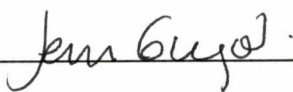


To the Graduate Council:

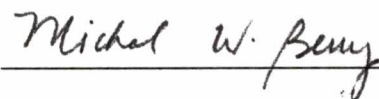
I am submitting herewith a thesis written by Gregory Joel Wrobel entitled "Interpolation and Registration of Medical Image Volumes from Multiple Modalities." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Computer Science.



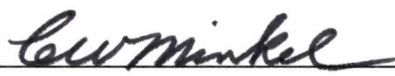
Jens Gregor, Major Professor

We have read this thesis
and recommend its acceptance:





Accepted for the Council:



Associate Vice Chancellor
and Dean of the Graduate School

**INTERPOLATION AND
REGISTRATION OF MEDICAL
IMAGE VOLUMES FROM
MULTIPLE MODALITIES**

A Thesis

Presented for the

Master of Science Degree

The University of Tennessee, Knoxville

Gregory Joel Wrobel

August 1996

Acknowledgments

I have had a unique and rewarding experience at the University of Tennessee. The faculty and graduate students in the Department of Computer Science are of high calibre and have challenged me to excel. I am very thankful for the support of my Thesis Committee, Michael Thomason, Michael Berry, and particularly Jens Gregor for his help and guidance. They have all reviewed this thesis and their suggestions and time have been invaluable to its completion.

I would like to thank Dr. Kent Hutson who proposed this project and consistently offered excellent advice with technical aspects. I owe much to John Martin, Dian Witter, and Ken Schmidt for their friendship and help both with my work and with the alleviation of it. Also, I have received generous financial support from the Department of Computer Science and the University of Tennessee Medical Center.

My thanks and love go to my wife, Vicki, who has given unflagging support and encouragement without which I would have long ago given up this pursuit.

Lastly, I owe my greatest thanks to God for the opportunities presented to me.

Abstract

In this thesis several methods of interpolating and registering anatomical MRI, CT, and PET image scans are reviewed. A modified linear interpolation procedure is developed to create three dimensional volumes from two dimensional image scans formatted according to the DICOM standard. This technique determines a weighted mean intensity for each unknown voxel using weights based on distances from known values. The volumes are then processed using a principal axes registration method which translates the centroids of two volumes to the same position and rotates the volumes so that their principal axes are aligned. The principal axes needed to accomplish this task are determined by computing the eigenvectors of the volume's covariance matrix. Merging the two procedures provides an efficient method of determining a global rigid transform to register two sets of image scans. Both algorithms have linear computational complexity with respect to the number of voxels in a volume, but the speed of the interpolation method is highly dependent on the separation between image scans. Interpolation speed increases as the distance between scans decreases. This technique provides an excellent coarse registration of images which can be easily followed with more intensive, local methods.

Contents

1	Introduction	1
1.1	Problem Statement	1
1.2	Volume Interpolation Overview	3
1.2.1	Difficulties in Interpolation	3
1.2.2	Interpolation Methods	6
1.3	Volume Registration Overview	9
1.3.1	Registration Classification	9
1.3.2	Registration Algorithms	14
1.4	The DICOM 3.0 Standard	21
1.5	Overview	25
2	Volume Interpolation	27
2.1	Interpolation Methods	27
2.1.1	Linear and Tri-Linear Interpolation	27
2.1.2	Cubic Interpolation	30

2.1.3	Shape-Based Interpolation	33
2.1.4	Kriging	35
2.2	Modified Linear Interpolation	37
2.2.1	Voxel Intensity Computation	38
2.2.2	Computing Voxel Positions	40
2.2.3	Implementation Strategy	42
3	Volume Registration	45
3.1	Registration Methods	46
3.1.1	Registration by Polynomial Approximation	46
3.1.2	Registration by Feature-Extraction	49
3.1.3	Elastic Matching	52
3.2	Principal Axes Method	55
3.2.1	Eigenvector Computation	55
3.2.2	Computing the Covariance Matrix	59
3.2.3	Derivation of the Hotelling Transform	60
3.2.4	Principal Axes Registration	62
4	Results	65
4.1	Accuracy Verification Methods	66
4.2	Error Analysis	69
4.3	Registration of Anatomical Data	75

