Rigid Registration of MR and Biplanar Fluoroscopy

by

Eric Cosman

Submitted to the

Department of Electrical Engineering and Computer Science

in Partial Fulfillment of the Requirements for the Degree of

Masters of Engineering in Electrical Engineering and Computer Science

at the

Massachusetts Institute of Technology

June 2000

Author . . . .
Department of Electrical Engineering and Computer Science
May 22, 2000

Certified by . . . .
William Wells III
Thesis Supervisor

Accepted by . . . .
Arthur C. Smith
Chairman, Department Committee on Graduate Theses

Rigid Registration of MR and Biplanar Fluoroscopy

by

Eric Cosman

Submitted to the
Department of Electrical Engineering and Computer Science

May 22, 2000

In Partial Fulfillment of the Requirements for the Degree of
Masters of Engineering in Electrical Engineering and Computer Science

Abstract

This thesis discusses the the rigid registration of MR data and biplanar, fluoroscpic radiographs of the head, by means of an intensity-based measure of similarity. We investigate a number of similarity measures which have been applied in other multimodal registration contexts, including that of CT-fluoroscopy registration. The similarity measures are evaluated by probing the space of transformations between the coordinate frames of an MR volume and biplanar, digitally-reconstructed radiographs (DRRs) produced from a corresponding CT volume. This method allowed us to know the "ground truth" of the registration, which could be established by proven methods for the 3D-3D CT-MR alignment.

Futhermore, we propose a method of DRR production called *voxel-projection* which drastically reduces processing time relative even to optimized *ray-casting* methods. The computational efficiency of *voxel-projection* makes it a useful tool for the investigation of similarity measures in this 3D-2D context. Its speed may also enable 3D-2D registration methods which rely on evaluation of a similarity measure over the entirety of a DRR and model radiograph.

Based on similarity measure chacteristics observed in our probing experiments, and using the *voxel-projection* method, we adapted the *uphill-simplex* optimization algorithm to implement an intensity-based MR-fluoroscopy rigid registatration engine.

Thesis Supervisor: William Wells III
Title: Research Scientist, MIT AI Lab

Acknowledgments

First and foremost, thank you Sandy for advising my thesis. Your comments and suggestions consistently clarified situations and pointed me to new directions of inquiry. At the same time, you gave me room to explore and to make discoveries on my own, in the way that I find most personally productive. Indeed, you contributed not only to my intellectual development, but also to my athletic betterment by introducing me to my new favorite sport, Hockey. This, I believe, is a unique impact for a thesis advisor to have, and I certainly appreciate it.

Thank you also to Dr. Norbash for providing energetic clinical support to the project. I cannot imagine doing research in the field of medicine without the guidance from those who practice it. The insights you provided into the clincial context for the work of this thesis was integral to its motivation and success.

And, finally, thank you to Tina Kapur. Your enthusiasm for this field was a great motivator to me, and your friendly interest and support was not only a nice surprise, but also one which proved very helpful.

This research is supported by the National Science Foundation Engineering Research Center contract #8810-274 administered through Johns Hopkins University.

Contents

1 Introduction 13

1.1 Multimodal Medical Image Registration 13

1.2 Registering MR Images to Biplanar, Fluoroscopic Radiographs 14

1.3 Procedures 16

1.4 Overview of the Thesis 17

2 Technical Issues and Background 19

2.1 Fluoroscopic Image Formation 19

2.1.1 Distortions 20

2.1.2 Differentiation of Tissue Types 21

2.2 MR 21

2.3 CT 23

2.4 Data Representations 24

2.4.1 Organization in Memory 25

2.4.2 Data Size Relative to Cache Sizes 25

2.4.3 Relation of Metric and Data Coordinates 26

2.4.4 Interpolation for Inter-Sample Values 27

2.5 Summary 29

3 Image Registration 31

3.1 Problem Definition 31

3.2 Transformation Parameters 32

3.3 Medical Image Registration Methodologies 32

3.3.1 Alignment of Fiducials 32
3.3.2 Feature Alignment 33
3.3.3 Intensity-based Similarity Measure 33
3.4 Intensity-based Image Registration 34
3.4.1 Matching Measure to Optimization Method 34
3.4.2 Optimization Schemes 35
3.5 Intensity-based, 3D-2D Image Registration 36
3.6 Reconstructing Radiographs 37
3.6.1 Comparing DRRs to Real Radiographs 38
3.6.2 Objective Functions 39
3.7 Summary 43

4 DRR-Production Algorithms **45**
4.1 Introduction 45
4.1.1 An Efficient Coordinate Transform 46
4.2 Ray-casting 47
4.2.1 Ray-casting Parameters 48
4.2.2 The Ray-casting Algorithm 49
4.2.3 Resolution of Ray-cast DRRs 50
4.2.4 Complexity of Ray-casting 51
4.2.5 Memory Access Patterns of Ray-casting 51
4.3 Voxel-projection 52
4.3.1 Development of the Method 53
4.3.2 Variations Among Voxels' Projective Images 58
4.3.3 2D Rectangular Pulse 59
4.3.4 2D Convolution of Orthogonal 1D Triangles (COT) 60
4.3.5 Voxel-projection Parameters 64
4.3.6 The Voxel-projection Algorithm 64
4.3.7 Voxel-projection Complexity 70
4.3.8 Voxel-projection Resolution 71

4.4 Running-time Comparisons 71

4.5 Summary 72

5 Probing Objective Functions 75

5.1 Overview 75

5.2 Input Data 76

5.3 Objective Function Evaluation 76

5.4 DRR production 77

5.5 Pertinent Objective Function Qualities 78

5.6 Pertinent Objective Function Qualities 79

5.6.1 Shape 79

5.6.2 Behavior at Different Resolutions 79

5.7 Transformation Parameterization 80

5.8 Range of Probing 81

5.9 Results and Discussion 81

5.9.1 Directions of Misregistration 82

5.9.2 Objective Functions 82

5.9.3 1D Probing at Full-Resolution 82

5.9.4 1D Probing at Lower Resolution 83

5.9.5 2D Probing at Full-Resolution 85

5.9.6 Discussion 88

6 Uphill-Simplex Registration Algorithm 91

6.1 Overview 91

6.2 Uphill-Simplex Function Optimizaiton 92

6.3 Transformation Parameterization 93

6.3.1 A Well-formed Transformation Space 94

6.3.2 A Euclidean Rotation Space 95

6.4 Multi-resolution Schedule 98

6.5 Summary 99

7 Conclusion 101

7.1 Related and Future Work 102

7.1.1 Voxel-projection 102

7.1.2 Objective Function Evaluation 102

7.1.3 MR Intensity Modification 103

A Memory Hierarchy and Data Caching 105

List of Figures

2-1 Fluoroscopic Image Formation 20

2-2 MR Image 22

2-3 CT Image 24

2-4 Volumetric Data in Memory 25

2-5 Framework for Trilinear Interpolation 28

3-1 Registering volumetric data with biplane fluoroscopy 36

4-1 An affine data coordinate transform 47

4-2 Ray-casting DRR Production 48

4-3 A Voxel's Projective Image (Nearest-Neighbor Interpolation) 53

4-4 Unsmoothed Voxel Projection 55

4-5 Smoothed Voxel Projection 56

4-6 A Voxel's Span of Influence on the DRR 57

4-7 Variation in the extent of voxel projections. 59

4-8 COT Kernel 60

4-9 Similarity of COT kernel and Trilinear Interpolation 62

4-10 Divergence of COT and trilinear interpolation 63

4-11 Oversizing the COT Kernel 63

4-12 Convolution by a 2D Kernel 68

4-13 Applying a Triangular Kernel 69

4-14 Triangular Kernel 70

4-15 Double-sweeping a rectangle to produce a trilangle 70

4-16 Running time comparision of voxel-projection and ray-casting over a series of DRR and volume sizes. 72
4-17 Voxel-projection running times across DRR and volume sizes. 73

5-1 Sagittal CT-DRR produced by ray-casting with trilinear interpolation 77
5-2 Coronal CT-DRR produced by ray-casting with trilinear interpolation 78
5-3 Sagittal MR-DRR produced by voxel-projection 78
5-4 Coronal MR-DRR produced by voxel-projection 79
5-5 Probing Directions Defined 82
5-6 Probing a wide range of x and z-axis misalignments 83
5-7 Details of peak over z-axis misalignments 84
5-8 Detail of objective function fluctuations 84
5-9 Probing over rotations about the x-axis (roll) 84
5-10 Probing over rotations about the y-axis (pitch) 85
5-11 Probing a wide range of translational offsets for low-res model 85
5-12 Probing a narrow range of translational offsets for low-res model . . . 86
5-13 Probing over rotational offsets for low-res model 86
5-14 Pattern Intensity, yz-plane probing 87
5-15 Pattern Intensity, y-axis and roll angle probing 87
5-16 Pattern Intensity, pitch and yaw angle probing 88

Chapter 1

Introduction

1.1 Multimodal Medical Image Registration

The variety of medical imaging techniques currently available offers a wide range of functionalities which are relevant to medical diagnosis, monitoring, surgical treatment planning, and surgical guidance. These functionalities are complementray across different modalities which convey different types of information. For instance, some imaging techniques give real-time information on changes in structure or function, but are limited in resolution or the manner of visualization they can afford. Whereas other techniques offer great sophistication of visualization and feature resolution, but require longer imaging and processing times. For this reason, many modern procedures involve the use of multiple imaging techniques. Very often, however, the process of using multiple images of differing modalities is a difficult one: it may be hard for a doctor to fully process and quantitatively compare the disparate data, or such a comparison may require tedious or error-prone measurement schemes. As a result, significant effort has been directed toward finding computational solutions to this problem of integrating differing images. An essential part of this integration is the *registration*, or spatial alignment, of these images. A significant amount of work has been done to automate this process for a variety of imaging modalities because of improvements in visualization and guidance that such multi-modal registration enables. Furthermore, as medicine continues to pursue less invasive alternatives to traditional surgical therapies, the impetus to bring multiple imaging techniques to bear in concert will be sustained.

1.2 Registering MR Images to Biplanar, Fluoroscopic Radiographs

In this thesis, we are interested in the problem of rigidly registering volumetric data from *Magnetic Resonance Imaging* (MR) to 2D, fluoroscopic images, and we work towards an algorithm by which this may be performed automatically. This kind of registration promises to have broad application in improving and facilitating guidance in a variety of interventional contexts. To understand these applications, we need to explore the complementary functionalities of these imaging modalities. We give an overview of this functionalities here, and defer discussion of the imagining processes and other data characteristics until chapter 2.

Biplane fluoroscopy is an x-ray imaging modality that is widely used in interventional medicine because it simultaneously provides two, often orthogonal, radiographic views of targeted structures in real time. This technique has a number of properties which make it very well suited for intraoperative monitoring of certain interventions.

- Acquisition is fast.
- Software methods exist to correct for geometric distortions introduced by the imaging apparatus, which can be characterized by imaging a reference phantom[26].
- Bony structures are well imaged by radiography in general, so fluoroscopy can provide updates on the intraoperative orientation internal bodies structurally-associated with bones.
- With two radiographic views, the position of interventional instruments can be located relative to the imaging apparatus and, thus, relative to any internal structures whose intraoperative position can also be so determined.

Unfortunately, fluoroscopy provides limited and ambiguous information about internal structures in general, due to the fact that it does not well resolve soft tissue without the introduction of contrast agents. Furthermore, volumetric information is drastically reduced by the radiographic process which "compresses" that 3D information into a 2D perspective projection.

On the other hand, MR is used widely for diagnostic purposes and in stereotactic guidance for the following reasons:

- MR data is volumetric with millimeter (in-slice) linear resolution, and can be processed to allow for sophisticated visualization (fly-through, zoom, and cutaway views) of internal structures [4].
- MR can differentiate many tissue types in a way that allows for identification and localization of a broad range of internal structures and pathologies.

However, acquisition and processing times for MR are generally long, and MR scanners generally obstruct access to the patient, so that imaging for intraoperative guidance is limited.

While these modalities individually have their strengths and weaknessnes, when we consider their qualities in union, we see that MR and biplane fluoroscopy together offer a more complete joint functionality for surgical guidance. Indeed, there are a number of therapies to which we might find application for the fusion of preoperatively-processed MR and intraoperatively-acquired fluoroscopy. Examples would be therapies that involve access to soft tissue targets, such as tumors, by their relation to bony structures nearby. Such therapies are suited to improvement by MR-fluoroscopy registration because MR can define preoperatively and in 3D the relationship of the soft tissue to bony structures with confidence, and the fluoroscopy can give accurate information about the state of the bony structures (and, by inference, the soft tissue) during the procedure. Structures in the head and spine are well suited to this technique because these areas are particularly critical, and surgeries performed in them often require superior guidance which, by current methods, require complicated measurement and reference schemes, use of fiducials, or near-continuous use of fluoroscopy (which not infrequently results in radiation-related injuries such as hair loss).

1.3 Procedures

Here we present a variety of therapies which might benefit from the fusion of MR to fluoroscopy. These descriptions are the product of correspondence with Dr. Alexander Norbash, Head of Neurointerventional Radiology at the Brigham and Women's Hospital in Boston, MA.

Pallidotomy The fluoroscopically-guided Pallidotomy is a prototypical example of a stereotactic neurosurgical procedure performed with the intraoperative aid of fluoroscopy. In this procedure, a small-diameter probe is passed through the brain, and its tip is placed in a very precise location where a small controlled injury results in interruption of the bundles that cause the symptoms found in Parkinsonism. This stereotaxy is currently performed using a cumbersome series of linear measurements on preprocedural MR data that relate the target location to a reference region in the ventricular system. The the validity of these measurements relies on the rigidity of calvarial structures, so that structures will not have shifted between the times of MR and fluoroscopic image aquistions.

The reference can be confidently visualized intraprocedurally using ventriculography, whereby it is temporarily made radiopaque with the introduction of contrast; however, the complicated coordination of MR and fluoroscopy that defines the current approach introduces a degree of error into the stereotaxy, which might be improved by an accurate, automatic image registration method.

Resection of Brain Tumors An outstanding problem in neurosurgery is the real-time update of residual volumes as a tumor is being resected. Present method of ultrasound, MR, and CT updating are indefinite and cumbersome. Preoperative MR images can define tumor volume accurately and in relationship to bony lankmarks or possibly implanted radiographic markers. Use of fluoroscopy during surgery would provide the surgeon with a real-time update of the course of his surgical apprach relative to the bony or marker environment. This MR-fluoroscopy fusion could provide him with a valuble real-time update of tumor margin relative to resection boarder.

Spinal Surgery A large nubmer of disc decompressions and spinal tumor resections are performed each year. These rely primarily on the identification of the correct disc level by intraoperative fluoroscopy. Preopertive MR images, however, can clearly reveal in 3D, the position of disc, neural abnormalites, and spinal, soft-tissue tumors. By MR-fluoroscopic fusion, the position of such critical target volumes can be visualized on the 2D fluoroscopic views, enabling the removal of a precise amount of the disc to decrease neural impingement, or the complete and safe removal of spinal tumors.

1.4 Overview of the Thesis

In Chapter 2, we present background on relevant imaging modalities, and technical information on how medical image data is stored in memory. In Chapter 3, we define the problem of registration in general and specify it to the MR-fluoroscopy context. In Chapter 4, we present algorithms for simulating radiographic imaging on volumetric data, focusing on a new method *voxel-projection* which is significantly faster than more straightforward approaches. In Chapter 5, we present probing experiments which evaluate a number of objective functions on the basis of utility in an intensity-based MR-fluoroscopy registration algorithm. In Chapter 6, we adapt the *Uphill-simplex* optimization scheme for an automatic 3D-2D registration method. Finally, in Chapter 7, we discuss the thesis and its extensions into future or related work.

Chapter 2

Technical Issues and Background

In this chapter, we detail the characteristics and functionality of MR and fluoroscopic imaging. We also discuss CT imaging, as it is relevant to the justification of our investigations in this thesis. Understanding these characteristics not only provide background information on how these imaging techniques can be brought to bear clinically, but they are a primary consideration in approaching the problem of automatically registering medical images of disparate modalities. Clearly, to automate the registration of two images, we must understand what information they contain which might relate them, and what are the limitations of the data that might complicate the process.

We also discuss how medical datasets are stored digitally. This will be relevant to the description and efficient implementation of the algorithms our method employs.

2.1 Fluoroscopic Image Formation

In this section we describe the process by which fluoroscopic images are formed[23]. Refer to Figure 2-1 during this discussion for a diagram illustrating this process and the fluroscopic apparatus. Fluoroscopic images are radiographs that are produced by the interaction of x-ray photons with a planar photon detector called an *image intensifier*. X-rays are emitted radially from a small-diameter, x-ray tube anode, located at a distance, the *focal length*, from the image intensifier which measures

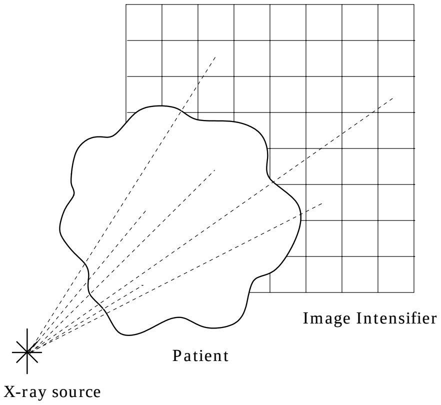

Figure 2-1: Fluoroscopic Image Formation

the intensity of rays incident on it. The imaging process that occurs when a body is placed between the x-ray source and the image-intensifier is based on variations in the attenuating properties of different types of tissues. The intensity of an x-ray emitted from the x-ray source and terminating at point (x, y) of the image intensifier is ideally governed by the following expression

$$I(x, y) \propto \exp(-\int_{ray} \mu(x, y, z) dr) \tag{2.1}$$

where $\mu(x, y, z)$ is the *linear attenuation coefficient* of the tissue located at (x, y, z) in space. The attenuation coefficient is primarily a function of tissue density, component atomic number, and x-ray energy. Due to the above relationship whereby attenuation coefficients are summed along the rays traveled by emitted x-ray photons, a radiograph is ideally a function of the perspective projection of tissue attenuation properties. The image intensifier defines the plane of this projection, and is constructed such that it gives submillimeter image resolution.

2.1.1 Distortions

Real radiographs differ from this idealized description of the imaging process in a number of ways[11]. A non-uniformity in the intensities of rays emitted from the x-ray tube results in the *heel effect* which is characterized by a gradual change in average image intensity values from top to bottom of radiograph. The image intensifier's response to incident rays is also non-uniform, so that the center of the radiograph is slightly brighter than the periphery. These intensity non-uniformities can be corrected by machine-specific calibration of the radiographic data. The image intensifier also introduces a geometric distortion called the *pincushion effect.*[1] Fortunately, software methods exist to correct this geometric distortion, which can be characterized by imaging a reference phantom[26][25].

2.1.2 Differentiation of Tissue Types

Due to the integral in the above radiographic-imaging equation, structures are resolved in radiography both by their local attenuation properties and by their extent. In decreasing order of attenuation-coefficient magnitude, we have bone; soft tissue and blood; and air (with value zero). Due to the large contrast between these divisions, bone and air are well-resolved in radiographs; however, differences between the coefficients of soft tissue and blood are not so great, so that the details of soft-tissue structures cannot be well visualized with radiography, except temporarily if high-attenuation-coefficient contrast mediums can be introduced (as they can into blood vessels, for instance). Interventional instruments are made out of radiopaque materials, so that they are very easily discerned in fluoroscopic images.

2.2 MR

We will discuss MR imaging from a qualitative point of view, primarily in relation to fluoroscopy. For details on the image formation, readers are refered to [23] or [12] which is an online text.

Magnetic Resonance Imaging (MR) is a technique which can produce volumetric renderings of anatomical structures based on the response of their component atomic nuclei to an external magnetic field. MR volumes are produced by measurements

[1] The next generation of fluoroscopic will have solid state image intensifiers that should not produce this kind of distortion.

and processing taken over tissues located in adjacent transverse slices of anatomy; these slices are stacked to form a volumetric image. The in-slice resolution is typically better than that between slices. High resolution MR is typically composed of slices containing 256^2 pixels of area 0.9^2 mm^2, whereas the slices are spaced by 1 mm. Low resolution MR has slices of the same number of pixels, but their pixel areas are 1.25^2 mm^2, and their slices separated by 4 mm [13]. Figure 2-2 shows the three planes cutting through the center of a low resolution MR aquistion (limited to the brain cavity). Within slices, pixel intensities map the alignment intensity of hydrogen nuclei

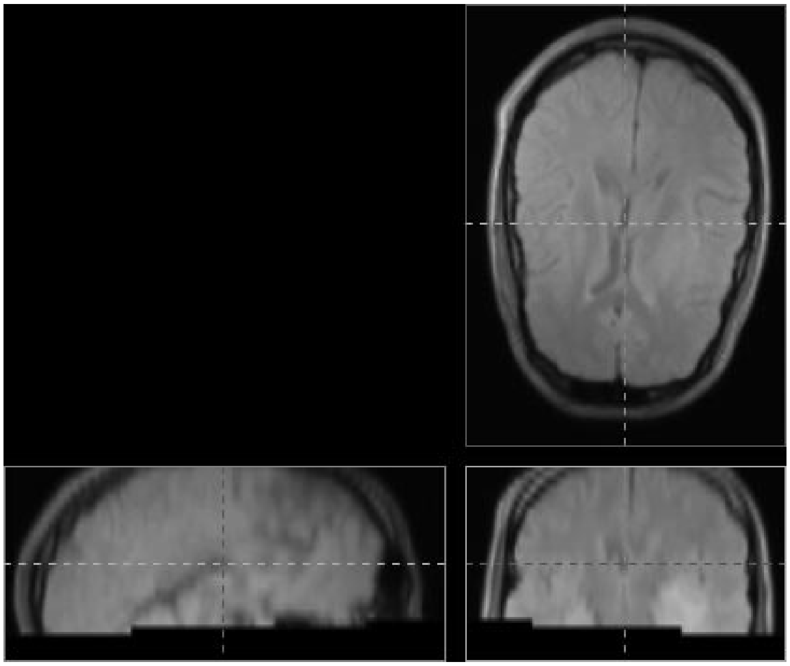

Figure 2-2: Three Central Planes of a Proton Density MR Volume

(protons) at those locations to an external, uniform magnetic field. Variations in the imaging technique emphasize different characteristic magnetic relaxation times (denoted T1 and T2) of aligned protons. These properties relate directly to the proton density of the nuclei, not the tissue density, so MR identifies tissue properites which are very different from the x-ray attentuation properties rendered in fluoroscopy. As a result, the two modalities have very different functionalities. Most notably, MR renders soft tissue variations very well due to their high hydrogen concentrations and differing chemical composition. On the other hand, as hard bone is relatively anhydrous, it is given the same low intensity as is air in MR. Contrast this to fluroscopy which poorly distinguishes soft tissues, but whose rendering of bone most contrasts that of air.

