biomedical Computing 1990*, pages 11–18, 1990.

[FAB97] J. Feldmar, N. Ayache, and F. Betting. 3D-2D projective registration of free-form curves and surface. *Computer Vision and Image Understanding*, 65(3):403424, 1997.

[FHR04] E. Freund, F. Heinze, and J. Rossmann. Direction dependent projection fields for the fast DRR generation for medical 2D/3D registration. In *Proceedings of the 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS), San Francisco, CA, USA*, volume 3, pages 1751–1754, September 2004.

[GGSC] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The lumigraph. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques.*

[GKBT98] A. Guéziec, P. Kazanzides, B.Williamson, and R. H. Taylor. Anatomy-based registration of CT-scan and intraoperative x-ray images for guiding a surgical robot. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):715–728, 1998.

[GLDH97] M.H. Gross, L. Lippert, R. Dittrich, and S. Häring. Two methods for wavelet-based volume rendering. *Computer and Graphics*, 21(2):237–252, March/April 1997.

[KWT87] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1:321–331, 1987.

[LaR01] D. A. LaRose. *Interactive X-ray/CT Registration Using Accelerated Volume Rendering*. PhD thesis, Dept. Electrical and Computer Engineering, Carnegie Melon University, U.S.A, May 2001.

[LCJ$^+$00] D. LaRose, L. Cassenti, B. Jaramaz, J. Moody, T. Kanade, and A. DiGioia. Post-operative measurement of acetabular cup position using x-ray/CT registration. In *Proceedings of Medical Image Computing and Computer-Assisted Intervention*, pages 1104–1113, 2000.

[Lev88] Marc Levoy. Display of surfaces from volume data. *IEEE Computer graphics and Applications*, 8(3):29–37, May 1988.

[Lev90] Marc Levoy. Efficient ray tracing of volume data. *ACM Transactions on Graphics*, 9(3):245–261, July 1990.

[LG95] L. Lippert and M. H. Gross. Fast wavelet based volume rendering by accumulation of transparent texture maps. *Computer Graphics Forum*, 14(3):431–443, 1995.

[LH] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd annual conference on Computer Graphics and Interactive Techniques*, volume 30.

[LH91] David Laur and Pat Hanrahan. Hierarchical splatting: A progressive refinement algorithm for volume rendering. In *Proceedings of ACM SIGGRAPH '91, Computer Graphics*, volume 25, pages 285–288, July 1991.

[LJF$^+$94] L. Lemieux, R. Jagoe, D.R. Fish, N.D. Kitchen, and D.G.T. Thomas. A patient-to-computed-tomography image registration method based on digitally reconstructed radiographs. *Medical Physics*, 21(11):1749–1760, 1994.

[LL94] Philippe Lacroute and Marc Levoy. Fast volume rendering using a shear-warp factorization of the viewing transformation. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 451–458, New York, NY, USA, July 1994. ACM Press.

[Low91] D. G. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):441450, 1991.

[LS95] S. Lavallée and R. Szeliski. Recovering the position and orientation of free-form object from image contours using 3D distance maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(4):378–390, 1995.

[LYJ03] H. Livyatan, Z. Yaniv, and L. Joskowicz. Gradient-based 2D/3D rigid registration of fluoroscopic x-ray to CT. *IEEE Transactions on Medical Imaging*, 22(11):1395–1406, 2003.

[Mal93] Tom Malzbender. Fourier volume rendering. *ACM Transactions on Graphics*, 12(3):233–250, July 1993.

[MCV$^+$97] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Seutens. Multimodality image registration by maximization of mutual information. *IEEE Transactions on Medical Imaging*, 16(2):187–198, 1997.

[MY96] Klaus Mueller and Roni Yagel. Fast perspective volume rendering with splatting by utilizing a ray-driven approach. In *VIS '96: Proceedings of the 7th conference on Visualization '96*, pages 65–72, Los Alamitos, CA, USA, 1996. IEEE Computer Society Press.

[NCSN99] Efthymios Ntasis, Wenli Cai, Georgios Sakas, and Konstantina S. Nikita. Real time digital reconstructed radiograph (DRR) rendering in frequency domain. In *Proceedings of the First Joint BMES/EMBS Conference*, volume 2, page 1041, October 1999.

[PMV03] Josien P. W. Pluim, J. B. Antoine Maintz, and Max A. Viergever. Mutual information based registration of medical images: a survey. *IEEE Transactions on Medical Imaging*, XX(Y), 2003.