5 Conclusion	79
Bibliography	82
Vita	86

List of Figures

1.1	Effect of transformations on image.	14
1.2	Steps in principal axes registration.	20
1.3	Network model of DICOM.	23
1.4	Position and orientation of image with respect to the patient. . . .	25
2.1	Tri-linear interpolation.	29
2.2	Cubic interpolation.	32
2.3	Shape-based interpolation.	34
2.4	Image position vectors.	41
2.5	Image orientation angles.	42
3.1	Orthogonal vectors, w and v , in computation of the L_{vv} operator.	51
3.2	Registration using centroids and principal axes.	64

4.1	Comparison of rotation versus voxel size for a phantom image with an initial rotation of 10 degrees about the y and z axes and an initial translation of 10 mm along the x, y, and z axes.	71
4.2	Comparison of translation versus voxel size for a phantom image with an initial rotation of 10 degrees about the y and z axes and an initial translation of 10 mm along the x, y, and z axes.	72
4.3	Comparison of rotation versus voxel size for a phantom image with an initial rotation of 0 and 10 degrees about the y and z axes respectively, and an initial translation of 10 mm along the x, y, and z axes.	73
4.4	Comparison of average rotation versus voxel size given initial rotations about the y and z axes of both 0 and 10 degrees.	74
4.5	Slices obtained from source, target, and registered source phantom volumes.	76
4.6	MRI slices of cranium and shoulder obtained from source, target, and registered source volumes.	77
4.7	MRI Cranial slices obtained from source, target, and registered source volumes.	77
4.8	MRI lices displaying misregistration due to improper segmentation.	78

Nomenclature

CT computer tomography.

MRI magnetic resonance imaging.

PET positron emission tomography.

DICOM Digital Imaging and Communications in Medicine.

ACR American College of Radiology.

NEMA National Electrical Manufacturers Association.

OSI Open System Interconnection.

ISO International Standards Organization.

RSNA Radiological Society of North America.

Chapter 1

Introduction

1.1 Problem Statement

Medical images have become an integral part in the process of diagnosing and treating cancer and other disorders. These images are primarily obtained through computer tomography (CT), magnetic resonance imaging (MRI) and positron emission tomography (PET) methods. Many times different sets of these images are obtained from the same part of the subject's anatomy over time or multiple sets are taken with differing slice angles or different imaging modalities are used in each set. It is often desirable to compare images from different sets that correspond to identical slice of the anatomy. This allows comparison of functional verses highly detailed anatomic images, for example PET images compared with CT images, or tracking of anatomical changes such as the growth of a tumor over a period of

time.

In order to make these comparisons, the sets of images must be aligned, or registered, to each other. In general, the individual images used in registration represent two dimensional slices separated by a displacement of several millimeters. From these slices, a three dimensional volume may be interpolated. The volumes are then registered and scaled as necessary, and matching slices are extracted from them.

In our case, we wish to register medical images that are formatted as objects conforming to the DICOM 3.0 standard which will be described later in this chapter. The images have an x, y resolution of approximately 0.6 mm with close to a 10.0 mm displacement between individual images. There are no constraints on the orientation of the images with respect to the patient or to each other. It is the goal of this thesis to describe a method to efficiently interpolate three dimensional volumes from images similar to those described above using a linear interpolation variant, and then to show how the resulting volumes may be globally registered using the Hotelling, or Principal Axes, transform.

The rest of this chapter describes concepts used in the interpolation and registration of medical images and introduces specific examples of each, noting the methods used in our implementation. A description of the DICOM 3.0 standard is provided with a discussion of the attributes of the standard applicable to our situation. Finally, an outline of the rest of the thesis is given.

1.2 Volume Interpolation Overview

Before one volume can be registered to another, both volumes must be interpolated from the individual images in the scan series. This process of interpolation involves estimating the greylevels of voxels in the volume from the greylevels of pixels in the images. Methods of interpolation range from simple averaging (linear interpolation) to algorithms which integrate structural or textural attributes obtained from nearby pixels in the images. Often the types of images used as well as the qualities they possess present unique problems to the interpolation process. This section describes some of the problems encountered in interpolating image volumes based on medical scans. Following these descriptions is a brief overview of some of the interpolation methods that can be used to address these problems.

1.2.1 Difficulties in Interpolation

Depending on the parameters of the images, several problems may be encountered in implementing the interpolation process. These problems are often interdependent and the solution to one problem may complicate another. Four significant problems are 1) widely varying resolution along different axes in an image; 2) large gaps between consecutive images in a series; 3) the need to retain as much frequency information as is possible; and 4) efficiency.