MR imaging is subject to a number of nonidealities including intensity variations and geometric distortions. The intensity variations are the product of nonuniformities in the imaging apparatus and inhomogeneties in the magnetic fields and RF pulses used in imaging [14]. Geometric distortions are most pronouced at boundaries between regions of significantly different magnetic susceptibility (e.g. soft tissue/bone, tissue/air). Software methods are available for the correction both of intensity variations [24] and geometric distortions[8].

2.3 CT

Like MR, *Computed Tomography* (CT) can be used to produce volumetric images of anatomical structures by the stacking of slices produced from measurements taken over transverse slices of anatomy[23]. High resolution CT is typically composed of slices containing 256^2 or 512^2 pixels of area 0.6^2–0.9^2 mm^2, with slices spaced by 1.5 mm. Low resolution CT has slices of the same number of pixels, with pixel areas 0.65 mm^2, and their slices separated by 4 mm [13]. A low-resolution CT volume is shown in Figure 2-3. Compare it to the MR of the same head in Figure 2-2.

CT imaging is based in x-radiography and measures tissues' x-ray attenuating properties in the same way that fluoroscopy does. Each CT slice is produced by rotating an apparatus around the patient that casts a plane of x-rays through the patient in a number of different directions. Processing is then performed such that a CT slice gives a map of the x-ray attenuating properties of the tissue section at each position. Conventionally, each of these attenuation coefficients are rescaled by a linear function to produce an intensity called a *CT number*. CT intensites differ from those given by fluoroscopy, as CT is processed with x-rays of energy less than those used in fluoroscopic radiography. This has the effect that there is less contrast

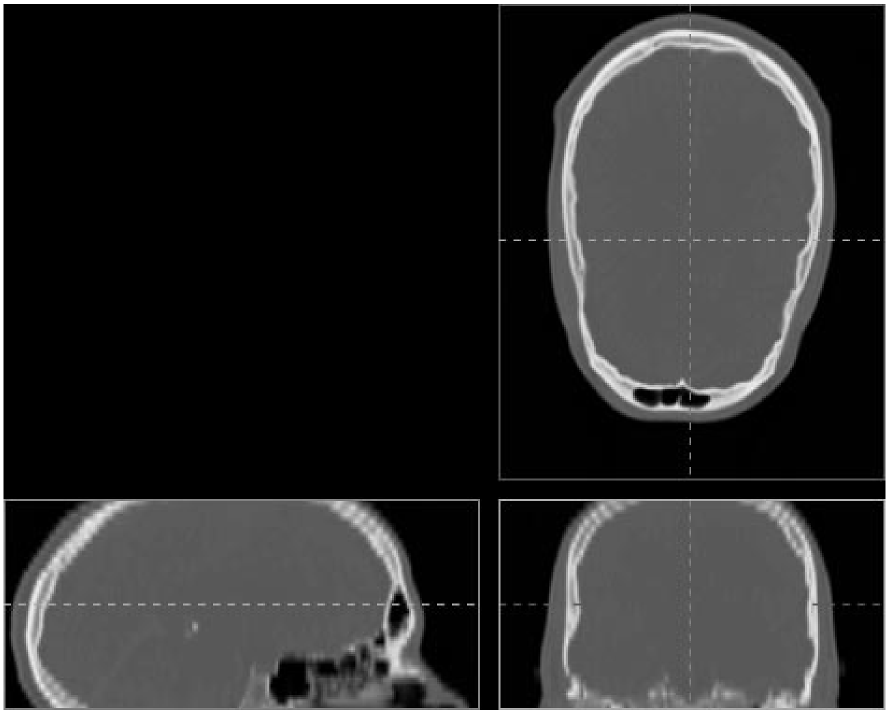

Figure 2-3: Three Central Planes of a CT Volume

between the attentuation coefficients of bone and soft tissue.

Modern CT imaging does not suffer from any significant geometric distortions, and is often used as the "gold standard" basis for trying to correct geometric distortion in MR.

2.4 Data Representations

In this section we'll discuss the manner in which medical data is typically stored in memory as this will bear on our discussion of DRR algorithms in Chapter 4.

Medical image data represents a sample of modality-specific measurements taken over a distribution of locations in space. Accordingly, this data consists of two parts: the measured values of these samples, and a mapping between each sample and its *metric coordinates*, the location in real space at which it was taken, in units of distance (such as mm).

2.4.1 Organization in Memory

Samples are taken in regular intervals so that the set forms a rectilinear array in two or three dimensions. This array is typically organized in memory in the way which streamlines reference to a particular sample $u(i, j, k)$ by its indices (i, j, k) along the array dimensions, which we'll refer to as the sample's *data coordinates.* A volumetric dataset consisting of S slices of size $R \times C$ samples, would occupy block of memory of size $\theta(S \times R \times C)$ within which each of S contiguous blocks (each of size $R \times C$) represents a slice of the data. Within each slice are R blocks of size $\theta(C)$ that each represent a row of voxels in its respective slice. Within each of these rows are the C sample values for that row, one for each volume column. By this organization, the sample with data coordinates (i, j, k) is indexed $(k \cdot R \cdot C + j \cdot C + i)$ in the volume's block of memory, where i is the column index, j is the row index, and k is the slice index.

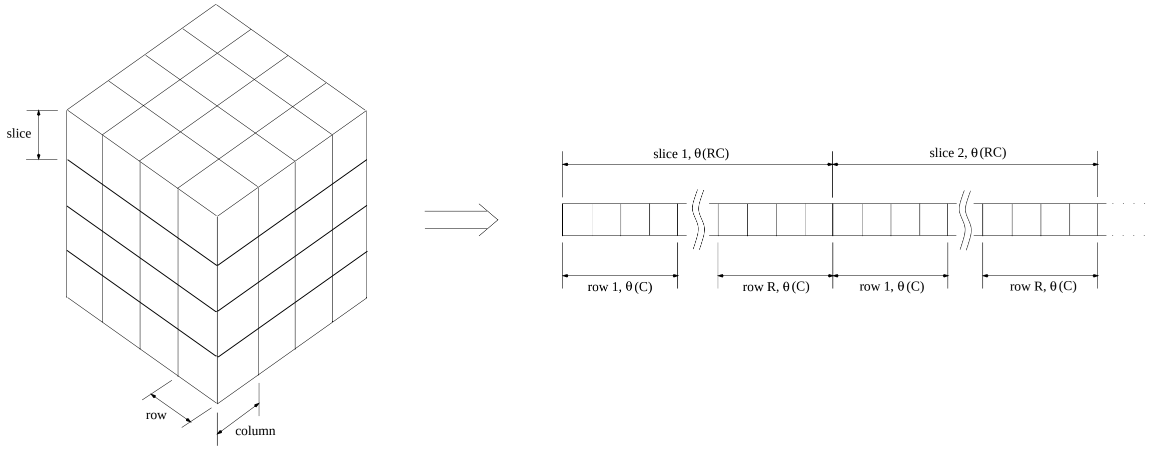

Figure 2-4: Volumetric Data in Memory

2.4.2 Data Size Relative to Cache Sizes

In processing any data, it can be useful to consider the memory architecture in which we perform that processing. Memory access overhead can be the most taxing component of data processing if algorithms do not have data reference patterns which exhibit the kind of spatial and temporal locality that takes advantage of computer caching schemes [2]. We give an overview of general memory achitectures and efficient memory access patterns in Appendix A. Here we outline the data dimensions of typical medical images, along with the cache sizes of the UltraSPARC architecture[19], on which all our implementations in this thesis were performed. Only slice sizes are given for volume data, because the number of slices may vary depending on the anatomy imaged. A typical full head CT or MR might have 125 slices, whereas an aquistion which only includes the brain might have less than 30 slices. Because projections are only taken rougly parallel to slices, these are the only dimensions that are really relevant. Contrast these slice sizes to the cache sizes on a typical workstation.

Table 2.1: UltraSPARC memory cache sizes

Cache	Size
L1	16K
L2	512K–4M

Table 2.2: 8-bit-pixel medical image sizes

Modality	Number Pixels	Storage Size
fluoroscopy	1024 × 1024 pixels	1M
MR	256 × 256 pixels/slice	64K
CT	512 × 512 pixels/slice	256K

2.4.3 Relation of Metric and Data Coordinates

With this medical data structure, samples may be mapped to their metric coordinates by a transformation from their data coordinates, and vice versa. In general, the transformation between a dataset's metric and data coordinates is affine, that is, one allowing translation, rotation, and ansiotropic but uniform stretching. Between two 3d coordinate systems, an affine transformation T_{affine} may be parameterized by an invertible 3x3 matrix **A** and 3-vector **b**. The transformation is calculated:

$$\mathbf{y} = T_{affine}(\mathbf{x}) = \mathbf{A}(\mathbf{x}) + \mathbf{b} \tag{2.2}$$

2.4.4 Interpolation for Inter-Sample Values

When a dataset is referenced by particular metric coordinates **x**, those coordinates will not generally transform to integral data coordinates that reference an actual sample in the array of data. To account for this, some interpolation method must be employed to determine the values which corresponds to points that fall between actual samples (in the case where a point falls outside the bounds of the dataset entirely, we set its values to zero). We'll consider two simple methods here, trilinear interpolation and nearest-neighbor interpolation. Their small computational burden make them well-suited for the kernel of an iterative procedure where higher-order methods are not required.[2] The more simple method, nearest-neighbor interpolation, picks the value for a point **v** by rounding each real-valued coordinate.

$$\mathbf{v}_{nn}[i_1, i_2, \ldots] = \mathbf{v}[\mathrm{round}(i_1), \mathrm{round}(i_2), \ldots] \tag{2.3}$$

Trilinear interpolation, on the other hand, works by doing linear interpolation in one coordinate direction among a point's eight nearest-neighbors, and then interpolating those results in another coordinate direction, and so forth for all coordinate directions. The trilinear-interpolated value v of a voxel **v** falling among voxels $\mathbf{v}_{ijk}$ as depicted in Figure 2-5 is given by

$$v = \sum_{i,j,k \in \{0,1\}} v_{ijk} \lambda_x^{(i)} \lambda_y^{(j)} \lambda_z^{(k)} \tag{2.4}$$

[2] Hill and Hawkes [6] show that artifacts produced by nearest neighbor and trilinear interpolation cause artifacts which frustrate alignment by correlation in multi-modal contexts, due to correlation's sensitivity to outliers.

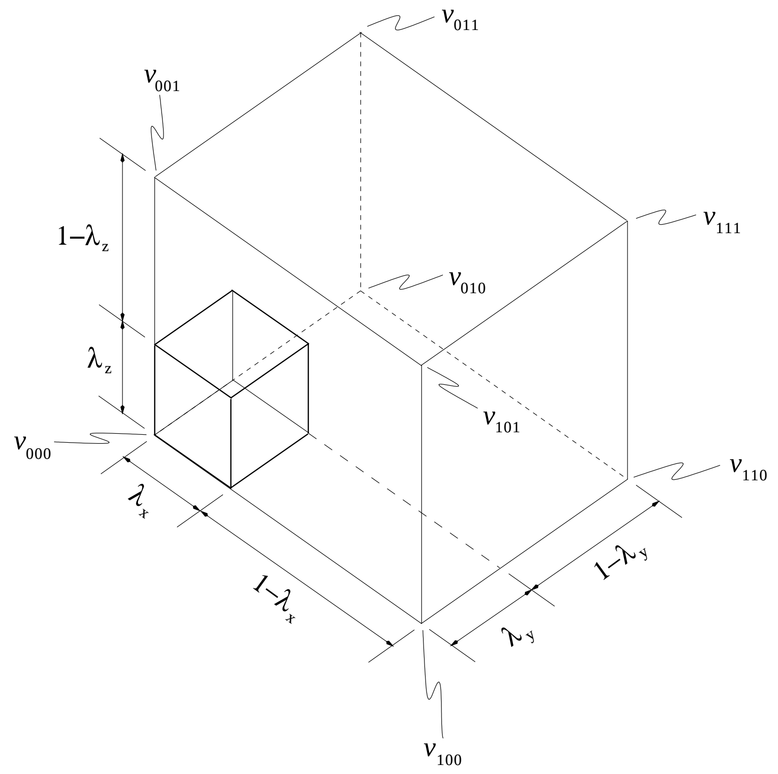

Figure 2-5: Framework for Trilinear Interpolation

where

$$\lambda_x^{(i)} = \left\{ \begin{array}{rl} \lambda_x, & i = 1 \\ (1-\lambda_x), & i = 0 \end{array} \right\} \tag{2.5}$$

where λ_x is the fractional distance from $\mathbf{v}$ to $\mathbf{v}_{0jk}$ in the x-direction. $\lambda_y^{(j)}$ and $\lambda_z^{(k)}$ are defined equivalently. This value is clearly linear in each of the eight contributing nearest-neighbor values v_{ijk}, each of whose respective contribution falls off (almost) linearly with distance from $\mathbf{v}$ (although with a different coefficient in different directions and some small quadratic and cubic terms). Not only does trilinear interpolation require more computation than nearest-neighbor, but it also requires accessing eight dataset samples as opposed to nearest-neighbor's one.

2.5 Summary

In this chapter, we have contrasted the process of image formation in MR and Flouroscopy, which occur by very different physical processes, to provide context for the MR-fluoroscopy registration problem. We have also discussed the size and representation of medical image data in computers, as this wil be important to our development of the voxel-projection algorithm in Chapter 4.

Chapter 3

Image Registration

In this chapter, we formalize the problem of image registration and outline the several categories of regitration methodolgies which have been applied in the medical context. This discussion complements that of Chapter 2 on the characteristics of MR and fluoroscopic data, in that they will together motivate the approach we take to developing an algorithm for the registration of these data.

3.1 Problem Definition

Registration is the process of aligning two images U and V rendered from the same physical structures. A registration problem is solved by determining the best values for the parameters that govern this alignment. These parameters may be ones of any type of spatial transformation T between two coordinate systems.

What precisely constitutes "alignment" is not completely clear when the images are formed by dissimilar processes which may introduce different types and degrees of distorion. However, when we are using one image to localize structures in the other, as is typical for surgical guidance applications, we would expect that the optimal transformation accurately aligns these structure in space.

3.2 Transformation Parameters

The type of transformation T employed to register images must be chosen to match image-specific characteristics. This choice is governed by the nature of imaging process and any geometic distortions which have been introduced.

Rigid-body This allows only translation and rotation. It is appropriate when rendered anatomies are inelastic (e.g. framed by bone) and when image distortions are small or can be corrected by some means of preprocessing, so that the image's metric coordinates accurately locate sample intensities. Such a transformation is applied between the two images' metric coordinate frames. A 3D, rigid-body transformation is governed by six parameters, three each for translation and rotation.

Affine This allows translation, rotatation, skew, and scaling. It is appropriate when one or both of the images are subject to some affine distortion.

Locally-varying These are appropriate when image geometries have geometries which vary non-linearly over different regions of the image space. This might be characterisic of the registration of soft-tissue anatomy, which is likely to have deformed non-rigidly between image acquisitions.

3.3 Medical Image Registration Methodologies

The manner in which medical images are registered may be divided into three categories which are described here.

3.3.1 Alignment of Fiducials

In this methodology, markers are fixed on the patient which can be confidently imaged by both involved modalites. Markers might be special beads adhered to surfaces of the skin, or part of an apparatus rigidly fixed to bone or dental structures. This approach has the advantage that registration is straightforward and accurate; however, it does require that fiducials be fixed during aquisition of both images. This may not be possible, for instance when we want to bring to bear on a procedure, preoperative data which was imaged without the procedure in mind.

3.3.2 Feature Alignment

In this registration methodology, anatomical features which can be confidently identified and segmented from the images are aligned. As with alignment by fiducials, this registration is straightforward and fast, but only after features have been segmented. Unfortunately, the nonlinearities and irregualrities inherent in medical images make robust feature identication difficult to automate and slow. Therefore, this type of approach may be impractical[9]. Furthermore, because segmentation techniques must often be specialized to particular imaging modalities and anatomical regions, a particular feature-alignment methodology may not be generally applicable.

3.3.3 Intensity-based Similarity Measure

This is the most generally-applicable of the registration methodologies mentioned here. In this approach, the values of image samples are compared directly without making any additional assumptions about the data. To match such an approach to a specific problem, an appropriate function is chosen to assign a measure of similarity $\mathrm{SIM}(U, V)$ between two images U and V based only on the values of their respective samples $U(\mathbf{x})$ and $V(T(\mathbf{x}))$ at all positions $\mathbf{x}$ and under the transformation T from the coordinate frame of image U to that of V. We will refer to this function as an *objective function* or *similarity mesaure.* Registration is then performed by means of an optimization algorithm which maximizes the objective function over the domain of possible transformations. The intensity-based approach has been successful in a variety of multi-modal, medical image registration contexts, due to its very general structure [22][9][1][20][11].

3.4 Intensity-based Image Registration

As described above, these methods require the existence of an objective function which takes on maximal value at ideal alignment, and which has characteristics such that it can be efficiently maximized by automatic optimization. Due to the large parameter space in which even the most simple rigid-body transformations reside, computational efficiency is a significant concern in intensity-based methodolgies. Without an appropriate pairing of objective function and optimization scheme, these methodologies can be impractically slow. We will explore these design considerations in concert, and mention optimization methods pertinent to the medical context. We will defer discussion of specific similarity measures, and consider them later in terms of 3D-2D image registration.

3.4.1 Matching Measure to Optimization Method

A variety of algorithms have been developed for the maximization of functions for which no closed-form solution is available[15]. Certain function characteristics will govern which of these methods is selected for optimization. Among these are the function's convexity and the availability of an analytic expression for the function's derivatives. Because iterative optimization algorithms typically use local information in the function variation to direct a search of the function domain, access to derivative information can be very useful. For the same reason, the success of an optimization is dependant on whether that local information can efficiently direct an algorithm to a function's global maximum. Therefore the existence of local maxima and regions of vanishing variation (i.e. the function is constant) can frustrate an optimization scheme. The function geometry of a long, narrow peak is also particularly difficult for optimization schemes to navigate (Numerical Recipies in C [15] has a good discussion on this and other topics relating to function optimization).

3.4.2 Optimization Schemes

In this section we'll mention a number of optimization strategies which have been applied in the medical context.

Hierarchical Search In this strategy, the input images are successively smoothed and downsampled to produce images of lower resolution. A search of the transformation space is performed using each pair of lower resolution images, and the optimum transformation is used as a starting point for high resolution levels. Using a hierarchical or *multi-resolution* approach may be applicable in conjunction with almost any optimization scheme. As long as optimization using lower resolution images provide a good starting point for higher levels, any scheme may be sped up by working on smaller images.

Gradient Descent The gradient descent uses the objective function's gradient to direct its search of the transformation space. *Conjugate Gradient Methods* use the gradient in conjunction with other considerations to improve on the search process [15]. These strategies perform poorly on non-convex functions.

Non-Gradient-Based These methods optimize without access to gradient information. The range of these strategies use local information in different ways to direct search towards a function maximum.

- Powell's Method is a *direction-set* scheme in which optimization is performed by successive line minimizations in strategically choosen directions.

- Three-step Search [18] calculates finite-differences on each parameter of the function domain, and adjusts the parameters accordingly. The approach is hierarchical in the sense that it uses a decreasing schedule of steps sizes for its finite-difference calcuations. A similar method has been used in CT-fluoroscopy image registration [11].

- Uphill-simplex also uses finite-differences to direct its search of the parameter space, but does it in such a way that its exploration is distributed over a region of the function domain. This method is described in detail in Chapter 6.