[PWL$^+$98] Graeme P Penney, Jürgen Weese, John A Little, Paul Desmedt, Derek LG Hill, and David J Hawkes. A comparison of similarity measures for use in 2D-3D medical image registration. *Medical Image Computing and Computer-Assisted Intervention (MICCAI) '98, LNCS 1496*, page 11531161, 1998.

[RRD$^+$04] Torsten Rohlfing, Daniel B. Russakoff, Joachim Denzler, Calvin R. Maurer, and Jr. Progressive attenuation fields: Fast 2D-3D image registration without precomputation, 2004.

[RRM$^+$05] Daniel B. Russakoff, Torsten Rohlfing, Kensaku Mori, Daniel Rueckert, Anthony Ho, John R. Adler Jr., and Calvin R. Maurer Jr. Fast generation of digitally reconstructed radiographs using attenuation fields with application to 2D-3D image registration. *IEEE Transactions on Medical Imaging*, 24(11):1441–1454, November 2005.

[Set96] J. A. Sethian. *Level Set Methods*. Cambridge University Press, 1996.

[SF90] K. R. Subramanian and Donald S. Fussell. Applying space subdivision techniques to volume rendering. In *Proceedings of the IEEE Visualization*, pages 150–159, October 1990.

[SFE00] Mikkel B. Stegmann, R. Fisker, and Bjarne K. Ersbøll. On properties of active shape models. Technical report, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, 2000.

[SHH96] C. Studholme, D.L.G. Hill, and D.J. Hawkes. Automated 3D registration of MR and CT images of the head. *Medical Image Analysis*, 1(2):163–175, 1996.

[SHH97] C. Studholme, D.L.G. Hill, and D.J. Hawkes. Automated 3D registration of MR and PET brain images by multi-resolution optimisation of voxel similarity measures. *Medical Physics*, 24:25–35, 1997.

[sim] Institute of Information and Computing Sciences, Utrecht University, Netherlands, `http://www.cs.uu.nl/groups/AA/virtual/surgery/`.

[SK00] Milos Sramek and Arie Kaufman. Fast ray-tracing of rectilinear volume data using distance transforms. *IEEE Transactions on Visualization and Computer Graphics*, 6(3):236–252, July-September 2000.

[TCL$^+$] T. P. Tian, Y. Chen, W. K. Leow, W. Hsu, T. S. Howe, and M. A. Png. Computing neck-shaft angle of femur for x-ray fracture detection. In *Proceedings of International Conference on Computer Analysis of Images and Patterns (LNCS 2756)*.

[TLP02] D. Tomaževič, B. Likar, and F. Pernuš. Rigid 2D/3D registration of intraoperative digital x-ray images and preoperative CT and MR images. In *Proceedings of SPIE Medical Imaging Conf.*, 2002.

[vdKPT$^+$05] E. B. van de Kraats, G. P. Penney, D. Tomaževič, T. van Walsum, and W. J. Niessen. Standardized evaluation methodology for 2D3D registration. *IEEE Transactions on Medical Imaging*, 24(9):1177–1189, 2005.

[VW95] P. Viola and W. M. Wells. Alignment by maximization of mutual information. In *Proceedings of 5th International Conference on Computer Vision (ICCV95)*, page 1623, 1995.

[WBLF97] J. Weese, T.M. Buzug, C. Lorenz, and C. Fassnacht. An approach to 2D/3D registration of a vertebra in 2D x-ray fluoroscopies with 3D CT images. In *Proceedings of the First Joint Conference on Computer Vision, Virtual Reality and Robotics in Medicine and Medial Robotics and Computer-Assisted Surgery*, pages 119–128, 1997.

[Wes90] Lee Westover. Footprint evaluation for volume rendering. In *Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, volume 24, pages 367–376, August 1990.

[WR00] Michel A. Westenberg and Jos B. T. M. Roerdink. Frequency domain volume rendering by the wavelet x-ray transform. *IEEE Transactions on Image Processing*, 9(7):1249–1261, July 2000.

[XP98] C. Xu and J. L. Prince. Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, 7(3):359–369, 1998.

[ZaAHKV92] Zuiderveld, Karel J. ad Anton H.J. Koning, and Max A. Viergever. Acceleration of ray-casting using 3D distance transforms. In *Proceedings of Visualization in Biomedical Computing 1992*, pages 324–335, October 1992.

[ZGW01] L. Zöllei, E. Grimson, and W. Wells. 2D-3D rigid registration of x-ray fluoroscopy and ct images using mutual information and sparsely sampled histogram estimators. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 696–703, 2001.