The first problem is typical of CT, MRI, and PET images. In general, these

images can have a 0.5 mm by 0.5 mm resolution along two axes and yet have up to a 10.0 mm resolution along the third. When a three dimensional volume is created, the differences in resolution distort the result. It is desirable to create a volume that has a uniform resolution throughout so that slices at varying oblique angles may be easily extracted. If, however, the grossest resolution of the images is chosen to be the resolution for the volume (e.g. 10.0mm in the x, y, z axes) too much information would be lost in the interpolation process. If the finest resolution of the images is used, then the interpolated values are highly unrealistic and could have large error.

The second problem is similar in many respects to the first. The images that are obtained can have a separation between them of up to 8.0 mm to 10.0 mm. Because of this the gaps between the images must still be interpolated from pixels that are distant and therefore the interpolated values of the voxels in these gaps is less likely to be accurate. It should be noted that this problem of reduced accuracy between images does not imply that the volume will necessarily have the reduced accuracy along the z axis. This is because the images may have been obtained at any oblique angle. Therefore if the images lie parallel to the y, z plane, the reduced accuracy will be along the x axis.

Essentially, because of the reduction in accuracy along some direction in the volume, banding is observed when the volume is viewed as a whole. The planes in the volume corresponding to an actual scan image are much more accurate

than the planes which represent the gaps. Therefore the planes representing the images are sharper in appearance, and the planes representing the gaps appear to be blurred. Also, if a 2-D slice is extracted from the volume that is not parallel to the original images, it too will exhibit banding.

An immediate solution to the banding problem is to apply a smoothing filter to the regions of the volume that are sharper in appearance. This creates a volume that is uniform in appearance which can be more pleasing to the viewer. However, this effects the seriousness of the third problem.

Many imaging applications use the Fourier transform, therefore frequency information in an image must be maintained. This introduces the third problem of needing to avoid unnecessary blurring of a volume so that frequency information is not lost. Because of this fact, the simple solution above to the banding problem is not acceptable if the volume is to be used for purposes other than viewing. Also, multiple interpolations of the same volume (e.g. to rotate or translate the volume) should be avoided. Solving this problem requires finding an interpolation method which will create a sharp image in all regions of the volume.

The last problem is the need for an efficient interpolation method. In general, methods which are more exact have a greater time complexity. A relatively low resolution volume, about 2.0 mm, can easily require the interpolation of over one million voxels. Higher resolutions increase the space complexity in a cubic fashion. Because of the size of the problem, the solution may require less accurate methods.

As can be seen, the solutions to different interpolation problems may be mutually exclusive or at the least be very difficult to determine simultaneously. Therefore, before an interpolation method is chosen, careful consideration must be given to what requirements are most important to the application. For the purposes of this thesis, it was determined that the most important requirements were to maintain frequency information in the volume and to be efficient. With this in mind, we now turn to a brief introduction of five different interpolation methods: linear, tri-linear, cubic, shape based, and kriging. A more thorough treatment of these methods will be given in Chapter 2.

1.2.2 Interpolation Methods

Several interpolation methods exist and we now summarize five of these described by Parrott et al. [16]. Probably the fastest form of interpolation is simple linear interpolation. It involves taking two points, determining parameters for a line that would fit between them, and estimating the values of other points lying along the line. In imaging, the coordinates of the points are simply the coordinates of two known voxels with function values being the intensity levels of those voxels. Although it is faster than other methods, it does not take into account information from other voxels or from the surrounding structure in the image. This makes linear interpolation less likely to maintain the detail of an image.

Tri-linear interpolation attempts to overcome part of the problem presented

by linear interpolation by considering more points from a volume. Instead of two points, 8 points are used to calculate a unknown voxel's intensity level. Because more information is being used in the calculation, one expects results that are more accurate. However, this form of interpolation still does not include structural information from the image. It is also computationally more intensive than linear interpolation.

Another method which considers only two points in the calculation of the value at an unknown location is cubic interpolation. This form of interpolation can be more accurate than linear interpolation because it utilizes the tangents of the imaging function at the known points. The best-fit cubic equation passing through those points with the correct tangent values is used to estimate the value of an unknown point lying between. Thus, if the image function is decreasing at the known points in the direction of the unknown point, the estimated value at the unknown point will reflect this aspect. Cubic interpolation does not involve more than two known points in the computation process, however it could easily be modified to do so at a greater computation cost.