3.5 Intensity-based, 3D-2D Image Registration

The problem of registering MR to biplanar fluoroscopy is one of spatially aligning the MR volumetric data with the real structures that the fluoroscopic radiographs render. More formally, it is the process of finding the rigid coordinate transformation T which places the MR volume V in this aligned position in the coordinate frames of the fluoroscopic images $I_{fl,1}$, $I_{fl,2}$. This process is depicted in Figure 3-1. Note that the fluoroscopic coordinate frames are related by a rotation θ about an axis parallel to one of each image's major axes (typically, $\theta = \frac{\pi}{2}$ radians). We'll call this transformation $T_{fl,12}$ from the frame of $I_{fl,1}$ to $I_{fl,2}$. Heretofore, when we refer to the transformation T which governs the volume's position, we will be refering specifically to the transformation from the volume's frame to that of $I_{fl,1}$, with the relation between the frames $T_{fl,12}$ fixed and implicit. The 3D-2D nature of registering volume data to radiographs requires that the process of radiographic projection is somehow

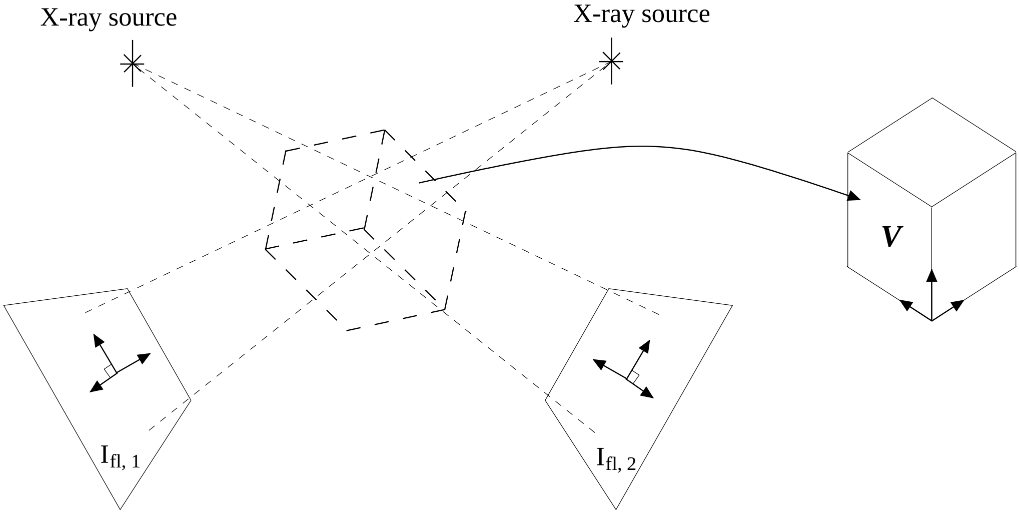

Figure 3-1: Registering volumetric data with biplane fluoroscopy

accounted for. For an intensity-based 3D-2D registration of this kind, the problem is one of finding a way to compare the intensities of radiograph and volume. In the context of CT-fluoroscopy registration, Penney et al. [11] approached this problem by simulating the radiographic projection of the volumetric, so that the resulting image I_{DRR}, called a *Digitally Reconstructed Radiograph* (DRR), can be compared to the real radiograph by a similarity measure. The method by which DRRs are produced are therefore an important step in the 3D-2D registration process, and the method by which they are produced is significant to its success and efficiency.

Because we are considering biplane fluoroscopy, the objective function will be evaluated on the model radiographs $I_{fl,1}$, $I_{fl,2}$ and the DRRs $I_{DRR,1}$, $I_{DRR,2}$ produced from the volume given transformation a T. $I(\mathbf{x})$ will refer to the intensity value of image I, when $\mathbf{x}$ is a sample location, either in the image's metric or data coordinates.

In the next section, we will discuss how DRRs are produced and how they compare to real fluoroscopic radiographs. In light of this discussion, we will describe a number of similarity measures which might be applicable to the 3D-2D registration problem.

3.6 Reconstructing Radiographs

DRRs are typically produced by a method called *ray-casting* that closely parallels the physical process of fluoroscopic image formation. Recall from chapter 2, that real radiographs are produced by a process which is effectively the perspective projection of the attenuation coefficients of the imaged object. In that radiographic projection, the attenuation coefficients are integrated along rays which extend from the x-ray source to positions on the image intensifier, which defines the plane of projection. In *ray-casting* DRR production, a volumetric dataset's voxel intensities are summed along rays connecting the x-ray source and each DRR pixel. An interpolation method (e.g. nearest-neighbor, trilinear) must be employed for this summation to compute intensity values for samples along each ray which fall in inter-voxel positions; this interpolation method impacts the resolution and quality of the DRR.

Note that DRR pixels may be made to align precisely with those of the radiograph, irrespective of the volume orientation under T. This gives the computational advantage of being able to evaluate objective functions exactly at pixel locations for both the fluoroscopic images and the DRRs, without requiring interpolation.

The details of algorithms for DRR production by ray-casting are discussed in Chapter 4, along with a presentation of computationally efficient means of simulating ray-casting, called *voxel-projection*.

3.6.1 Comparing DRRs to Real Radiographs

The ray-casting process makes considerable sense when the volumetric data is CT, whose intensities are proportional to the attenuation coefficients of imaged tissues (at the x-ray energy used to form the CT), although differences in tissue attenuating properties due to the differing x-ray energies used in CT and fluoroscopy, may result in CT-DRRs having less contrast between bone and soft tissue.

The validity of ray-casting is less clear when the volumetric data is MR, whose intensity values are certainly not a function of tissue density. The most conspicuous inconsistency in producing MR-DRRs is the fact that MR gives high-density hard bone the same low intensity as air (and thus the same small attentuation coefficient), so that bony structures are less pronouced in MR-DRRs.[1] Notwithstanding, MR-DRRs produced by ray-casting may still exhibit some gross structural features and intensity correspondences of which an objective function can take advantage.

Of course, geometric distortions inherent in fluroscopy (e.g. pincushion distortion), are not modeled by the ray-casting process. These distortions should be corrected before a fluoroscopic image is compared to a DRR, rather than complicating the process of DRR production by trying to model that warping.

Differences between DRRs and real radiographs do not end with differences in the imaging process by which they are produced. For instance, they differ notably in resolution [11], both due to the fact that volumetric data from CT and MR is of lower

[1] While it is not possible to perfectly map MR intensities to values which are proportional to attenuation coefficients[22], work which has been done to make CT intensities resemble those of MR [20], might be applied here to adjust MR voxel values so that MR-DRRs more closely resemble true radiographs. We'll discuss approaches to correct for this in Chapter 7

resolution than is fluoroscopy, but also due to the fact that a volume's voxel spacings are effectively broadened by the divergent rays of projection. This effect will differ for different implementations of DRR production (for instance, for different choices of an interpolation function), as described further in Chapter 4 where we discuss the implementation details of DRR production algorithms.

Futhermore, because volumetric data subsumes only a limited volumetric region, rays which pass through spaces where structures were truncated will produce erroneous pixel intensities where they terminate in the DRR. However, we can generally characterize these rays as those which enter or exit the volume through the top or bottom slice, so they can be automatically omitted on that basis.

3.6.2 Objective Functions

In this section, we present a number of objective function which have been used in other multi-modal image registration approaches, and specifically for CT-fluoroscopy registration. We will discuss them in terms of the kind of joint image behavior by which they are maximized. In these descriptions, the functions $I_{fl}(\mathbf{x})$ and $I_{DRR}(T(\mathbf{x}))$ give the intensity values of images I_{fl} and I_{DRR} at location $\mathbf{x}$ (in the coordinate system of image I_{fl}). T is the transform from image I_{fl}'s coordinate frame to that of image I_{DRR}'s. To apply the objective function $\mathrm{SIM}(I_{fl},I_{DRR})$ (as all are defined below) when there are two fluoroscopic images taken from different orientations, as in the biplane scenario, we may simply sum the functions evaluated between corresponding radiograph and DRR:

$$\mathrm{SIM}(I_{fl,1}, I_{DRR,1}, I_{fl,2}, I_{DRR,2}) = \mathrm{SIM}(I_{fl,1}, I_{DRR,1}) + \mathrm{SIM}(I_{fl,2}, I_{DRR,2}) \tag{3.1}$$

Let us preface this listing of functions with a note on their evaluation. The summations over $\mathbf{x}$ in the following formulas are limited to values within the extent of the images. The fact that these are discrete summations suggests that they are taken only at certain sample positions, such as the image pixel locations. Though this cannot generally be done over both images[2], because we can align the pixels of the model radiographs and the DRRs irrespective of the volume position (as noted in the previous section), the summations in the objective function definitions can be made to always reference image pixels at integer data coordinates.

Alternatively, certain of these objective functions could very well be evaluated over some sparce sampling of the image data, as has been explored in MR-CT registration [22]. The statistical and information-theoretic measurments (i.e. correlation coefficient, correlation ratio, entropy of the difference image, mutual information, model-based likelihood) are consistent with this kind of approach; whereas, those defined by the structure or variation in the image (i.e. pattern intensity), cannot. The form of the functions specified below does not change if a sampled approach is used.[3].

Correlation Coefficient This measure is widely used in image registration as it is a basic statisical measure. It best identifies the alignment of datasets whose intensities are related by a monotonic function [21], as it is a measure of linear dependance between cooresponding pixel intensities [16].

$$CC = \frac{E[I_{fl} I_{DRR}]}{\sqrt{\sigma^2_{I_{fl}} \sigma^2_{I_{fl}}}} = \frac{\sum_{\mathbf{x}} (I_{fl}(\mathbf{x}) - \bar{I}_{fl})(I_{DRR}(T(\mathbf{x})) - \bar{I}_{DRR})}{\sqrt{\sum_{\mathbf{x}} (I_{fl}(\mathbf{x}) - \bar{I}_{fl})^2} \sqrt{\sum_{\mathbf{x}} (I_{DRR}(T(\mathbf{x})) - \bar{I}_{DRR})^2}} \tag{3.2}$$

where $\bar{I}$ is the sample mean of I.

Entropy of the Difference Image This function operates on the intensity histogram of a single image I_{diff} formed by subtracting one image from a scaled version of the other.

$$I_{diff}(\mathbf{x}) = I_{fl}(\mathbf{x}) - s \cdot I_{DRR}(T(\mathbf{x})) \tag{3.3}$$

[2] Even if $\mathbf{x}$ falls at integer data coordinates in one image, the transformation T will generally cause $T(\mathbf{x})$ to correspond to some inter-pixel location in the other.

[3] However, the intensity histograms used for some measures (e.g. mutual information, model-based likelihood) become estimates of the histogram, generated by a Sum-of-Gaussians [9] or the Parzen Window method [22]

The histogram gives the probability $P(w)$ of selecting intensity w from the samples of the difference image I_{diff}.

$$H(s) = -E_{I_{diff}}[\log P(w)] = -\sum_{w} P(w) \log P(w) \tag{3.4}$$

The scale factor s is set to the value which minimizes the $H(s)$. Entropy roughly measures the spread of a distribution [17], so it should be a successful objective function when alignment minimizes variation in the difference image intensities. Due to the fact that all samples are weighted equally in the histogram, this measure should be robust to infrequent outliers in the difference image distribution [11].

Mutual Information Mutual Information (MI) has been used successfully in a number of multi-modal image registration schemes, such as those registering C-T to MR and MR to PET. MI operates on probabilities which can be approximated by the joint intensity histogram formed from image sample correspondences $I_{fl}(\mathbf{x}), I_{DRR}(T(\mathbf{x}))$ under T. In other words, the histogram gives the probability $P_{I_{fl},I_{DRR}}(v, w|T)$ of selecting the intensity pair (v, w) from spatially-aligned samples of I_{fl} and I_{DRR} given T. $P_{I_{fl}}(v)$ and $P_{I_{DRR}}(w)$ are the marginal intensity distributions for images.

$$\begin{aligned} MI &= E_{I_{fl},I_{DRR}}\left[\log \frac{P_{I_{fl},I_{DRR}}(v, w|T)}{P_{I_{fl}}(v)P_{I_{DRR}}(w)}\right] \\ &= \sum_{v,w:P_{I_{fl}}(v)>0,P_{I_{DRR}}(w)>0} P_{I_{fl},I_{DRR}}(v, w|T) \log \frac{P_{I_{fl},I_{DRR}}(v, w|T)}{P_{I_{fl}}(v)P_{I_{DRR}}(w)} \end{aligned} \tag{3.5}$$

The success of mutual information in image alignment does not depend on a functional relationship between the intensities of the two images, but only on a statistical one [22], such that MI is effective when registration aligns the pixels with the most probable values.

Pattern Intensity Pattern Intensity has been applied in the registration of CT and radiographic data [11]. The function operates on the difference image I_{diff} as defined in the section on "Entropy of the Difference Image" above. Pattern Intensity is successful when there are a minimum number of *structures* in the difference image. An image pixel located at $\mathbf{x}$ is part of a *structure* when its intensity $I_{diff}(\mathbf{x})$ differs drastically from the intensities of all pixels within the neighborhood of radius r.

$$PI(s,r,\sigma) = \sum_{\mathbf{x}} \sum_{\mathbf{y}:|\mathbf{y}-\mathbf{x}|\leq r} \frac{\sigma^2}{\sigma^2 + (I_{diff}(\mathbf{x}) - I_{diff}(\mathbf{y}))^2} \tag{3.6}$$

The constant σ is added so that small deviations in intensity do not shift the function much from its maximum value. Pattern Intensity is robust to large differences in pixel intensity Δ due to its $1/(1+\Delta^2)$ form. Furthermore, differences between U and V which occur on a scale larger than r, such as those due to soft tissue, have little effect on the function.

In [11], the constants were set $\sigma = 10$ and $r = 3$ for full-resolution images. For coarse images, it was found that using $r = 5$ maintained the function's performance.

Correlation Ratio Correlation Ratio was presented by [16] for multi-modal volumetric data registration. This function measures the functional dependence between the intensities of corresponding samples.

$$CR(I_{fl}|I_{DRR}) = \frac{\mathrm{Var}(E[I_{fl}|I_{DRR}])}{\mathrm{Var}(I_{fl})} = 1 - \frac{E_{I_{fl}}[\mathrm{Var}(I_{fl}|I_{DRR} = w)]}{\mathrm{Var}(I_{fl})} \tag{3.7}$$

The rightmost expression is more easily implemented in practice. This function is evaluated over the joint intensity histogram which gives the probability $P_{I_{fl},I_{DRR}}(v,w|T)$ that a randomly-selected sample at position $\mathbf{x}$ will have intensities $(I_{fl}(\mathbf{x}), I_{DRR}(T(\mathbf{x}))) = (v,w)$.

Note that the above expression is not symmetrical in I_{fl},I_{DRR}, so the behavior of $CR(I_{DRR}|I_{fl})$ may be very different from that of $CR(I_{fl}|I_{DRR})$.

Model-based Likelihood [9] proposed a methodology by which data of different modalities are registered based on the a prior model for the joint intensity distribution for aligned data of those modalities. The prior model M is simply a structure which gives the probability $P(v, w|M)$ that corresponding samples from I_{fl} and I_{DRR} have intensities $v = (I_{fl}(\mathbf{x})), w = I_{DRR}(T(\mathbf{x})))$, when T aligns the images. Based on the model, this objective function gives the log-likelihood of a sample of the intensity correspondences between the images based on the model.

$$L(M) = \sum_{\mathbf{x}} \log P(I_{fl}(\mathbf{x}), I_{DRR}(T(\mathbf{x}))|M) \tag{3.8}$$

3.7 Summary

In this chapter, we have framed the MR-fluoroscopy registration problem in the broader context of medical image registration problems. We have also disussed the application of an intensity-based registration scheme to this problem which would involve the selection of an appropriate objective function and optimization algorithm, as well as a method for simulating the radiographic process on volumetric medical data. This frames the remainder of the thesis, in which we investigate these three issues.

Chapter 4

DRR-Production Algorithms

4.1 Introduction

Simulating x-ray images of the volumetric dataset is essential to our registration method. It is an especially important topic in this 3D-2D registration context as it is by far the most computationally-intensive part of every iteration of any intensity-based, automated registration approach. Indeed, the production of DRRs is governed by its own dimension $n = \sqrt{\text{number DRR pixels}}$ and the dimension of the volume from which it is produced $m = \sqrt[3]{\text{number volume voxels}}$, so its computation can be bounded above by $\theta(n^2 m^3)$, with terms that may be associated with large computational constants. On the other hand, evaluation of typical objective functions are $\theta(n)$ or $\theta(n^2)$ and associated with smaller constants (pattern intensity [11] and correlation ratio [16] are some notable exceptions). Unfortunately, every time an automated registration algorithm would like to measure the objective function on a new volume *pose* (position and rotational orientation), new DRRs of the volume must be produced with the volume in its new orientation and position; this computational burden is unlikely to be obviated by schemes of precomputation or use of previous DRRs, both because the conical geometry of a radiographic projection is not aligned with the rectilinear geometry of the volume data, and also because the volume's pose may vary in increments unrelated the volume data geometry, and unknown at the outset of registration (if we knew what poses an algorithm would test, then we would already know the pose of ideal alignment that it arrives at).

One way to improve the speed of projection is by reducing dataset size. For instance, Penney et al. [11] described the automatic registration of CT and fluoroscopy in the lumbar region of the spine, where projections were limited to the L3 vertebra alone. Of course, the serviceability of this approach is limited by the compactness of bony structures (the most prominent features in x-ray images) in the region where registration is being performed. We may not be able to take advantage of this method in registering images of the head where the skull spans a large region. Even if we were to limit our focus to the face, only sagittal (and transverse, which are nonsensical here) DRRs are free of influence from the rest of the skull.[1] Another way to improve projection speed is to reduce the resolution (i.e. downsample) of the volume and/or radiograph; this leads to multi-resolution or hierarchical registration approaches explored in many registration contexts. Yet another way to improve projection speed, and the topic of this section, is by choice of the digital-projection algorithm. We'll describe two methods for producing DRRs: *ray-casting*, and one that we'll term *voxel-projection*, the latter of which has a significant speed advantage over the former. Before we get started, however, we'll preface our discussion with a note on coordinate systems which will have bearing throughout the chapter.

4.1.1 An Efficient Coordinate Transform

For the projection algorithms which follow, because we ultimately need to use data coordinates to reference our volume and projection datasets, we should convert from metric coordinates at the outset of our iterative algorithms to avoid conversion overhead in its inner loop. Furthermore, because the main loops of our algorithms reference corresponding points in both the volume and the DRR, it is advantageous to collapse the conversion between their data coordinates through the intermediate rigid transformation T between metric coordinates, into a single affine transforma-

[1] X-rays forming a coronal image pass through the back of the skull and the face, and along the skull's side. Though we have not explored the effect of omitting the effect of skull's back and side in registrations, it is likely that ignoring the sides, where x-rays running roughly parallel to them are highly attenuated, would make the biggest difference.

tion T_{affine} as shown in Figure 4-1. Note that although the radiographic data exists

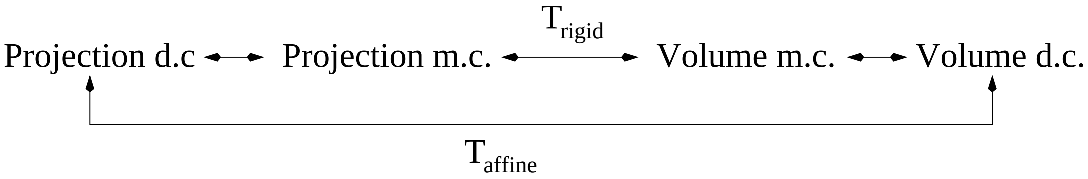

Figure 4-1: An affine data coordinate transform

only in two dimensions, its data coordinates may be defined by adding an orthogonal, out-of-plane dimension of arbitrary metric scaling. It is in this dimension that the x-ray source resides.

We will find that this characterization will provide computational advantages that are especially important to the voxel-casting approach.

4.2 Ray-casting

The ray-casting algorithm models ideal radiographic image formation by simulating the attenuation that x-rays experience as they pass through a dense object. It does so by taking the digital volume intensities as representative of local x-ray attenuation coefficients, and performs a Reimann sum of those values along each ray (approximating an integration). To the result Σ we may then apply the function which yields radiographic intensity noted in Section 2.1:

$$I(x, y) \propto \exp(-\Sigma) \tag{4.1}$$

This makes the most sense when volume intensities correspond to the density of the actual structures it depicts. This is the case for CT, and less so for MR, but as noted in Section 3.6.1, the ray-casting algorithm produces images from MR whose structure is similar to that of a true x-ray image of the object the MR depicts.

4.2.1 Ray-casting Parameters

The ray-casting algorithm is governed by a number of the parameters depicited in Figure 4-2. Those that are specific to the x-ray device which is being modeled include focal length and image pixel resolution. Ray-casting is also governed by parameters specific to the volumetric data being projected: volume voxel resolution and the projective sum's spatial sampling period. The volume's resolution (in the direction parallel to the projective plane) and the fluoroscope's resolution both bear on the DRR's resolution. The latter (fluoroscope resolution) is typically larger than the former, and thus provides an upper bound on the DRR's resolution in practice; however, one might reasonably reduce the DRR resolution to the level of the volume's, as the volume from which the DRR is produced can probably provide no precision beyond that. The sampling period used to sum along each ray Δ_{ray} is reasonably set to the voxel resolution in the direction of each projective ray (which is roughly the same for all rays in a single projection when the object is sufficiently far from the x-ray source) so that each voxel's intensity contributes roughly once to each projective summation.

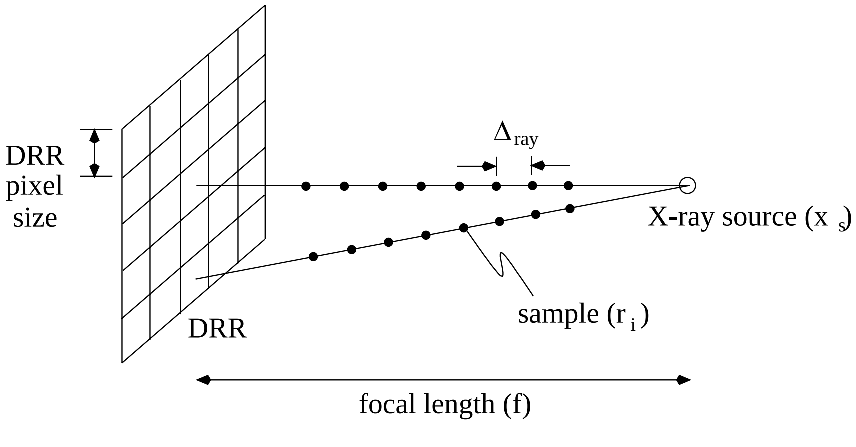

Figure 4-2: Ray-casting DRR Production

4.2.2 The Ray-casting Algorithm

This algorithm goes through the pixels of the DRR exhaustively, and determines each pixel's value by summing volume intensities VOLUME($\mathbf{x}$) along the ray $\mathbf{r}$ that connects it to the x-ray source.

1. Crop the volume to exclude blank space (below some threshold intensity) which may surround the imaged object.
2. Set $\mathbf{x}_s$ as the x-ray source according to the focal length f.
3. For each DRR pixel $\mathbf{x}_p$:
   (a) Initialize DRR($\mathbf{x}_p$) $= 0$.
   (b) Find the ray $\mathbf{r} = (\mathbf{x}_p - \mathbf{x}_s)$ connecting the target DRR pixel and the x-ray source.
   (c) Clip the ray $\mathbf{r}$ with the volumetric dataset at $\mathbf{r}_0$ and $\mathbf{r}_m$ to avoid summing blank voxels.
   (d) Find DRR($\mathbf{x}_p$) $= \sum_{i=0}^{m}$ VOLUME(r_i) where $\mathbf{r}_i = \mathbf{r}_0 + i * \Delta_{\text{ray}} * \frac{\mathbf{r}}{||\mathbf{r}||}$, $i = 0, \ldots, m$, are samples spaced by Δ_{ray} along the ray $\mathbf{r}$ clipped by the volume. For each sample an interpolation method must be used to find the intensity for each non-integral ray sample.
4. Fix the intensity DRR($\mathbf{x}_p$) $\propto$ exp($-$DRR($\mathbf{x}_p$)).

Note that step 3(d) comprises most of the burden for calculating each DRR pixel. In implementation, there are a number of ways to speed this algorithm's operation, some of which are stated explictly in the above outline.

1. Clip each projective ray with the extent of the solid-rectangular volume cropped to exclude any surrounding low-intensity regions.

2. Because a large number of volume samples are referenced in the loop which calculates the projective sum for each DRR pixel, the geometric calculations ought to all be done in the volume's data coordinates. In the later section on the voxel-projection algorithm, we will discuss a way to efficiently extrapolate the volume coordinates for all DRR pixels from those of a single pixel, by taking advantage of the linearity of the affine transformation relating the image and volume data coordinates. The resulting reduction in operations has less effect on

running time in this ray-casting context where the time savings are eclipsed by the overhead of non-sequential data access (which we discuss in Section 4.2.5).

3. Using nearest-neighbor interpolation instead of trilinear reduces running time by about 50 percent (See Section 4.4). This is likely due to the reduction in the number of volume voxels referenced more than the reduction in operations. The resulting DRRs are more “blocky,” as we would expect.

4. Going through the DRR along its columns (in data order) is most straightforward; however some speed gains can come from covering the DRR in small blocks and summing the rays of those blocks in parallel (i.e. for all rays terminating in each block, start at roughly the same distance from the source, and increment all of their sums before referencing the next samples). Because samples from nearby rays come from the same regions in the volume, we would hope that summing rays in parallel will yield more cache hits. Unfortunately, in practice, this only works for block sizes which are too small to yield time-savings large enough to significantly overwhelm the additional computational overhead that comes from going through the DRR in blocks, rather than in data order. While we have not explored the possibility, were we to represent compact regions of the volume as contiguous blocks in memory, this approach may become serviceable.

4.2.3 Resolution of Ray-cast DRRs

The resolution of the DRRs produced by ray-casting is on the order of the smaller of the DRR pixel resolution and the resolution of the volume from which it was produced. The DRR samples do not suffer from any smoothing effects beyond the interpolation that occurs at ray samples within the volume, as rays are cast directly to projection data coordinates. The effect of the interpolation method is less clear and relies somewhat on the smoothness of the original volume data. If we assume that trilinear interpolation accurately describes what would have been the intensity of a sample at non-integral data coordinates (which is probably a good assumption), then it is clearly superior to nearest-neighbor interpolation in respect to resolution.

4.2.4 Complexity of Ray-casting

The computational complexity of ray-casting is given by $\theta(n^2 \frac{m}{k})$, where

$$\begin{aligned} n &= \sqrt{\text{DRR pixels}} \\ m &= \text{voxels along projective ray} \simeq \sqrt[3]{\text{volume voxels}} \\ k &= \text{spatial sampling period along projective ray} \end{aligned}$$

But note that, in fact, the image size is a function of the volume size (and their relative resolutions) because the image size is only as big as the volume can project into. Thus, the image size n^2 is roughly proportional to m^2, the extent of the volume's projective image, and the complexity here is really $\theta(m^3)$. This asymptotic expression has a large constant coefficient associated with it due to cache misses from non-sequential volume data access as decribed in the next section.

4.2.5 Memory Access Patterns of Ray-casting

The speed of ray-casting methods is negatively effected by their accessing of the $\theta(m^3)$ volumetric data in a manner that does not display locality in relation to the organization of the data in the computer's memory. Given the typical organization of medical data described in Section 2.4.1, and due to the fact that radiographs are generated by divergent (non-parallel) rays, ray-casting algorithms will never sequentially access volumetric data organized this way. Indeed, even if the focal length of the projection and the volume position were such that the rays passing through the extent of the volume were roughly parallel, a registration algorithm would very rarely encounter poses in which the projective rays align with any of the volume's data axes, and even more rarely encounter poses in which the projective summations access the volume in data order for even short intervals. In practice, the rays will almost always traverse the 3d matrix cutting accross slices and rows, and thereby cause little locality of reference. Furthermore, without changing the represenation of the volume, summing nearby rays in parallel (as described in Section 4.2.2) apparently does not produce enough overlap in reference to yeild an appreciable proportion of cache hits. This is especially true as we sample nearby rays at greater distances from the source, where coherence in reference breaks down as rays diverge. Changing the volume's representation in memory to one which stores compact regions in contiguous memory blocks would likely improve cache performance; however, the access patterns would still not be sequential.

4.3 Voxel-projection

The motivation for the *voxel-projection* method of DRR production came from the need to speed the process of generating DRRs, to facilitate our objective function probing experiments, and to make our automated registration algorithm serviceably fast. When the best of such algorithms require hundreds of iterations to derive the best alignment, the full-resolution ray-casting's typical running time of 1+ minutes (even with nearest-neighbor interpolation) leads to tremendously slow registrations. The voxel projection algorithm reduces this running time by more than an order of magnitude by using the volume data to structure the projection process rather than the DRR data. Recall that the ray-casting algorithm first calculates the ray that terminates at each DRR pixel and then accesses the volume in a way inconsistent with the volume's structure in memory. The voxel-projection method instead projects each volume voxel onto the DRR, so that volume data access $\theta(m^3)$ is sequential in memory, and then does minimal processing (sometimes in data order) on the much smaller $\theta(n^2)$ DRR, which can fit entirely in the L2 cache (see Section 2.4.1). The optimized algorithm relies on the linearity of the ray-casting process, first to estimate and apply the contribution that each voxel makes to the DRR without actually casting rays through the volume; and second, to separate processing of the volume data and of the DRR data into two distinct stages, thus reducing the overall number of calculations and the amount of data accessed during each stage.

4.3.1 Development of the Method

Access the Volume in Data Order The voxel-projection method is probably best described as modeling the ray-casting method. As rays are cast through the volume, the amount of intensity that is contributed to each ray from a single voxel varies inversely with the distance between ray and voxel (irrespective of the interpolation method used). So, were we to render the amount of intensity that a single voxel contributes to each of the DRR pixels, the resulting image would contain a parallelogram or irregular hexagon with bowed sides (the projection of a rectangular solid), whose intensity is greatest at its center and falls of to zero at its boundaries (precisely how it falls off is determined by the specific interpolation method). Because each projective ray is just the sum of scaled intensities from every voxel, this contribution to the DRR from each voxel could be sequentially added to the DRR pixels, without affecting the resulting projection. By the way in which each voxel's projective footprint's intensity falls off from its center, the intensities of adjacent volume voxels would be effectively interpolated. Furthermore, if we assume that a linear method

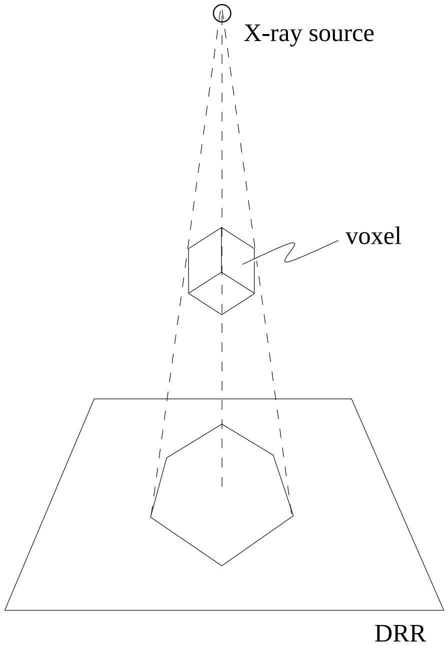

Figure 4-3: A Voxel's Projective Image (Nearest-Neighbor Interpolation)

of interpolation accurately gives inter-voxel intensity values, the contribution of each voxel could be considered independently of the surrounding voxels (nearest-neighbor and trilinear interpolation are both linear in the voxel intensities).

Therefore, if we could easily compute each voxel's projective footprint (or *voxel projection* as we will refer to it), we could go through the voxels in data order and project these footprints to the image in sequence. Unfortunately, this is not practical, for calculating the particular contribution of every voxel would require as much work as is done in ray-casting directly. To solve this problem, we can assume that all voxels have a projective footprint K of the same shape and extent, but with amplitudes that are scaled according to the intensity of each voxel. We will discuss the validity of this simplification, and where it breaks down, in section 4.3.2 but it can be generally justified by the fact that the focal length is long relative to volume dimensions, so that rays pass though all voxels in the roughly the same direction, and roughly diverge the same amount after passing through each voxel. Notwithstanding this justification, by assuming the propriety of this generic projective footprint K (or *kernel* as we will now refer to it), we may now strucutre the rendering of a DRR by the following algorithm:

> Starting with a blank DRR image (all zeros), for each volume voxel:
>
> 1. Find the DRR pixel to which the ray passing directly through the voxel center projects.
> 2. Use this pixel to position the generic projective kernel K which has been scaled in intensity according the intensity of the voxel.
> 3. Add the scaled intensities of the kernel K to spatially-corresponding DRR pixels.

Separating Volume Processing from DRR Processing Note that the above algorithm interleaves the processing of volume and DRR data, in that each voxel access is separated from the next by a significant amount of calculation on the DRR data. This may defeat some of the speed advantages derived from accessing the volume in data order. However, the above process is suggestively identical to that of applying a 2D linear, spatially-invariant filter K to an input comprising a 2D field of impulses (each of which is the sum of voxel intensities placed at the location of their respective voxel's center projection). In fact, due to the linearity of the projective process, the above procedure can be performed as follows, so that the primary processing on the DRR data occurs after volume access is complete.

1. Starting with a blank DRR image (all zeros), for each volume voxel:
   (a) Find the DRR pixel to which the ray passing directly through the voxel center projects.
   (b) Add the voxel's intensity to that pixel.
2. Filter the DRR image with generic projective kernel K.

Note that two voxels may project to the same pixel, but due to the linearity of process, the result of filtering is still valid. Also note that a voxel-center will rarely project to a *DRR* pixel location, but to some inter-pixel location. Due to the smoothing effect that we will later in this section, we can simply add the voxel-center projection to the nearest neighboring DRR pixel. Figures 4-4 and 4-5 depict the stages of DRR production by this method of voxel projection.

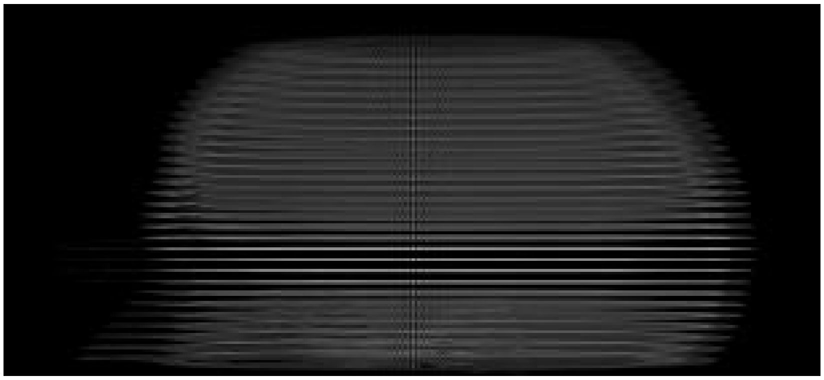

Figure 4-4: This image is the result of the first stage of the voxel-projection algorithm in which the intensities of voxel-centers are projected into the DRR. Here the CT volume shown in cross-section in Figure 2-3 is being projected.

Rejecting Large Kernels We might look to the method of ray-casting to help define the shape of the generic kernel K which has been left loosely specified up to this point. Indeed given the orientation of the volume and the manner of interpolation used, we can determine what region of influence an individual voxel has on the

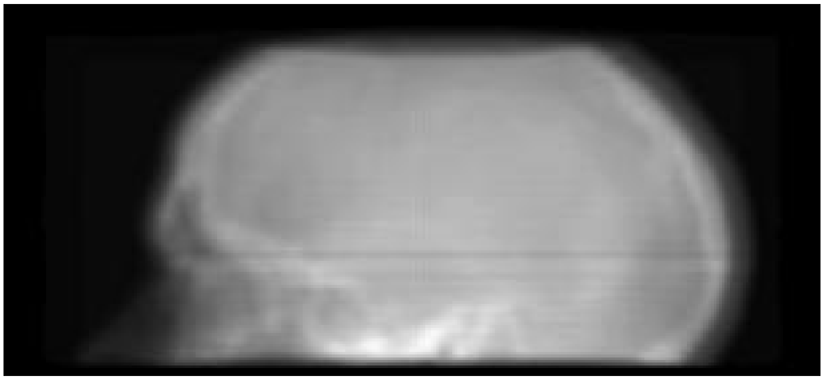

Figure 4-5: This figure shows the resulting DRR after the COT kernel is applied to the image shown in Figure 4-4

DRR. If we restrict our focus to linear methods of interpolation, we can do this by casting rays though a volume in which one voxel has unit intensity, while all others have zero. Because nearest neighbor interpolation simply assigns the voxel's intensity to the point of any ray passing within the rectilinear extent of the voxel, the region of influence rendered by this method using nearest-neighbor interpolation, is loosely bounded by the projections of the centers of the six adjacent voxels. Trilinear interpolation would gives a voxel a broader area of influence on the DRR because each voxel's intensity is mixed with the intensities of the twenty-six surrounding voxels. Both these regions of influence are depicited in Figure 4-6. While using such kernels would probably render images which look very similar to DRRs produced by ray-casting under these interpolation methods[2], the application of 2D kernels of these sizes was found to be impractically slow in light of this algorithm's potential place in the inner loop of an iterative registration procedure. This long running time is due both to the number of convolution multiplications and additions required, and to the fact that such filtering cannot be done in-place by direct convolution.[3]

[2] We did not attempt to do this.

[3] We did not explore in-place filtering in the Fourier domain, as it would probably take a considerable amount of time.

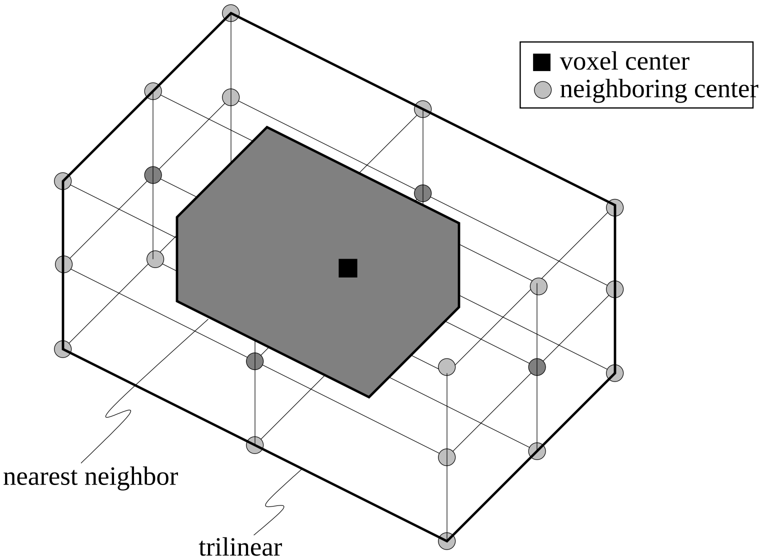

Figure 4-6: A Voxel's Span of Influence on the DRR

Fast Kernels In light of the slow performance of 2D kernels, we explored the application of a kernel which has the effect of interpolation which we'd expect by analogies we have drawn to ray-casting methods, but which also has the special property that it can be generated by the 2D convolution of two 1D, orthogonal functions h_1, h_2. Then, we can rely on the associativity of the convolution operation to turn the filtering of image I by a large 2D kernel K, into a sequence of convolutions by two small 1D kernels.

$$K * I = (h_2 * h_1) * I = h_2 * (h_1 * I) \tag{4.2}$$

This allows us to do the filtering in-place with a small buffer (of size equal to half the length of the 1D kernels) to hold the original image values for pixels which are modified by the application of each 1D kernel to nearby pixels. Two kernels which met these criteria were the 2D, rectangular pulse, and one formed from the 2D-convolution of orthogonal, 1D triangles (COT) as shown in Figure 4-8. Both these kernels have the additional advantage that they may be applied using only additions, with no convolutional multiplications. We will defer to Section 4.3.4, discussion on the implementation details of filtering with these kernels, as well as on the theoretical effect that they have. We now note some performance results which will be detailed later. The COT pulse was found to produce a very reasonable approximation to a true ray-cast DRR. For reasons we will also discuss in Section 4.3.4, the kernel must in general be oversized, which slightly blurs the resulting DRR, which appears to fall somewhere in between a ray-casted DRR produced using trilinear interpolation, and one produced using nearest-neighbor interpolation.

4.3.2 Variations Among Voxels' Projective Images

In our discussion above we mentioned the assumption that all voxels have similar enough shape and extent that we can use the same projective image for each in our projection. Here we will discuss that assumption and its limitations.

As the extent of imaged structures is generally small compared to the focal length of typical fluoroscopic apparatus (consider the cross-sectional extent of a body relative to a focal length of 1.5 meters), variations in the shape and extent of voxel projections due to the divergence of projective rays is not very pronouced.

More pronounced variation in the size of a volume's voxel projections are are a product of the the fact that voxels which are closer to the x-ray source will generally have a larger projective footprint than will voxels that are closer to the plane, as the rays which pass through them will subsume a larger conical angle and have a longer travel distance over which to diverge (see Figure 4-7). This effect is more pronounced for larger voxel spacing, shorter focal length, and greater volume extent in the direction of the projection. We found that the projection of a volumetric dataset's central voxel (i.e. with data coordinates $(\frac{columns}{2}, \frac{rows}{2}, \frac{slices}{2})$ serves as a good model for all voxels in the set, as it it's extent is roughly midway in between the extent of extremal voxels. This result was derived with experiment on head images (small spatial extent) with large slice spacing (4mm), so we anticipate that it will be a generally-applicable one.[4]

[4] We have not explored the projection of data of so large an extent and slice spacing that this is not a good approximation. It might involve a simple extension of the algorithm described here,

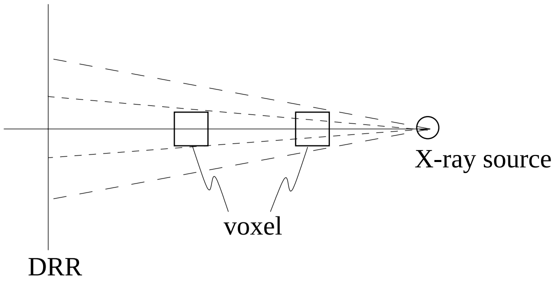

Figure 4-7: Variation in the extent of voxel projections.

4.3.3 2D Rectangular Pulse

At the outset of the chapter, we mentioned two 2D kernels which have the property that they can be produced by the 2D convolution of two 1D kernels. One of these kernels was the 2D rectangular, constant pulse. It is rectangular in two orthogonal directions, and when properly-sized, seems to approximate the projective kernel due to ray-casting with nearest-neighbor interpolation. Its unscaled amplitude is unity, so that it contributes to the DRR exactly the intensity of the voxel with which it is associated. This quality affords the additional computational advantage that convolving with it does not require any multiplication.

In practice this kernel does not produce suitable approximations to a ray-casted DRR. This failure is apparently due to the fact that our filtering approach implicity uses the extent of a single voxel's projective image for all the voxels. If a voxel with a large projective image is used as the model voxel, then the DRR suffers from high-intensity banding due to excessive kernel overlap among voxels whose projective image is smaller. If a voxel with a small projective image is used as the model voxel, then the DRR suffer from low-intensity banding due to insufficent kernel overlap among voxels whose projective image is larger. Unfortunately, there does not appear a broad enough middle ground that a model voxel may be selected which will work over all

whereby multiple passes are made over the volume, and initial projections are made to more than one DRR data set. While performance would be compromised due to the additional range checking operations required, we could probably minimize processing time at the image level by interleaving the pixels of the DRRs in memory (dividing each block of memory into segments that hold the intensity of corresponding pixels of different images).

volume orientations. Smoothing the edges of the rectangular kernel did not seem to solve this problem, although perhaps there is room for further investigation.

4.3.4 2D Convolution of Orthogonal 1D Triangles (COT)

Performance Overview As introduced in Section 4.3.1, another kernel that satisfies the requirement of being composed of two 1D orthogonal kernels is one that is composed by the convolution of two orthogonal triangular kernels. The COT kernel is shown from a number of viewpoints in Figure 4-8. When properly-sized, this kernel performs a linear interpolation which is similar to that performed by a proper trilinear-interpolation kernel, although in general it is very crude approximation. Nevertheless, it has proved to produce a very reasonable approximation to a true ray-casted DRR over all volume orientations, when we used the projective image of the volume's central voxel as a model for all voxels'. For reasons that we will discuss shortly, its extent must generally be overestimated, which slighly blurs the DRR. The result is a DRR which appears to fall somewhere in-between a ray-casted DR-R produced using trilinear interpolation, and one produced using nearest-neighbor interpolation. While these COT-kernel DRRs have slight high-intensity banding in certain orientations due to the same kernel-extent overestimation which foiled the

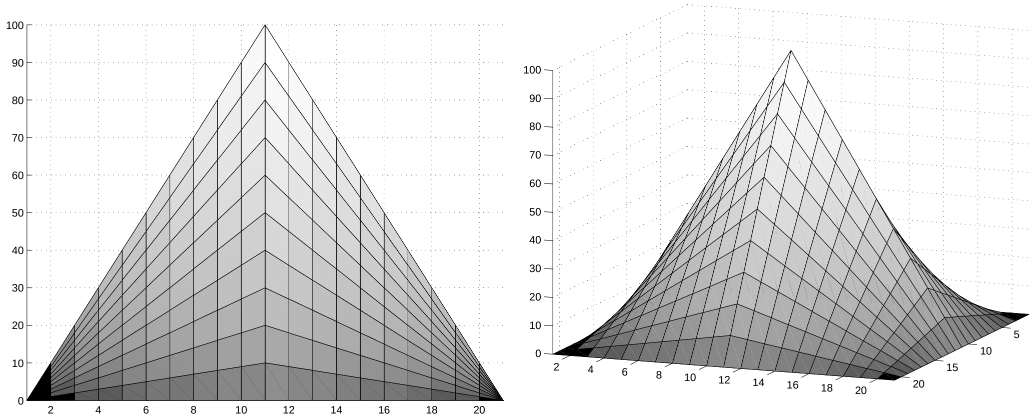

Figure 4-8: The COT kernel. This plot was produced by the outer product $A'A$, where $A = (0, 1, ..., 9, 10, 9, ..1, 0)$

2d rectangular kernel, the blurring effect of the pyramid kernel seems to minimize sensitivity to differences in the extent of voxels' projective footprints.

The COT has the same additional computational advantage as the 2D rectangular pulse kernel in that filtering with it does not require any multiplications. This is due to the fact that each of the 1D triangular, orthogonal kernels from which the pyramid is produced, can themselves be produced by the convolution of two 1D, rectangular pulses. Therefore, the COT can be applied by sequentially applying two passes of unit amplitude, 1D, rectangular pulses in each orthogonal direction. [5] We'll discuss the details of this application in a later section, but we'll find that we can double-sweep each orthogonal 1D rectangular pulse with a single pass over the DRR (one of which occurs in data order).

Relation to Trilinear Interpolation In particular volume orientations, like those depicted in Figure 4-9, the action of the COT is most similar to that of trilinear interpolation in ray-casting, when the kernel's extent is determined by the projected distance between adjacent voxels. In that figure, each application of the COT kernel to the voxel-center projections (as part of the convolutional filtering process) is depicted as hovering with its associated voxel. The voxel-projection algorithm models the projection process as casting each of these kernels down onto the projective plane. We can also model the process occurring by ray-casting, where each ray picks up units of intensity equal to the amplitude of the part of each hovering kernel through which it passes (as if samples we taken at exactly these points). With the volumes oriented as they are in the figures, each ray receives an intensity contribution equivalent to that of one voxel (in fractions of intensities of multiple voxels) over each step of pixel size in its direction. This is exactly like ray-casting with trilinear-interpolation, where the fraction of a voxel's intensity contributed to a sample taken in the same data plane (i.e. trilinear interpolation reduces to bilinear interpolation) is given by the product of two terms which are functions of its linear distance from the sample in two orthogonal

[5] Note that this results in a scaling of the central amplitude of each triangle of roughly half the triangle width (see Figure 4-8). The scaling is uniform, so this does not present a problem.

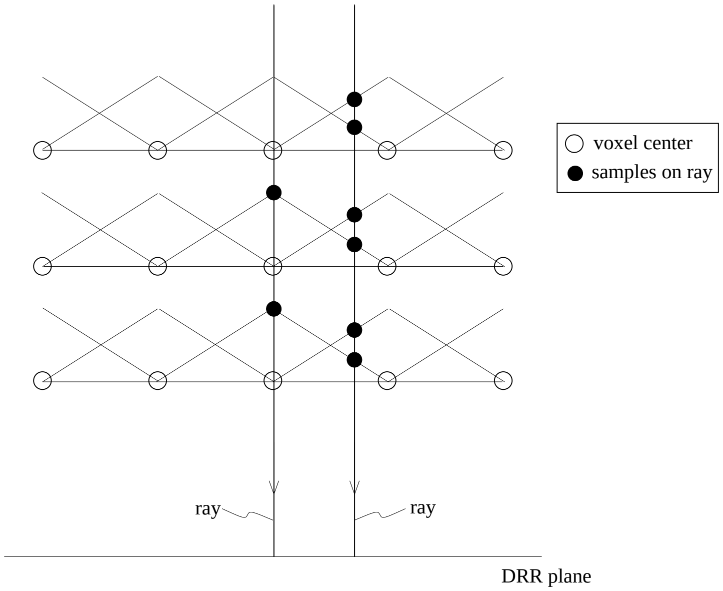

Figure 4-9: A volume orientations where smoothing by the COT kernel is similar to ray-casting using trilinear interpolation.

directions (see Equation 2.4). The COT kernel has the same form, which is suggested by the fact that it is triangular (linear) in two orthogonal directions. Indeed, the convolution by which the COT kernel can be produced is also the outer product of a vector with itself whose terms give the amplitudes of a triangle (this is exactly how the rendering in Figure 4-8 was produced).

However, the analogy soon breaks down when we consider more skewed volume orientations (see Figure 4-10).

Sizing the COT Kernel Instead trying to tie the action of the COT kernel to that of trilinear interpolation, we allow that it is performing a different kind of interpolation, one which is also reasonable and empirically effective. This action can be most accurately described as smoothing among the intensities of adjacent volume voxels (in the 6-connected sense). This kind of smoothing is motivated not only by its functional similarity to trilinear and nearest-neighbor interpolation kernels, but also out of a need to regularize the resulting DRR. We can see this as follows. For a typical volume orientation, the projective image of the centers of voxels adjacent

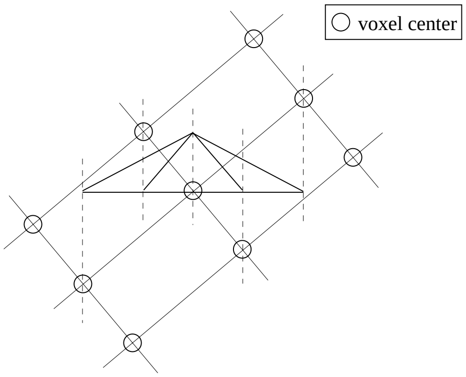

Figure 4-10: The proper COT width is not clear in this orientation.

to a central voxel are depicted in Figure 4-11. We can imagine how such a structure might generate the image in Figure 4-4 by imagining this structure being duplicated along the projections of volume data axes. We also notice that if the extent of the COT filter does not encompass the longest of these projected inter-voxel lengths, then the kernel will not properly interpolate the central voxel's value with all of its six nearest neighbors, and the low intensity rows characteristic of the image in Figure 4-4 will not be fully closed. Therefore, it seems that the smoothing kernel must at least interpolate among adjacent voxels, lest an irregular DRR result.

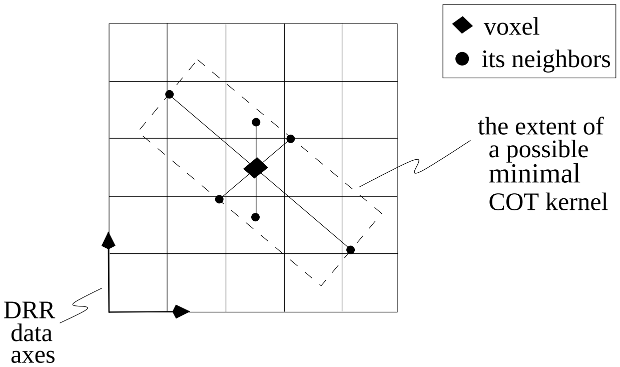

Figure 4-11: Oversizing the COT Kernel

With this in mind, we could take the COT to have minimal extent while still encompassing the projection of each of the central voxel's six neighbors, were it not for the fact that the volume's data axes (and thus the projection of these volume centers) will not generally be aligned with the data axes of the DRR and of the kernel which will be used to smooth the DRR (see Figure 4-11). To derive the computational advantage that motivated this choice of kernel, the axes of the pyramid must be aligned with those of the DRR. It turns out that we can solve this problem simply by oversizing the kernel to be isotropic with sides equal to the length of the maximum projected inter-voxel spacing. While this generally results in oversmoothing of voxel intensities (and more so in certain directions than in others), the resulting DRR is regardless a good approximation of one produced by ray-casting.

4.3.5 Voxel-projection Parameters

The voxel-projection algorithm is governed by all the same parameters that govern ray-casting projections except the projective-sum step size. The relative resolutions of the volume and DRR, the orientation of the volume, and the volume position along the focal length, all influence the extent of a derived parameter: the extent of the square COT kernel applied in the second stage of the voxel-projection process.

4.3.6 The Voxel-projection Algorithm

This section presents a detailed outline of the voxel-projection algorithm. It is followed by implementation details by which the method is optimized for speed.

1. Crop the volume to exclude blank space (below some threshold intensity) which may surround the object imaged.
2. Set $\mathbf{x}_s$ as the x-ray source according to the focal length f.
3. Initialize all DRR pixels to zero intensity.
4. For each volume voxel $\mathbf{x}_v$:
   (a) Find the DRR pixel $\mathbf{x}_p$ with integer coordinates, nearest to the point where the ray $\mathbf{r} = (\mathbf{x}_v - \mathbf{x}_s)$ intersects the projective plane.
   (b) Increment pixel $\mathbf{x}_p$ with the volume intensity at voxel $\mathbf{x}_v$, VOLUME($\mathbf{x}_v$).
5. Find the DRR data coordinates of the projections of the volume's central voxel (i.e. with volume data coordinates $(\frac{columns}{2}, \frac{rows}{2}, \frac{slices}{2})$) and three adjacent voxels, each of whose data coordinates differ from the central voxels by unity in a different dimension. Find the maximum distance d_{max} in DRR pixels between the central voxel's projection and one of its adjacent voxel's projection.

6. Set $(2\lceil d_{max} \rceil + 1)$ as the length and width of the rectangular-pyramidal projective kernel K. This guarantees the kernel will always have a center point with unit amplitude to which the full intensity of the associated voxel can be applied.

7. Smooth the DRR with the kernel K.

Step 4 In this step, the volume voxels are actually projected to the DRR. This interaction between plane and volume requires application of the affine transform between the volume and DRR data coordinates. Naturally, we'll index the loop in volume data coordinates, for it rasters through all the volume's voxels; however, at some point in each loop iteration, a conversion must be made to DRR data coordinates so that the DRR pixel to which the current voxel projects, can be incremented. We can drastically reduce the number of operations in each loop iteration by making this conversion at the outset of each iteration.

First, this allows us to do all the projection calculations in DRR data coordinates, which have a simplified form due to the plane of projection being defined by two of the coordinate system's basis vectors. For a general coordinate system, the projection is calculated:

$$\begin{aligned}
\mathbf{x}_v &= \text{voxel coordinates} \\
\mathbf{x}_s &= \text{x-ray source coordinates} \\
\mathbf{x}_p &= \text{projection point coordinates} \\
\mathbf{n}_p &= \text{projection plane unit normal} \\
\mathbf{x}_0 &= \text{plane offset}
\end{aligned}$$

$$\mathbf{x}_p = \mathbf{x}_s + (\mathbf{x}_v - \mathbf{x}_s)\frac{\mathbf{n}_p^T(\mathbf{x}_0 - \mathbf{x}_s)}{\mathbf{n}_p^T(\mathbf{x}_v - \mathbf{x}_s)} \tag{4.3}$$

But, in projection data coordinates, where the plane resides in the coordinate system's xy-plane, we get a simplified expression:

$$\begin{aligned}\mathbf{n}_p &= (0,0,1)\\ \mathbf{x}_0 &= (\text{center-col}, \text{center-row}, 0)\\ \mathbf{x}_s &= \mathbf{x}_0 + f\mathbf{n}_p\end{aligned}$$

$$\mathbf{x}_p = \mathbf{x}_s + (\mathbf{x}_v - \mathbf{x}_s)\frac{z(\mathbf{x}_0 - \mathbf{x}_s)}{z(\mathbf{x}_v - \mathbf{x}_s)} = \mathbf{x}_s + (\mathbf{x}_v - \mathbf{x}_s)\frac{-f}{z(\mathbf{x}_v - \mathbf{x}_s)} \tag{4.4}$$

where $z(\mathbf{x})$ is the z-coordinate of $\mathbf{x}$. This simplification obviates two inner products and saves 6 multiplications and 4 additions.

Second, we can take advantage of the linearity of the affine coordinate transform, and the sequence of voxel coordinates on which we are applying it, to extrapolate the DRR data coordinates of *every* volume voxel by applying the full affine transformation *only once*. Note that the loop rasters through the volume voxel in data order, so that when on the previous iteration it accessed the voxel with volume data coordinates $\mathbf{x}_{i-1}$, on the current iteration it accesses the voxel with volume data coordinates $\mathbf{x}_i = \mathbf{x}_{i-1} + \mathbf{e_i}$, where $\mathbf{e_i}$ is one of the basis vectors of the volume data coordinate system, $\{(1,0,0),(0,1,0),(0,0,1)\}$. This leads to the following simplification:

$$\begin{aligned}T_{affine}(\mathbf{x}_i) &= A(\mathbf{x}_i) + \mathbf{b} = A(\mathbf{x}_{i-1} + \mathbf{e}_i) + \mathbf{b} = (A(\mathbf{x}_{i-1}) + \mathbf{b}) + A(\mathbf{e}_i)\\ &= T_{affine}(\mathbf{x}_{i-1}) + (\text{column } i \text{ of } A) \end{aligned}\tag{4.5}$$

Therefore, we can transform each voxel's volume data coordinates to its DRR data coordinates simply by adding the appropriate column of the coordinate transform matrix (3 additions) to the preceeding voxel's DRR data coordinates; instead of applying the full affine coordinate transformation (9 multiplies, 9 additions). This requires very little additional bookkeeping to transition between the rows and slices of the volume dataset (assuming that the rows of the volume data are organized in contiguous blocks of memory). The resulting loop requires only two additional vectors to hold the beginning of the current volume slice and row, and is outlined as follows.

$$
\begin{aligned}
\mathbf{x}_s &= \text{volume data coordinates of the beginning of the current slice} \\
\mathbf{x}_r &= \text{volume data coordinates beginning of the current row within the current slice} \\
\mathbf{x}_c &= \text{volume data coordinates the current voxel (column, row, slice)} \\
T_{affine} &= A(\mathbf{x}) + \mathbf{b} = \text{transform from volume to DRR data coordinates} \\
A &= [\mathbf{e}_c, \mathbf{e}_r, \mathbf{e}_s], \text{so that } \mathbf{e}_i \text{ are the columns of } A
\end{aligned}
$$

```
Initialize $\mathbf{x}_c = \mathbf{x}_r = \mathbf{x}_s = (0, 0, 0)$ (the first voxel).

For all volume slices {
    For all volume rows {
        For all volume columns {
            Project voxel $\mathbf{x}_c$.
            Update $\mathbf{x}_c = \mathbf{x}_c + \mathbf{e}_c$. }
        Update $\mathbf{x}_c = \mathbf{x}_r = \mathbf{x}_r + \mathbf{e}_r$. }
    Update $\mathbf{x}_c = \mathbf{x}_r = \mathbf{x}_s = \mathbf{x}_s + \mathbf{e}_s$. }
```

The time overhead required for data access in this loop is relatively small because volume intensities are referenced sequentially, and because DRR access most likely exhibits some locality of reference, and probably fits entirely in the L2 cache. As a result, these reductions in the number of operations per iteration yield sizable time savings.

Step 7 In this step, the rectangular pyramidal kernel is applied to voxel-center projections currently occupying the DRR. As mentioned before, this kernel can be applied (almost) in-place and using only additions. To do it (almost) in-place, the smoothing is implemented using the fact that the rectangular pyramidal kernel is the 2d convolution of two orthogonal, 1d triangular kernels. By the nature of the convolution operation, when the kernel is applied to each DRR pixel, it is scaled by that pixel's value, and the values of surrounding pixels are accordingly incremented. This operation cannot be done in-place directly: The kernel must be applied to each pixel's original value, but applying the kernel to one pixel requires a number of surrounding pixel values to be modified. For an arbitrary 2d kernel of area p^2, the original values of about $(n\frac{p^2}{2})$ of image pixels, where n is the image width, would have to be saved at any one time. To see this, imagine a 2d kernel applied to an image by sweeping it along the rows (Figure 4-12). For typical, full-resolution DRR widths of $n = 250$, this can be a rather large buffer. However, because the pyramidal kernel can be applied by independent, sequential application of 1d kernels, the number of original image values to save can be much smaller, about $\frac{p}{2}$, where p is the number of kernel samples. Such a kernel would be applied by sweeping it along the rows (or columns) of the image, so only a small buffer would be required to hold the original values of the DRR pixels that lie half-a-kernel-radius ahead of the kernel's movement (Figure 4-13). To apply each of these 1d trianglar kernels without using any multiplications, we use the fact that a triangular kernel is the convolution of two constant pulses (Figure 4-14). Note that one of these rectangles trails the input impulse to which it is applied, and one extends to indices greater than the central index. We can use this fact, along with the fact that we would apply these 1d pulses by sweeping along DRR rows (columns), to apply the two rectangular kernels in a single pass. If the rectangular kernel which extends to positive indices is applied to the DRR first, then as soon as it has been applied to a pixel, that pixel has its final value and is ready to have the trailing pulse applied to it. But note that because we are applying these kernel by sweeping in order of increasing index, and because the trailing kernel extend to indices with values less than the pixel to which it is being applied, we need no buffer to hold any original pixel values when applying this kernel. Therefore, the trailing pulse can be swept along directly behind the first pulse, and an original-value buffer is only required over the extent of the first pulse (Figure 4-15).

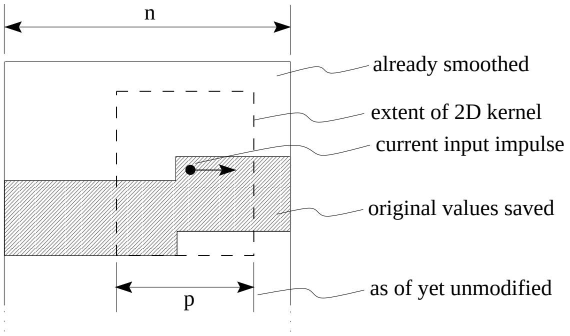

Figure 4-12: Direct convolution by a 2D kernel cannot be done in-place.

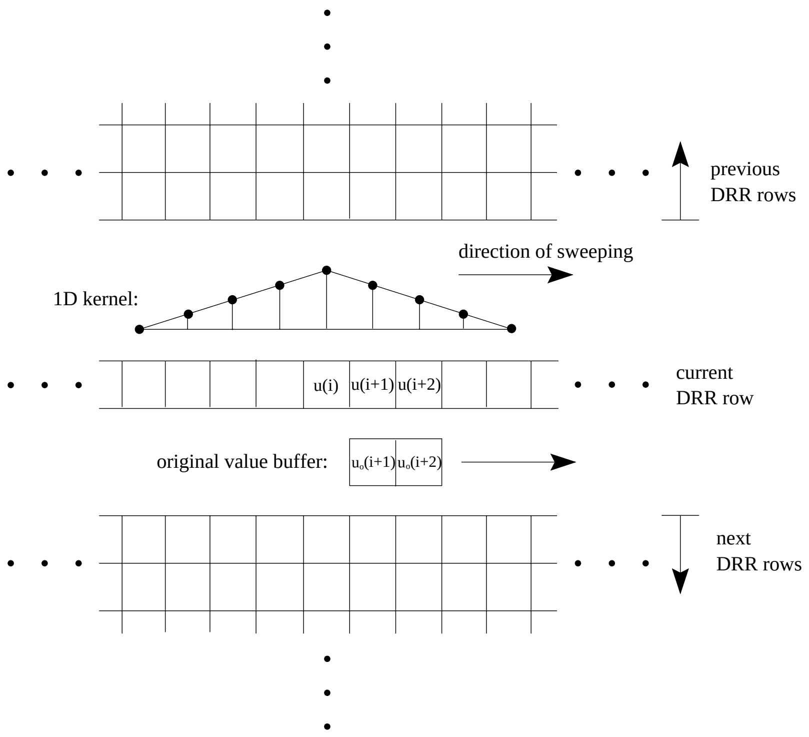

Figure 4-13: Applying a Triangular Kernel.

When the 1d triangle is being applied along the DRR axis which parallels the DRR's data order, this double-rectangle sweeping results in nearly-sequential memory access (and exhibits some locality of reference in general). We can use this technique to apply the kernel only to the projections of voxel centers (non-zero DRR pixels) which are sparce in the image at this stage (refer back to Figure 4-4). Due to the linearity of the convolution operation, ignoring input pixels with zero intensity has no effect on the resulting image, but speeds things up considerably. Indeed, not only do voxel-center projections tend to produce an image with many blank regions, there is often a significant amount of blank space surrounding the projected extent of the object rendered in the volumetric data. This is especially true for head data, and volume orientations in which the volume's data axes are highly skewed relative to the plane of projection.

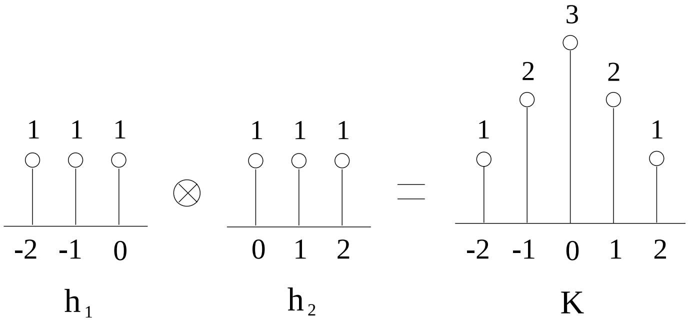

Figure 4-14: A triangle is the convolution of two square pulses.

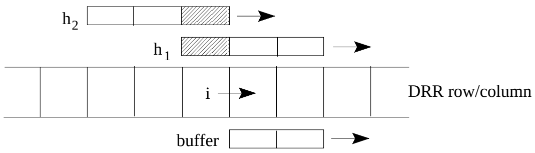

Figure 4-15: Double-sweeping a rectangle to produce a trilangle

4.3.7 Voxel-projection Complexity

The computational complexity of voxel-projection is given by

$$\theta(m^3 + n^2 * s^2) \tag{4.6}$$

where

$$\begin{aligned} n &= \sqrt{\text{DRR pixels}} \\ m &= \sqrt[3]{\text{volume voxels}} \\ s &= \sqrt{\text{kernel pixels}} \end{aligned}$$

Again, the image size n^2 is roughly proportional to m^2, the extent of the volume's projective image. The relationship between s and m is related by the relative resolutions of the volume and DRR. It certainly varies in proportion to the image size (assuming we're increasing resolution for a fixed metric dimension), in which case the term $\theta(n^2 * s^2) = \theta(m^4)$. On the other hand, if the relative resolutions of the volume and DRR are held constant, the overall complexity is $\theta(m^3)$, just like that of ray-casting. Of course, the constant coefficients implicit in these asymptotic expressions are small due to voxel-projection's efficient memory access patterns.

4.3.8 Voxel-projection Resolution

The resolution of a voxel-projected DRR is limited by the extent of the projective (smoothing) kernel applied to the DRR. Because this kernel has roughly the dimension of the maximum projected inter-voxel spacing, we can estimate a voxel-projected DRR to have that resolution (assuming the DRR's pixel resolution is not the limiting factor). Because of the divergence in projective rays, the resolution of voxel-projected DRRs is smaller than that of ray-casted DRRs produced using trilinear interpolation (where the resolution is governed by the unprojected inter-voxel spacing, assuming sufficiently small DRR pixel size). The effects on DRR resolution due to nearest-neighbor interpolation are less transparent, although an argument could be made that ray-casted DRRs produced with nearest-neighbor have an ansiotropic resolution equivalent to the projected inter-voxel spacings.

The disparity in resolution between the two projection methods begs investigation the relative running-times of these projective methods when the resulting DRR resolutions are roughly matched (especially because we claim that voxel-projection is so much faster). Instead of trying to normalize for resolution between these two methods which incorporate information into a DRR in different ways, we direct our attention to the running time comparisons of each method (Section 4.4), where we can clearly see that the voxel projection method affords significant improvements in speed over ray-casting over all typical pixel resolutions.

4.4 Running-time Comparisons

In the following figures, we compare the running-times of voxel-projection and ray-casting algorithms. These experiments are geared to show variation with DRR size and volume size. The graphs plot running times against DRR pixel width, which varies as the square root of the DRR size. We varied the DRR pixel width in a schedule (0.5, 1, 2, 4, 8), each step quartering the DRR size. Voxel-projection is considerably faster than either ray-casting formulation for full and half resolution DRRs; however, although the computational burden of ray-casting declines as fewer rays are cast to a smaller DRR, the computational burden of the voxel-projection alogrithm is always limited by the fact that it covers the entire volume, irrespective of the DRR resolution, even when the width of its smoothing kernel goes to 1.

These experiments were performed on a single CT head image of size 512×512×29, with 8-bit pixels. In the first set of experiments (on the left in Figure 4-16), this image was cropped tight to the head, yielding an image size of 317 × 390 × 29. In the second set of experiments, the original CT was downsampled by a factor of 2 to size 256 × 256 × 29, and then cropped around the head to size 154 × 191 × 29, roughly a quarter of the original image's cropped size.

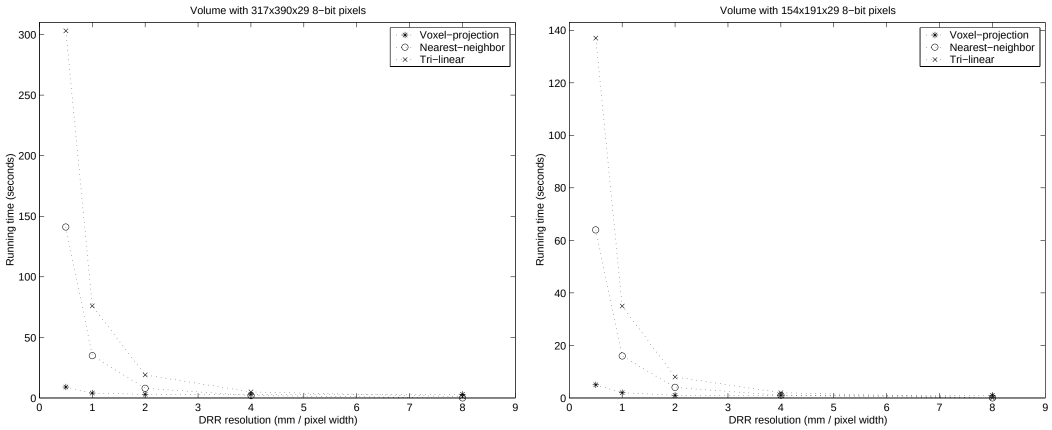

Figure 4-16: Running time comparision of voxel-projection and ray-casting over a series of DRR and volume sizes.

4.5 Summary

In this chapter we have presented the voxel-projection method of DRR production which is significantly faster than the more straightforward ray-casting method. We

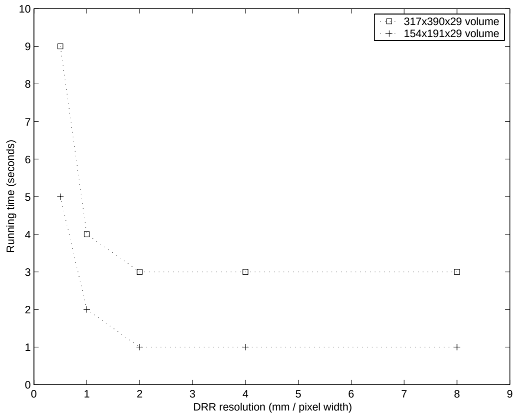

Figure 4-17: Voxel-projection running times across DRR and volume sizes.

have also discussed the limitation of the method in terms of the resolution of resulting DRRs and in terms of the similifying assumptions made in its development.

Chapter 5

Probing Objective Functions

5.1 Overview

In this thesis, we are interested in the intensity-based, rigid registration of volumetric MR to biplanar fluoroscopy. As we discussed in Chapter 3, intensity-based methods require an appropriate objective function and manner of evaluating it, as well as an optimization scheme which can efficiently optimize that objective function. For this reason, we performed a series of experiments to evaluate qualities of the objective functions described in Section 3.6.2, that are relevant to their utility in such a scenario.

In these experiments, objective functions were evaluated on DRRs from CT and MR images of the same anatomical region (the head). Biplanar CT-DRRs were produced as models for true radiographs. Relative to these models, the MR volume was stepped over a range of misregistrations, and at each misregistration, an MR-DRR was produced on which our series of objective functions was evaluated. By this manner of probing, we could discern and visualize the behavior of the objective functions.

The approach of using a CT-DRR as a model for a fluoroscopic radiograph has the advantage that "ground truth" is known for the registrations, because the CT and MR can be registered by existing, validated methods for 3D-3D registration [22]. Furthermore, using CT-DRRs allowed us to avoid the calibration and dewarping issues involved in using fluoroscopic data, that would have complicated our investigation.

We reason that the performance of objective functions in registering CT-DRR to MR is indicative of their performance in registering radiographs to MR, due to the great similarity of CT-DRRs and true radiographs as described in Chapter 3.6.1.

In this chapter, we will describe the data on which our experiments were performed, the manner in which we evaluated the objective functions, the qualities of the functions we considered, and the type of misregistrations we considered. We'll conclude with the results of these investigations and discussion of how these results lead to the choice of optimization algorithm.

5.2 Input Data

For our investigations we used CT and MR head data from the Vanderbilt *Retrospective Registration Evaluation Project* [3]. These data include CT and Proton Density MR, which were imaged with fiducials adhered. The fiducials were subsequently erased from the images, and versions of the MR images were made available to which geometric-distortion correction techniques had been applied.

The images that we used are cropped around the brain, as is depicted in Figures 2-3 and 2-2. The CT data contain $512 \times 512 \times 29$ 8-bit voxels of dimension $0.65 \times 0.65 \times 4$ mm^3. The MR data contain $256 \times 256 \times 24$ 8-bit, Protein Density voxels of dimenstion $1.25 \times 1.25 \times 4$ mm^3.

5.3 Objective Function Evaluation

We fixed the MR-DRR dimension of the model CT-DRR so that their pixels were perfectly aligned. We then evaluateed the objective functions over all pixels of both images, each intensity of which was given by an 8-bit integer.

We initially investigated the sampled evaluation of mutual information and correlation coefficient, whereby pixels (50-200 count) were chosen at uniformly random positions in the model radiograph. For mutual information, a histogram was estimated using Parzen Windows, and evaluation was performed in the manner described in

[22]. We found that these functions had shallow peaks often of amplitude on the same order as the very high-frequency fluctuations due to randomization. Furthermore, in evaluating rotational purturbations, we found that some of the objective functions investigated here (specifically mutual information and correlation coefficient, which were the only measure investigated at that time) did not have their global maximum at perfect alignment.

With these results, we decided on evaluation over the full images. While this is less computationally efficient, the performance of an objective function evaluated over the entire images is likely indicative of its performance over all evaluation schemes, and therefore, appropriate for our investigations in this thesis.

5.4 DRR production

The biplanar, CT-DRR models were produced by the casting rays through the CT data set, as decribed in Chapter 4. Trilinear interpolation was used to calculate intervoxel intensities in this process, as it is relatively efficient compared to higher-order interpolation methods, and at the same time, produces DRRs of good quality.

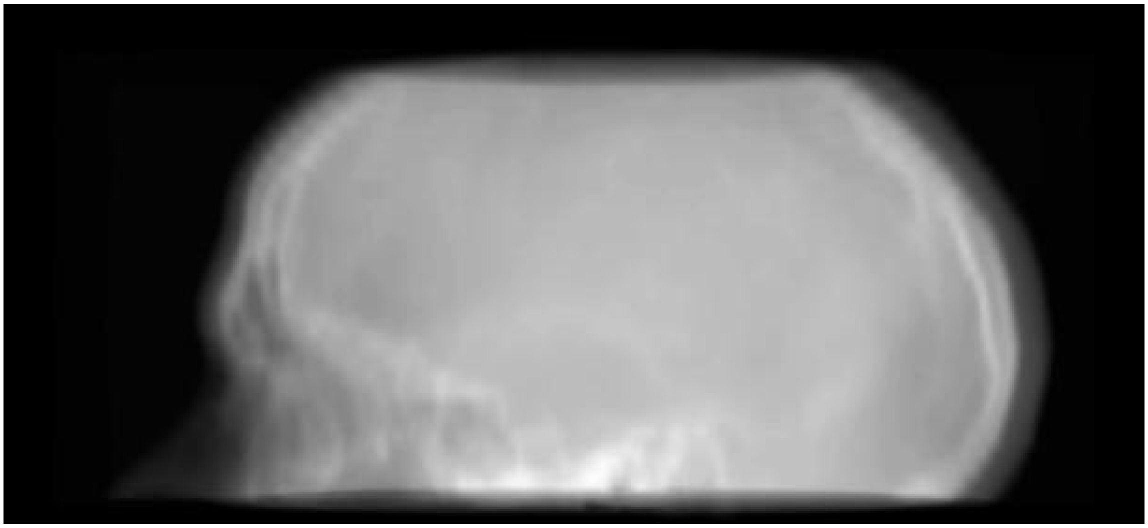

Figure 5-1: Sagittal CT-DRR produced by ray-casting with trilinear interpolation

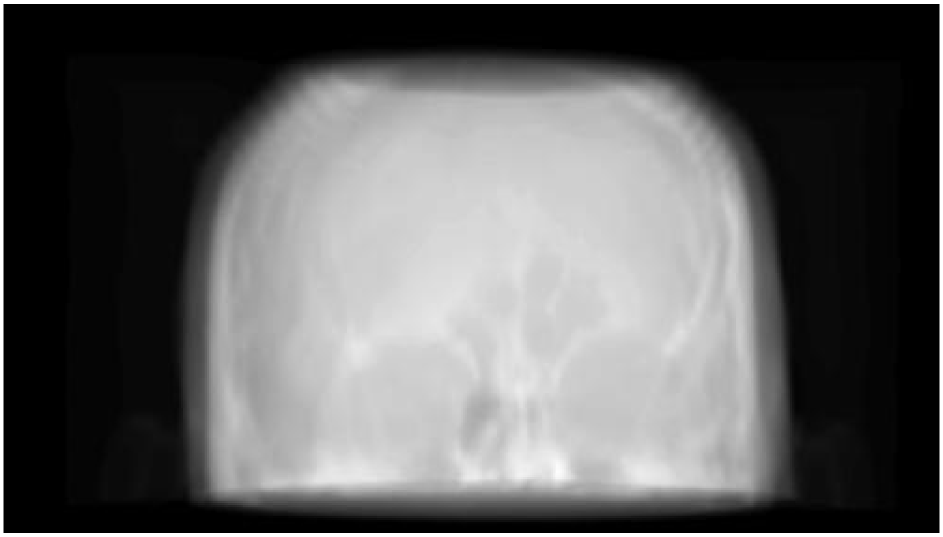

Figure 5-2: Coronal CT-DRR produced by ray-casting with trilinear interpolation

5.5 Pertinent Objective Function Qualities

Given the long processing times (1+ minutes) required for even streamlined version of ray-casting DRR-production algorithms, it was impractical to produce the MR-DRRs by this method for our probing experiments, when large numbers of these DRRs needed to be produced with the MR volume in different orienations. Therefore, we used the much faster *voxel-projection* algorithm to produce all MR-DRRs in our probing experiments. The details of the algorithm are also given in Chapter 4.

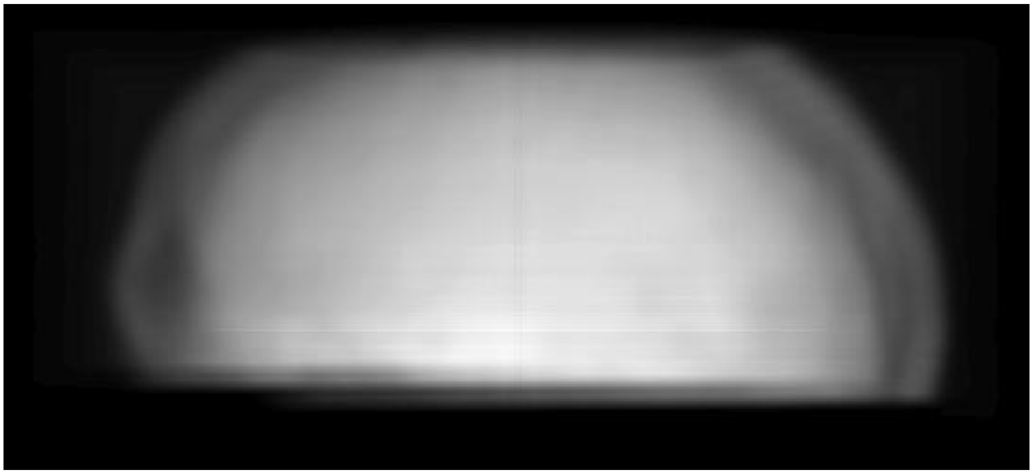

Figure 5-3: Sagittal MR-DRR produced by voxel-projection

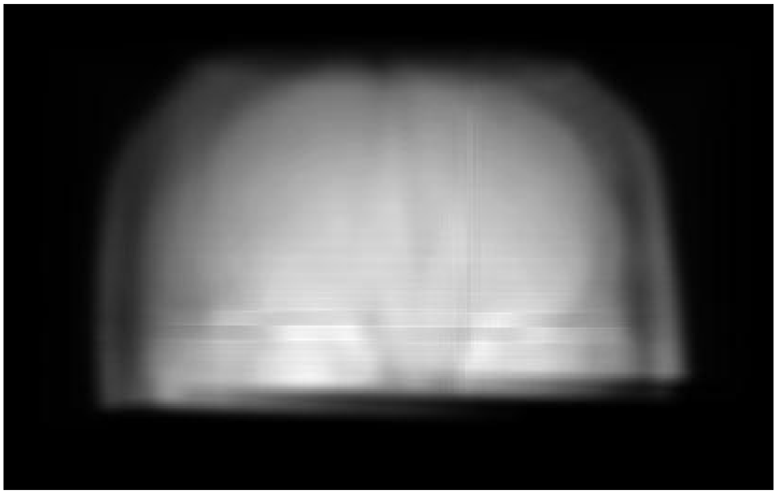

Figure 5-4: Coronal MR-DRR produced by voxel-projection

5.6 Pertinent Objective Function Qualities

5.6.1 Shape

As we discussed in Chapter 3, a successful objective function is one which takes on maximal value at ideal alignment, and which has characteristics such that it can be efficiently maximized by automatic optimization. These characteristic include having few local maxima and exhibiting local variation which is indicative of position of the global maximum.

5.6.2 Behavior at Different Resolutions

The shape and location of the global maximum of an objective function may change when the function is evaluated over a lower resolution version of the same dataset. When registration algorithms often use multi-resolution approaches to speed registration time and to overcome local extrema, it is important that objective functions are well-behaved at different resolutions, that is, new local maxima do not appear and the global maximum does not shift very much from its full-resolution position.

5.7 Transformation Parameterization

Without going into too much detail here about the formal parameterization of the transformation T relating the metric coordinate frames of the MR volume and the model radiographs (as we do in Chapter 6), we will give a qualitative depiction of the manner in which we adjusted the MR volume's postion in the course of our probing experiments. Refer back to Figure 3-1 where the volumetric data is positioned above the plane. In our experiments, we moved the MR volume over the projective plane by simple translations of its position, and by rotations about its central voxel.

Given this formulation, there are a set of types of misalignment over which we can probe, that well characterize an objective function's general behavior. The projective context and the fact that it only affords a two-dimensional view of variations in the projected volume's orientation, naturally divides translational and rotational misalignments into two categories: *in-plane* and *out-of-plane*. These two categories relate directly to how pronounced in the projective plane, a type of variation in the volume's orientation is. For translational displacements, it is easy to see that volume movement parallel to the projective plane will be most pronounced. Movement in the direction perpendicular to the plane (i.e. in the direction of the projection) will be less pronounced and limited to an expansion/contraction of the projective image due to the divergence of the projective rays. Note that there would be no such variation if these rays were parallel. For rotation, the in-plane mode comes from rotation around the axis perpendicular to the projective plane, and causes a pronounced spinning of the projective image in the projective plane. Rotations around axes parallel to the plane of projection might be considered out-of-plane modes as they cause variations in the projection that occur in the dimension perpendicular to the plane; these are rotations that cause different parts of the volume to overlap each other in the direction of the projection, so while their effect may be very clear, its mechanism are not so transparent from observing variation in the projection alone.

5.8 Range of Probing

Doing a fine probing of any appreciable volume of the six-dimension domain of the objective function takes an overwhelming amount of time due the number of samples being taken and the amount of time it takes to generate a function value for each of those samples. Furthermore, because we are limited to effective visualization of function of no more than two variables, probing over a six-dimensional space is also not of great utility. Therefore, we limited our probing to one or two parameters over ranges near perfect alignment, to get a sense for the function shape for independent modes of displacement and for modes in unison. We probed over broad ranges with low sampling resolution, and near the maximum of functions with higher resolution.

5.9 Results and Discussion

Before we present the results, we will first note some important bounds on our investigations and the conclusions that we can draw from them.

Due to limitations in the availability of data, our probing experiments were performed on single pair of cooresponding CT and MR head images from the Vanderbilt datasets described in Section 5.2. Besides this representing only a single data point of, the nature of these data restricts our study in a number of ways. First, these data are cropped around the brain cavity (tighter in the MR then in the CT, as can be seen in Figures 2-2 and 2-3. It is common that medical data is aquired in this way, as imaging is usually limited to the anatomical region of focus in a procedure. However, this presents a complication in this radiographic context, as any significant degree of out-of-plane rotation will cause a significant number of the simulated rays to pass through regions where material has been truncated from the image, and therefore the DRR pixels at the ends of these rays will not accurately reflect the attentuation that would have been shown has the cropped material been there (Section 3.6.1).

This severely limits our ability to explore out-of-plane rotations, which is the type of motion which most distiguishes this 3D-2D problem from a 2D-2D problem.

Nevertheless, we will offer a qualitative review of a variety of the objective functions noted in Chapter 3 as a starting point for further investigations.

5.9.1 Directions of Misregistration

We performed our probing experiments with a biplane model in which the planes of projection are orthogonal (as is common in biplane fluoroscopy apparatuses) and produce sagittal and coronal DRR images. We probed over misregistrations which center on the "ideal" alignment calculated by 3D-3D registration of the CT and MR volumetric data. The direction of probing were defined by the major axes of the sagittal plane, and involved translations in these directions, and rotations around them. The coordinate system is depicted in Figure 5-5

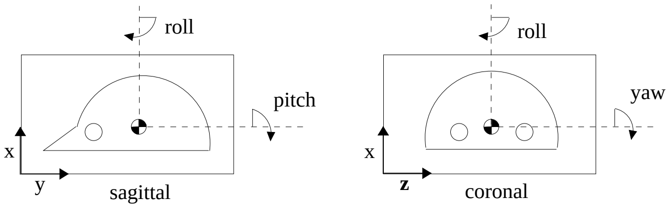

Figure 5-5: Probing Directions Defined

5.9.2 Objective Functions

In our probing experiments, we investigated six objective functions: mutual information, pattern intensity, entropy of the difference image, correlation coefficent, correlation ratio CR(I_{CT-DRR}|MR), and correlation ratio CR(I_{MR}|CT-DRR). The last, CR(I_{MR}|CT-DRR), performed very badly in all tests, and has therefore been omitted from the graphs which follow.

5.9.3 1D Probing at Full-Resolution

The following graphs are characteristic of objective function evaluations over misregistrations in a variety of directions individually, with other directions held at their "ideal" values. Note that the peaks in these functions *do not* represent a maximum of the function (global or local), as the derivative in the directions held constant is not zero. The DRR resolution for these probings was set to 1 mm^2.

- Figure 5-6. Translational misregistrations over the range [-10cm,10cm] with step size 2mm.
- Figure 5-7. Detail of function peaks for z-axis misalignments.
- Figure 5-8. Fine probing of y-axes misalignments highlighting small fluctuations in the functions.
- Figure 5-9. Probing over rotations about the x-axis (out-of-plane in both images).
- Figure 5-10. Probing over rotations about the y-axis.

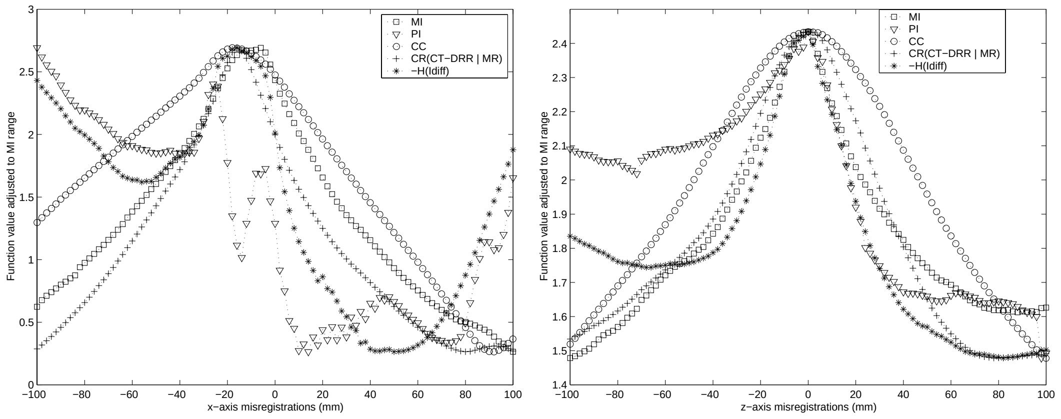

Figure 5-6: Probing a wide range of x and z-axis misalignments

5.9.4 1D Probing at Lower Resolution

The following graphs depicit probings of the objective functions where the model CT-DRR resolution has been set to 2mm^2 and 4mm^2. Contrast these to the graphs over 1mm^2 pixel DRRs. The structure is not considerably changed.

- Figure 5-11. Z-axis misregistrations over the range [-10cm,10cm] with step size 2mm for downsampled models. Compare to Figure 5-6.

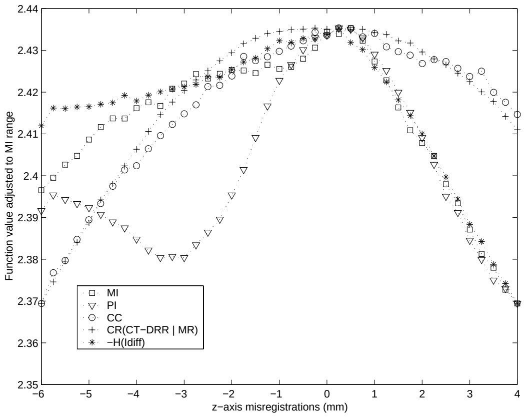

Figure 5-7: Details of peak over z-axis misalignments

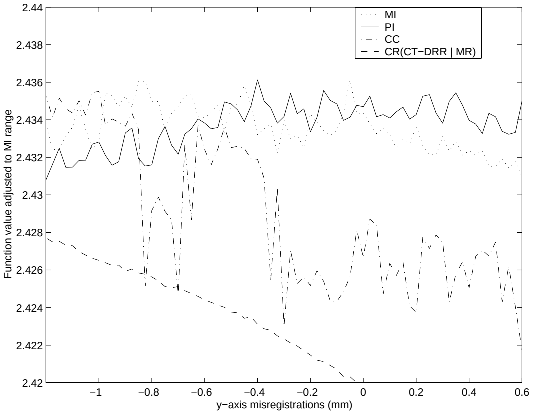

Figure 5-8: Detail of objective function fluctuations

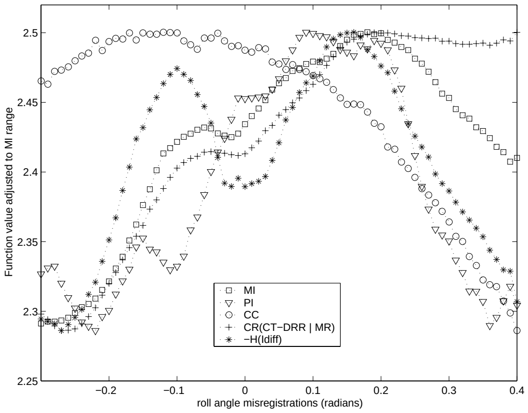

Figure 5-9: Probing over rotations about the x-axis (roll)

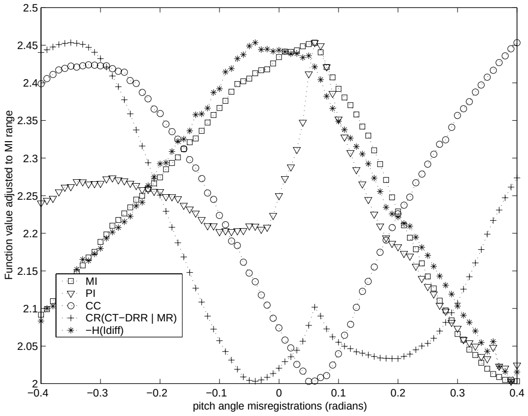

Figure 5-10: Probing over rotations about the y-axis (pitch)

- Figure 5-12. Detail of function peaks for z-axis misalignments and downsampled models. Compare to Figure 5-7.
- Figure 5-13. Probing over rotational misregistrations about the y-axis (pitch). Compare to Figure 5-10.

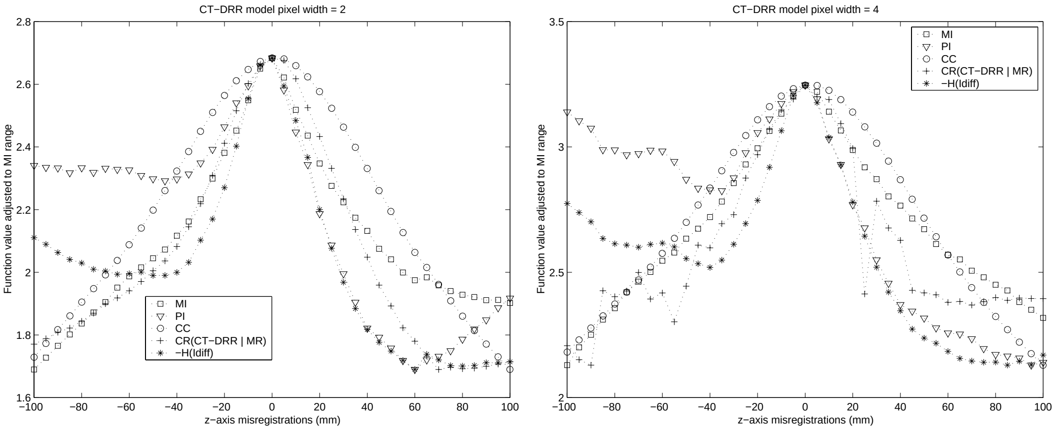

Figure 5-11: Probing a wide range of z-axis misalignments for low resolution CT-DRR models

5.9.5 2D Probing at Full-Resolution

In the course of our experiments, we found that Pattern Intensity performed best in the sense that its peak corresponds to a transformation which is close to that

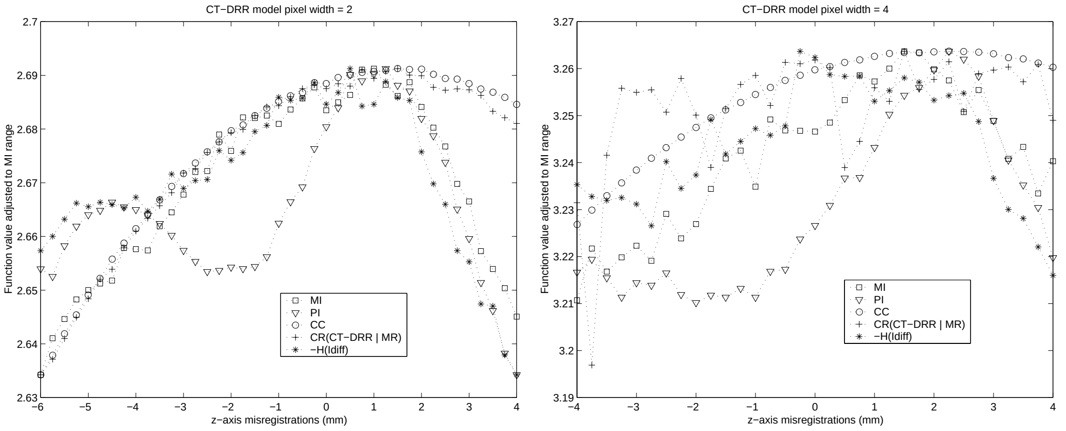

Figure 5-12: Probing a narrow range of z-axis misalignments for low resolution CT-DRR models

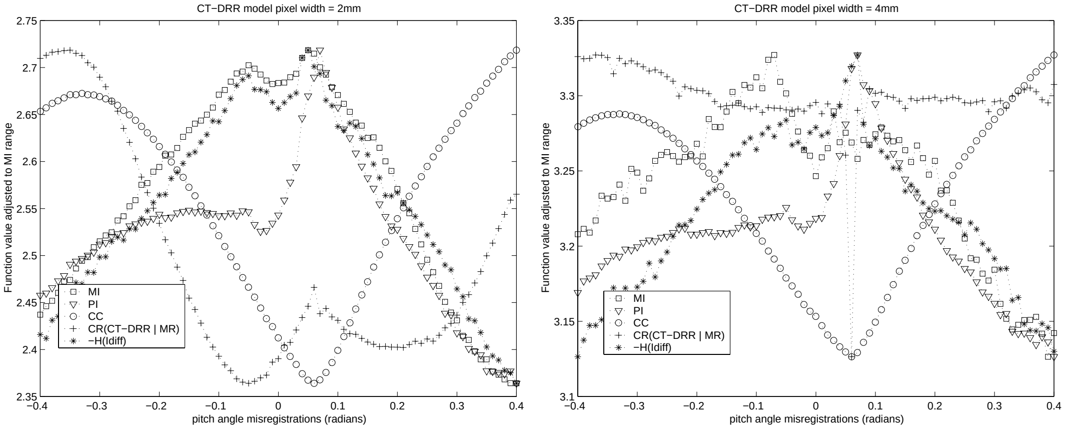

Figure 5-13: Probing over rotational misregistrations about the y-axis (pitch) for low resolution CT-DRR models

which produces the best alignment of the CT and MR volume, and in the sense that it has few local maxima. Therefore, we present some graphs which depicit Pattern Intensity's behavior over two directions of misregistration.

- Figures 5-14. Y and Z-axis misregistrations.
- Figures 5-15. Y-axis and roll angle misregistraions.
- Figures 5-16. pitch and yaw angle misregistraions.

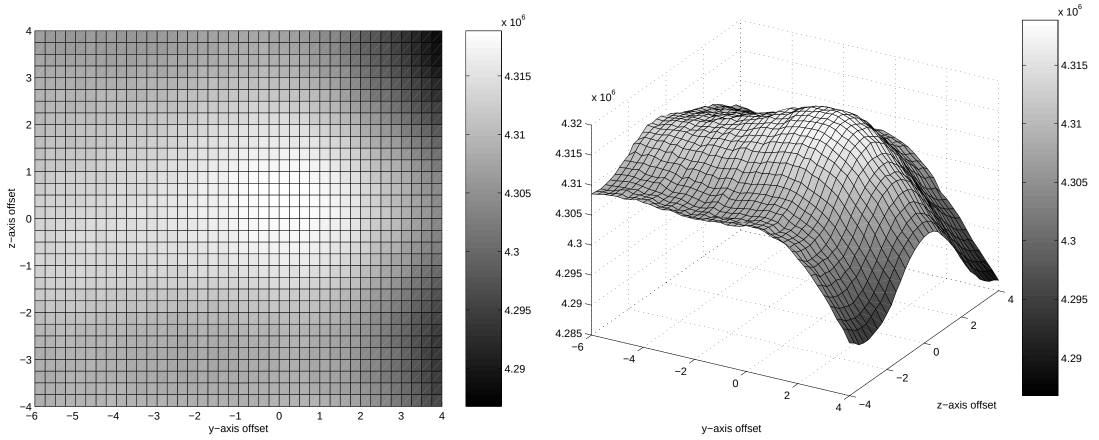

Figure 5-14: Probing of Pattern Intensity in both the y and z-axis directions.

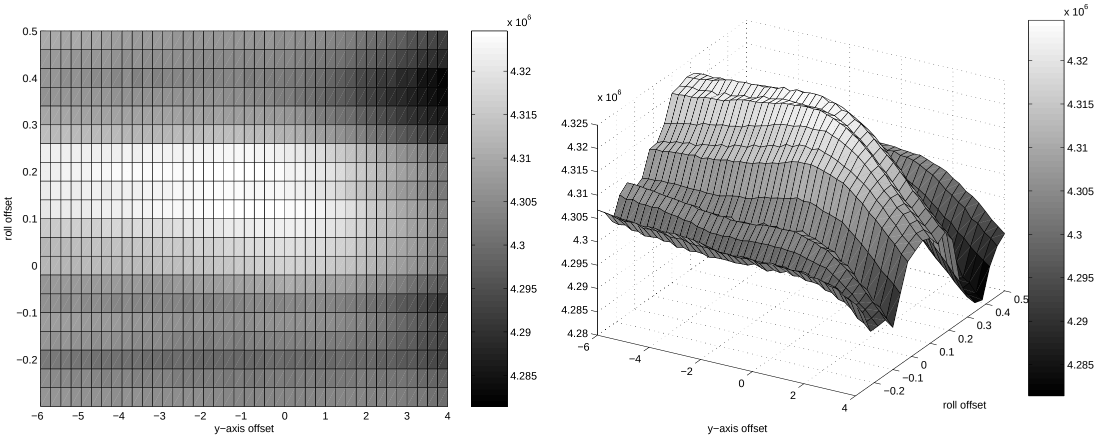

Figure 5-15: Probing of Pattern Intensity in both the y-axis and roll-angle directions.

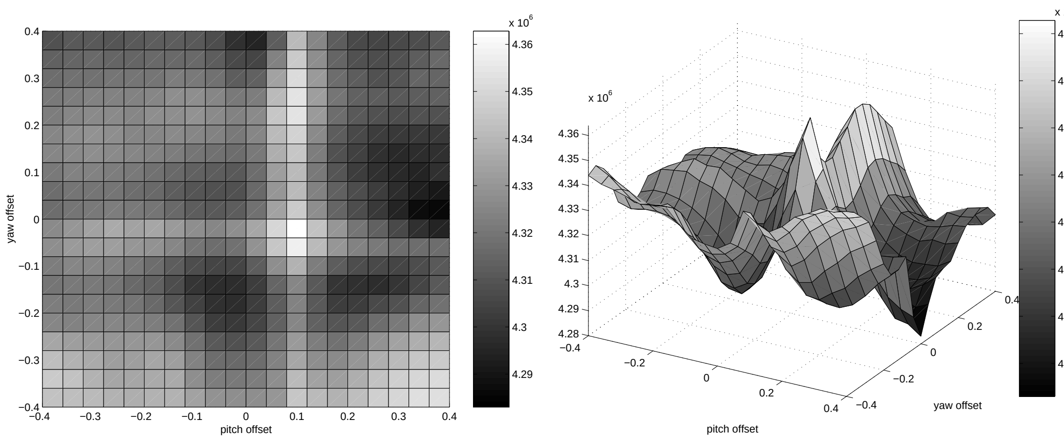

Figure 5-16: Probing of Pattern Intensity over misalignments in volume pitch and yaw.

5.9.6 Discussion

From these explorations, we make the following observations:

- **The objective functions are not smooth.** In Figure 5-8, where probing was performed with a small step size, we can see that there are fine fluctuations in all the objective functions. Furthermore, we see in a number of these functions there are multiple maxima. This is true of all the functions, as can be visualized by higher dimensional probing (3D) which we performed, but which could not be well-presented in this document.

- **Lower resolution impacts behavior minimally**. Comparision of our full-resolution and half-resolution probings suggests that the behavior of these objective functions does not change very much with a limited degree of down-sampling. Although greater degrees will effect function behavior in irregular ways.

- **Capture radius is limited for rotational offsets.** The capture radius in translational directions is apparently very large (Figure 5-6), but this is not the case for rotational offsets. We should expect this, however, given the nature of the data as decribed at the outset of this results section: every rotational direction will be out-of-plane with respect to one of the planes. Therefore, there are few valid rays available to the objective functions in the plane where the out-of-plane rotation is large. Experiments with more complete head images will be required to better characterize objective function behavior in these rotational directions.

- **Pattern Instensity Performs Best.** As was found in CT-fluroscopy registration [11], pattern intensity apparently has the best performance of the objective functions explored, as its peaks are consistently close to the "ideal" values. N-evertheless, any conclusion of this nature must be verified on a larger set of data and under more realisitic interventional conditions (such as, having interventional instruments in fluoroscopic image which are not present in the MR).

Chapter 6

Uphill-Simplex Registration Algorithm

6.1 Overview

In order to use an objective function to register an MR volume to a set of biplanar fluroscopic images, we need a way to automatically optimize the function's value over the domain of rigid transformations relating the datasets' respective metric coordinate systems. Because most of the objective functions we have explored exhibit some degree of non-convexity and are poorly behaved for large misregistrations, a successful optimization method for this 3D-2D multimodal image registration problem will likely have to be robust to these function qualities; that is, one which is able to overcome local maxima and subsets of the transformation range in which the DRR and model radiograph do not overlap (where objective function values are correspondingly constant). One such algorithm which can be robust to these non-idealities is the uphill-simplex method [15]. In this chapter, we will discuss how this optimization scheme can be adapted to this problem, as a way of suggesting its utlimate application.

6.2 Uphill-Simplex Function Optimizaiton

The uphill-simplex algorithm optimizes a function over an n-dimensional domain by maintaining $n+1$ points (referred to as a *simplex*) and adjusting the positions of these points by *reflections* and *contractions*, based on their relative function values. In its main mode of operation, the least-valued point of the simplex is *reflected* across its opposing face (an n-plane determined by the other n points), essentially stepping that worst point in the estimated direction of the function ascent. When no progress can be made by this process, the simplex is *contracted* toward its greatest-valued point. By these contractions, if the simplex reaches the base of a peak in the function, it effectively adjusts its step size to match the peak's sharpness. These reflections and contractions are made by linear extrapolation and interpolation between simplex points; therefore, the n-space over which the simplex method maximizes a function should be Euclidian [5], so that linear combinations of points has the proper effect of bringing points closer together, or extrapolating a point the proper distance.[1]

The simplex method's distributed exploration of a function's domain is what helps it avoid getting stuck at local maxima. Because a number of points are always maintained in different regions of the function's domain, the simplex maintains information over an area of the domain, allowing its focus to shift from a region that initially seems promising, to a region which later proves better. Clearly, the span of the simplex (distance between its component points) mediates how *local* or *global* its information about the function's domain is. If the simplex points are very close together, the simplex maintains only local information, and will adjust its configuration in step sizes of that order. If the simplex points are more spaced, although the simplex doesn't necessarily maintain information about the entire area it subsumes (only about neighborhoods of $n+1$ points in it), it will be able to explore that area by making adjustments matched to the area's scale. This characteristic is very suggestive of the method by which we should initialize the points of the simplex at the outset of an

[1] Becuase rotations are non-Euclidian, we will find that this is an important consideration in choosing a parameterization for the rotational component of the transformation over which our simplex-based registration algorithm optimizes an objective function.

optimization: we should pick our simplex points to span an area in the n-dimensional domain over which we expect to find the global maximum. Of course, the simplex algorithm is not guaranteed to find the global maximum in the area defined at the outset. Indeed, it is possible for a simplex to contract around a non-global maximum, and thus reduce its span so much that it would no longer be able to detect or step to the global maximum. Restarting the simplex algorithm can be effective in such cases, for the precisely the same reason that we reasoned would help the simplex method to overcome local maxima in the first place. The simplex may be restarted so that one of its points rests of the apparent maximum, with the others distributed around that point so that they span an area in which we might find true global optimum. We don't lose very much in the way of time if the previous iteration had found the actual global maximum, because in that case, the simplex is likely to converge back to its best point relatively quickly [15].

6.3 Transformation Parameterization

To use the Uphill-Simplex algorithm in our registration scheme, we must chose a parameterization for the six-dimensional, rigid-transformation space over which the simplex will optimize an objective function. We have two restrictions under which we make this choice. First, we should choose a parameterization which allows our function optimization scheme to explore efficiently the range of misregistrations we expect to see in this problem. Second, as we alluded in the previous section, the domain ought to be parameterized so that it is Euclidean and, thus, consistent with the linear interpolations and extrapolations on which the simplex method relies for proper operation.[2]

[2] Keep in mind, throughout the discussion of this section, that the parameterization we are describing does not necessarily have anything to do with how the transformation is actually applied between the radiograph and volume coordinate frames. It has only to do with the parameters that the simplex optimization scheme works with. A conversion from these parameters to the computational object by which transformation they specify is actually applied, may very well be required.

6.3.1 A Well-formed Transformation Space

We'll explore the first restriction first. As we discussed earlier in this chapter, automatic optimization schemes operate best when a function's variation is slow and free of sharp peaks. An especially pathological function geometry is the long, narrow peak. This geometry is difficult for optimization algorithms to navigate in general [15], and is probably disruptive to the uphill-simplex algorithm because such a peak varies on two very different scales: in one direction its variation is slow, and in the other, fast. Intuitively, we can see that a poor choice of coordinate system might accentuate or create such a geometry, for instance by stretching the scale in one direction, and compressing it in the other, so that a peak becomes very anisotropic. In the case of the parameterization of the transformation between the frames of radiograph and volume, there are certainly formulations that create this problematic geometry, as we can see by the following example:

Let us pick the following formulation of the transformation from the metric coordinates of radiograph to that of the volume, $T_{\text{rigid}}(\mathbf{x}) = R(\mathbf{x} + \mathbf{t})$, where R is a rotation and $\mathbf{t}$ is a translation.[3] Also, say that $\mathbf{t}_a$, the translational component of the ideal transformation, had magnitude $|\mathbf{t}_a|$ large compared to the translational scale τ over which the optimization algorithm could be expected to operate.[4] Now consider what would happen in a typical situation where the volume is centered on its proper position in the radiograph's coordinate frame, but it's orientation is skewed by angle θ. In this situation, $\mathbf{t}$ has ideal magnitude $|\mathbf{t}_a|$, but its direction and the rotation R are non-ideal. Bringing the rotational component to its optimal value, will require a change in the translational component $|\mathbf{t}_a - \mathbf{t}|$ on the order of the arc length $|\mathbf{t}|\theta >> \tau$. For values of $|\theta|$ other than those very small, this will be very large compared to the scale τ on which we expect the optimization algorithm to operate. Furthermore, small variations in the rotation R will lead to drastic shifts in the volume's position with respect to the plane, as well as the value of the objective function (especially if the

[3] The translation is applied first, and the rotation follows.

[4] We might encounter this situation if $\mathbf{t}$ moved the volume from the center of the projective plane to its position above. Then, $|\mathbf{t}_a|$ would be roughly half the focal length of the projection $f \simeq 1.5m$, dwarfing the scale τ of even a large misregistration, less than ten centimeters.

variation in rotation shifts the volume's projection outside the radiograph's bounds).

This parameterization is clearly unwieldy and accordingly unsuccessful in practice; however, its failures are suggestive of a formulation for the transformation which affords an appropriate degree of "independence" between its translational and rotational components. We parameterize the transformation T from the metric frame of the volume to that of the radiograph, with a translational component $\mathbf{t}$ and a rotational component R, where $\mathbf{t}$ determines the position of the volume's center (i.e. center of mass, the central voxel, or whatever makes sense) in the radiograph's metric coordinate frame, and R determines the orientation of the volume about that center. We can formalize this as follows, where $\mathbf{x}$ is given in the coordinate frame with origin at the volume's center:

$$T(\mathbf{x}) = R(\mathbf{x}) + \mathbf{t} \tag{6.1}$$

This formulation seems very natural as it is defined relative to the volume in a way that reflects what we would do if we were able to manually register the volume by physically positioning it in space: we would be able change its rotational orientation with respect to the plane without affecting its position, and vice versa. For this same reason, it is also more successful in the context of our registration algorithm, as it allows our optimization algorithm to efficiently explore the full range of rotational misalignments.

6.3.2 A Euclidean Rotation Space

Now that we have appropriately formulated the transformation between the volume and radiograph metric frames, we must consider the second restriction that we mentioned at the outset of this chapter, and parameterize the transformation such that it defines a Euclidean domain on which the objective function can be maximized. For the translational part, this is easy, as we can simply pick a Cartesian coordinate system, which has this property automatically. The rotational parameterization does not follow so easily, as rotations in three dimensions (which we're considering here) are non-Euclidean so that many parameterizations have singularities and geometries under which linear interpolation and extrapolation do not behave regularly. One standard formulation for rotations which suffers from this problem [5] is that of Euler Angles, by which rotations are characterized by an angle of rotation about each of the coordinate system's basis vectors. On the other hand, another standard parameterization for rotations, the *Quaternion*, has been used in differential control and computer graphics systems because it has the property that interpolations behave regularly over a series of rotational orientations. The *Reparameterized Quaternion* (RQ) parameterization, discussed by Grassia [5] for these applications, is also appropriate in the medical image registration setting, as we will see.

Quaternions are 4d quantities that characterize rotations as an angle of rotation θ about a unit axis $\hat{\mathbf{n}}$ in a three dimensional coordinate system. A quaternion can be formalized as

$$q = [s, \mathbf{v}] \tag{6.2}$$

where $s = \cos(\frac{\theta}{2})$ is a scalar and $\mathbf{v} = \sin(\frac{\theta}{2})\hat{\mathbf{n}}$.[7] Quaternions have some special properites that make them a generally useful representation of rotations. Rotations $r = p \circ q$ may be composed by quaternion multiplication, defined as follows.

$$r = pq = [s_p s_q - \mathbf{v}_p \cdot \mathbf{v}_q,\ s_p \mathbf{v}_q + s_q \mathbf{v}_p + \mathbf{v}_q \times \mathbf{v}_p] \tag{6.3}$$

where $p = [s_p, \mathbf{v}_p]$ and $q = [s_q, \mathbf{v}_q]$ are quaternions. Furthermore, a vector $\mathbf{x}$ may be rotated by quaternion r by forming the quaternion $q_{\mathbf{x}} = [0, \mathbf{x}]$ and applying the following multiplication.

$$r q_x r^* \tag{6.4}$$

where r^* is the conjugate of quaternion r, defined as $r^* = [s_r, -\mathbf{v}_r]$.

The fact that the vector component $\mathbf{v}$ of the quaternion is defined in a 3d Euclidean space, suggests that we might use this quantity alone for our purposes as a parameterization of a rotation in a 3d space. In such a space, rotations with the same angle θ are found on the same origin-centered sphere with radius $\sin(\frac{\theta}{2})$. The space is bounded by the unit sphere ($\max |sin(\cdot)| = 1$), and is periodic in the sense that if we travel along a given axis of rotation $\hat{\mathbf{n}}_0$ in the direction of increasing θ, when we reach the unit sphere and virtually travel beyond it to $\theta = 2\pi + \epsilon$, we will loop to the opposite side of the sphere to position $\mathbf{v} = -\sin(\frac{2\pi-\epsilon}{2})\hat{\mathbf{n}}_0$. For the purposes of a simplex-based optimization scheme, especially for volume-radiograph registration, this formulation presents a number of problems when the optimization is focused near the unit sphere. First, near the unit sphere, angles of rotation are not well differentiated due to the compression of vector magnitudes in that region ($\frac{\partial}{\partial\theta}\sin(\frac{\theta}{2})$ small for $|\theta|$ near 2π). Second, the periodicity would cause improper behavior for certain simplex reflections: were a simplex point to be reflected (virtually) beyond the unit sphere, the resulting quaternion would be ill-defined, having magnitude greater than unity. Simply reflecting such a point to its proper position at the other side of the sphere would drastically change the area of the domain subsumed by the simplex in a way not intended by the reflection operation. This behavior is not desirable in general and can result in pathological behavior if the objective function is not well-defined over the entire rotational domain.

We can salvage this approach, however, by using the RQ parameterization for our rotations. This simply involves changing the scaling of the quaternion vector, so that magnitude in 3d vector space corresponds to angle of rotation θ directly, rather than to $\sin(\frac{\theta}{2})$. Therefore, the RQ vector for a rotation $\mathbf{v}_r$ is defined with respect to its quaternion vector-component $\mathbf{v}$ by the relation

$$\mathbf{v}_r = \frac{\theta}{\sin(\frac{\theta}{2})}\mathbf{v} = \theta\hat{\mathbf{v}} \tag{6.5}$$

The rotation's vector space is now very similar to what it was before, where rotations of the same magnitude reside on the same origin-centered sphere, but the space is no longer bounded by the unit sphere and the angular scale is uniform along radii of these spheres. Furthermore, by using RQ, we have a parameterization for the transformation in which both its translational and rotational components reside in a Euclidean space and are thereby consistent with the linear interpolations and extrapolation of simplex points by which our objective function optimization scheme operates.

6.4 Multi-resolution Schedule

With the parameterization we've discussed, we can now use the uphill-simplex algorithm to find the maximum value of an objective function over the space of possible transformations between radiograph and volume. As we discussed in describing the simplex optimization algorithm in general, such a registration would be started by initializing a simplex to subsume the range in which we expect to find the proper alignment, and perhaps restarted a small number of times once the algorithm converged, to reduce the chances of getting stuck in a local maximum. We also have the option of using a multi-resolution scheme for the purpose of reducing the overall running time.

Multi-resolution schemes have been used in a number of medical image registration algorithms for the purpose of increasing the *capture radius*[5] of the scheme and decreasing the running time. In these schemes, the datasets are smoothed and decimated in a schedule of decreasing degree of downsampling. At each level of the schedule, the images are registered and the resulting pose is used as a starting point for the next level. To increase the capture radius, these schemes rely on the fact that smoothing and decimating the datasets may lead to objective function behavior which is smooth over a broader region surrounding its maximum. A multi-resolution scheme may speed registration because part of the schedule involves processing of smaller datasets. However, Pluim et al. found that this was not the case for their approach to the registration of multimodal volumetric data [13]. Indeed, in our preliminary investigations, the four-level schedules corresponding to downsampling factors of (4,

[5] *Capture radius* may be defined as the approximate maximum degree of initial misalignment under which a registration scheme can operate successfully.

3, 2, 1) proved to slow down volume-radiograph registration by the approach we're employing here. This seemed to be due to the fact that registration at the level of lowest resolution (highest downsampling factor) came close enough to the full-resolution registration, for a full-resolution to work sucessfully. In any case, we also found that large downsampling factors introduced new extrema into the objective function's range. Therefore, by these prelimiary investigations, a multi-resolution schedule which begins with one or two cycles at the lowest resolution, followed by one or two cycles at the highest resolution seems to make sense for simplex-based, volume-radiograph registrations. Note that at each of these levels, the span of the simplex should reflect the magnitude of the misregistration we expect at that level. Because with each level of registration, we expect that the algorithm is getting closer to the ideal transformation, the area subsumed by the simplex should be accordingly reduced. This schedule generally improved the registration running time from that of a purely full-resolution schedules.

6.5 Summary

In this chapter we have outlined a manner by which we can apply the uphill-simplex function optimization scheme to the intensity-based registration of volumes and radiographs, assuming that we have an appropriate objective function. We suggested a parameterization for the transformation from the coordinate frame of the radiograph to that of the volume, which is consistent with the mechanisms of the uphill-simplex optimization algorithm and which should allow this algorithm to explore the transformation space efficiently. Finally, we suggested a two-level multi-resolution scheme which has proved to improve the running time of the registration process in our prelimiary experiments.

Chapter 7

Conclusion

The *voxel-projection* algorithm for the production of DRRs has been developed, which provides drastic speed improvements over *ray-casting* methods serving the same function. This method has proved to be a useful tool for the investigation of objective functions used in the registration of volumetric MR images to radiographs. Voxel-projection may also prove to be a useful method for making 3D-2D intensity-based registration practically fast.

A variety of objective functions were studied to determine their utility in the context of an MR-fluoroscopy, intensity-based, rigid registration scheme. Applications were developed to evaluate these functions over a range of misregistrations between an MR volume and biplane DRRs formed from a corresponding CT volume. The CT and MR were first registered by proven 3D-3D methods, so that the ideal alignment of MR and radiographs was known and could be used to evaluate the performance of the objective functions in identifying that alignment. These functions were also evaluated over different levels of model smoothing and downsampling, in anticipation of their potential application in a multi-resolution scheme. It was found that their behavior did not change drastically when the model was downsampled by a factor of 2 or 4.

Based on the non-convexity of the objective functions evaluated, the uphill-simplex method was adapted as a multi-resolution optimization scheme for MR-fluoroscopy, intensity-based registration.

7.1 Related and Future Work

7.1.1 Voxel-projection

The voxel-projection algorithm has the nice property that it is very easily parallelizable: multiple DRRs may be produced from volume data with a single pass over the volumes voxels in data order. In fact, we implemented such scheme for the production of biplane DRR (or any number of DRRs from different angles) simultaneously on a single-processor system. While performance did not improve using only a single processor, were multiple processors employed to handle each projection, an arbitary number of projections could be produced from a single volume in memory in the same time it would take to make one individually. A three-step search (Section 3.4.2, [18]) modification like that used in [11] would be drastically speeded by such parallelization, as each iteration of that registration algorithm requires objective function evaluation on twelve newly-produced DRRs per plane (i.e. a biplane setup would require twenty-four new DRRs on every iteration). Furthermore, this parallelization might be very useful in registering volumetric data to fluoroscopic "C-arm" radiographs taken from many viewpoints.

7.1.2 Objective Function Evaluation

Effective evaluation of objective functions for MR-fluoroscopy registration will require a broader library of test data than was employed for this thesis. An essential component of such a library is a set of MR data which has been aquired over a broad enough region to allow for a sufficient degree of out-of-plane rotation. One objective function (see Chapter 3) which measures the model-based likelihood [9] of an alignment could be explored if a sufficently broad set of test data were available to construct a model for the joint distribution of aligned MR-DRRs with real radiographs.

7.1.3 MR Intensity Modification

The modification of MR intensities to better reflect the x-ray attenuation coefficients of the tissues from which they were produced, may have a very large impact on the success of an intensity-based MR-fluoroscopy registration scheme. One scheme which we have begun to investigate involves segmenting the skull from MR head images and then adjusting its intensity to resemble that of bone in CT. Hard bone has the same low intensity of air in MR (see Chapter 2) and is thus easily distinguishable from surrounding tissue by thresholding. Bone and air were distiguished in head images by using connected components analysis to pick out the air which surrounds the head and which is in the sinuses (by their size), and morphological tools (structuring elements) [10] to break any erroneous connections from the skull to the sinuses.

DRRs resulting from this operation did have more pronouced bony structure than those from MR which has not been modified, but so far, this has had little apparent impact on the behavior of investigated objective functions. However, work by Petra van den Elsen in CT-MR registration might be applied here, in which she modified CT intensties so that bone was given low-intensity [20], or in which she used morphological tools to pick out head skin edges [10].

Appendix A

Memory Hierarchy and Data Caching

Any discussion of algorithmic performance is well-served by considering the effects of computer memory architectures, especially now when computation speeds so greatly outstrip those for memory access. In this appendix we'll give an overview of the cache effects that bear on our discussion from Chapter 4 comparing the ray-casting and voxel-projection methods of DRR production.

Modern computer architectures typically have five levels of memory which vary in size and access time. In order from slowest to fastest, and largest to smallest, the hierarchy is shown in Table A.1. The access latencies for each level (by current standards) are also given there. Caching systems (L1 and L2 in Table A.1) are designed to provide performance improvement to programs whose memory access

Table A.1: Memory Latencies

Element	Latency
Disk	10ms
Main Memory	50ns
Level 2 (L2) off-chip cache	4-8ns
Level 1 (L1) on-chip cache	2ns
Registers	2ns

pattern exhibit qualities of locality in time and space. In other words, caching system speed programs which (repeatedly) access data that is proximate in memory, without many intervening references to data stored in remote memory locations.

They do this in two ways. First, they store recently-accessed data in a location from which it may be accessed again with little latency, so that programs which repeatedly access the same locations will suffer less from data access overhead. Second, when an element of memory is referenced and pulled into the cache, along with it is pulled the block of memory, called a *cache line*, in which that element is contained. Cache lines are typically 32 bytes in size, aligned along 32-byte boundaries in memory[1]. This scheme anticipates that when a program references one memory location, it may soon access some nearby one. Therefore, even routines that do not repeatedly reference the same data, will benefit from data caching. Furthermore, routines take maximal advantage of this caching scheme when they access every element of a cache line once it has been pulled, for each of these accesses will be subject only to the latency of the cache, not that of some slower storage device, higher up in the memory hierarchy. An access pattern that guarantees this behavior is that of referencing data in the order of their memory locations. This type of pattern also has the advantage of requiring a minimal amount of computational architecture to support it (requiring only an incrementing of the memory location).

[1] The L1 cache of the UltraSPARC architecture used in this thesis has 16-byte cache lines.

Bibliography

[1] Ravi Bansal, Lawrence Staib, Zhe Chen, Anand Rangarajan, Johnathan Knisely, Ravinder Nath, and James Duncan. A novel approach for the registration of 2d portal and 3d ct images for treatment setup verification in radiotherapy. Medical Image Computing and Computer-Assisted Intervention (MICCAI), page 1075, October 1998.

[2] Dov Bulka and David Mayhew. *Efficient C++: performance programming techniques.* Addison-Wesley, Massachusetts, first edition, 2000.

[3] John Michael Fitzpatrick. The retrospective registration evaluation project. http://cswww.vuse.vanderbilt.edu/ image/registration/.

[4] D. Gering, A. Nabavi, W.E.L. Grimson, N. Hata, P. Everett, F. Jolesz, , and W. Wells. An integrated visualization system for surgical planning and guidance using image fusion and interventional imaging. Proceedings of the Second International Conference on Meical Image Computing and Computer Assisted Intervention., 1999. Submitted.

[5] F. Sebastian Grassia. A practical prarmeterization of 2 and 3 degree of freedom rotation. Technical Report CMU-CS-97-143, Carnegie Mellon University, School of Computer Science, Pittsburgh, PA, May 1997.

[6] DLG Hill, DJ Hawkes, C Studholme, PE Summers, and MG Taylor. Accurate registration and transformation of temporal image sequences. In *Proc. 2nd Annual Meeting Society of Magnetic Resonance*, page 820, 1994.

[7] Berthold Klaus Paul Horn. *Robot Vision.* The MIT Press, third edition, 1986.

[8] CR Maurer Jr., GB Aboutanos, BM Dawant, S Gadamsetty, RA Margolin, R-J Maciunas, and JM Fitzpatrick. Effect of geometrical distortion correction in mr on image registration accuracy. In *Medical Imaging VIII: Image Processing*, number 2167, pages 200–213, 1994. Proc. SPIE.

[9] ME Leventon and WEL Grimson. Multi-model volume registration using joint intensity distributions. In *Proceedings of the Second International Conference on Medical Image Computing and Computer Assisted Intervention.* http://www.ai.mit.edu/projects/vision-surgery.

[10] J.B. Antoine Maintz, P.A. van den Elsen, and M.A. Viergever. Registration of 3d medical images using simple morphological tools. Technical report, Image Sciences Institute, PO Box 8550, AZU-E.01.334, NL-3508 GA Utrecht, the Netherlands.

[11] Graeme P Penney, Jurgen Weese, John A Little, Paul Desmedt, Derek LG Hill, and David J Hakes. A comparison of similarity measures for use in 2d-3d medical image registration. Technical report, Image Sciences Institute, PO Box 8550, AZU-E.01.334, NL-3508 GA Utrecht, the Netherlands.

[12] Joseph P. Hornak Ph.D. The basics of mri. http://www.cis.rit.edu/htbooks/mri/.

[13] Josien PW Pluim, JB Antoine Maintz, , and Max Viergever. Mutual information matching in multiresolution contexts. Technical report, Image Sciences Institute, Utrecht, the Netherlands.

[14] B. A. Porter, W. Hastrup, , and M. L. Richardson. Classification and investigation of artifacts in magnetic resonance imaging. *RadioGraphics*, 7(2):271–287, mar 1987.

[15] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C.* Cambridge University Press, second edition, 1992.

[16] Alexis Roche, Gregorie Malandain, Xavier Pnnec, and Nicholas Ayache. Multimodal image registration by maximization of the correlation ratio. Technical Report 3378, INRIA, 2004, route des Lucioles, B.P. 93, 06902 Sophia Antiplis Cedex, France., 1998.

[17] C Studholme, Derek LG Hill, and David J Hakes. Using voxel similarity as a measure of medical image registration. Technical report, UMDS, London, England.

[18] A. Murat Tekalp. *Digital Video Processing*. Prentice Hall, 1995.

[19] The ultrasparc design philosophy. http://www.sun.com/microelectronics/whitepapers/UltraSPAR

[20] P.A. van den Elsen, E-J. D. Pol, T.S. Susmanaweera, P.F. Hemler, S. Napel, , and J.R. Adler. Gray value correlation technique use for automatic matching of ct and mr brain and spine images. In *Proc SPIE*, number vol.2359, pages 227–237, 1994.

[21] Paul A. Viola. Alignment by maximization of mutual information. Technical Report 1548, MIT Aritifical Intelligence Lab, Cambridge, MA, June 1995.

[22] Paul A. Viola and William M. Wells III. Alignment by maximization of mutual information. Fifth International Conference on Computer Vision, pages 16–23, Cmabridge, MA, 1995. IEEE.

[23] Steve Web, editor. *Medical Science Series*. Medical Science Series. Institute of Physics Publing Ltd, Bristol and Philadelphia, 1998.

[24] WM Wells, R Kikinis, WEL Grimson, and F Jolesz. Adaptive segmentation of mri data. *IEEE Transactions on Medical Imaging.*, 15:429–442, 1996.

[25] Z. Yaniv, L. Joskowicz, A. Simkin, M. Garza-Jinich, and C. Milgrom MD. Fluoroscopic image processing for computer-aided surgery. Medical Image Computing and Computer-Assisted Intervention (MICCAI), pages 328–329, 1998.

[26] Jianhua Yao, Russel H. Taylor, Randal P. Goldberg, Rajesh Kumar, Andrew Bzostek, Robert Van Vorhis, Peter Kazanzides, Andre Gueziec, and J. Funda. A progressive cut refinement scheme for revison total hip replacement surgery using c-arm fluoroscopy. Medical Image Computing and Computer-Assisted Intervention (MICCAI), pages 1012–1015, 1999.