Shape-based interpolation integrates information about the shape of an image in a region surrounding the unknown voxel in the calculation of its intensity. This shape information is based on the contrast between the boundaries of different materials represented in the image. An image is represented in binary form so that the boundary of a particular material can be easily determined. Voxels which lie

inside the boundary are interpolated from known voxels also within the boundary. This method is also computationally expensive because it requires the following independent steps in the interpolation process: image segmentation, boundary interpolation, and voxel interpolation. However, it does involve structural components of the image as well as several data points per interpolated value.

The final interpolation method we will introduce is known as kriging. This method estimates a voxel's value by considering a neighborhood surrounding the voxel and calculating a weighted average from the known values within this neighborhood. The weights are obtained by solving a system of equations known as kriging equations. These equations statistically model the spatial relationships between the known and unknown values. This method can theoretically be the best linear unbiased estimator because one can minimize the estimation error variance while keeping the estimation procedure unbiased. Like other methods, it is computationally expensive because it requires approximating the solution of the kriging equations for each unknown voxel to obtain the desired weights for the known values in the neighborhood.

As was mentioned previously, the interpolation method chosen is a linear interpolation variant. Instead of using only two points, or the eight points necessary for tri-linear interpolation, we use a variable number of points in a three-dimensional neighborhood surrounding the unknown voxel. From these points we calculate a weighted sum of the known points with weights determined by the distances of

the points to the unknown voxel. This method will be discussed in further detail in Chapter 2.

1.3 Volume Registration Overview

Once two volumes have been generated using an interpolation method, they may be aligned by applying one or a combination of multiple registration techniques. Registration is the process of determining a mathematical mapping, F , from one image to another which will align, or match, the images. Generally, image registration algorithms iteratively search a restricted domain for generating F . At each iteration, F is applied to a source image and the result is compared with a target image using a similarity function, S . When a transformation, F , is determined which maximizes S within a given tolerance, the procedure stops and outputs F . Several techniques for medical image registration have been discussed in the literature, and a good classification system is given by van den Elsen [19]. In this section, we review van den Elsen's work and define terms used in classifying registration techniques. We then provide an introduction and taxonomy of several different methods indicating which method we have chosen.

1.3.1 Registration Classification

Van den Elsen [19] provides several categories for differentiating between registration techniques. These categories are 1) origin of image properties, 2) level of

interaction, 3) domain of transformation, 4) dimensionality, 5) elasticity of transformation, and 6) parameter determination. The first of these categories, origin of image properties, differentiates between techniques which register images based on information which is extrinsic to the image and registration techniques based on intrinsic properties of the image. An example of an extrinsic image property is a warping of an image due to a known, measurable lens defect, whereas an intrinsic property is one based on an internal quality of the image itself such as edges or polygons which can be abstracted from the image. For the purposes of this thesis, we confine ourselves to registration methods based on intrinsic properties.

The second category, level of interaction, refers to the amount of user input necessary during the registration process. Three levels of interaction are given: automatic, semi-automatic, and interactive. A fully automatic registration method would determine the transformation needed to match the images without any human interaction. Semi-automatic methods require some user input during the registration process, such as initial parameter values for the registration transformation, selection of image properties to be matched, or termination points for the algorithm. However, semi-automatic methods need no further input to determine the actual transformation. With interactive methods, user input is required to determine the transformation. In this thesis, we restrict ourselves to automatic and semi-automatic methods.

Domain of transformation, which is the third category, refers to how much of

an image is affected when one or more of the parameters to the mapping function are changed. This domain can be either global or local. A global transformation affects the entire image whereas a local transformation changes a part or subsection of the image.

The fourth category is dimensionality and refers to the number of dimensions of the image that is to be registered. Current methods are typically 2-D or 3-D; matching slices of a volume or the entire volume itself. If a 2-D method is chosen, the slices must be obtained from exactly the same plane with respect to the patient's coordinate system.

Elasticity of transformation is the fifth category and refers to the degree of warping that occurs during the transformation to match the images. It can be designated rigid, affine, projective, or curved. If there is no warping at all, i.e. the distance between any two points in an image is preserved by the mapping, then the transformation is considered to be rigid. Transformations of this type are rotations, reflections, and translations. 2-D rigid transformations can be represented using four parameters, t_x , t_y , s , θ as follows:

$$\begin{vmatrix} x_2 \\ y_2 \end{vmatrix} = \begin{vmatrix} \cos \theta & -\sin \theta \\ \sin \theta & +\cos \theta \end{vmatrix} \begin{vmatrix} x_1 \\ y_1 \end{vmatrix} + \begin{vmatrix} t_x \\ t_y \end{vmatrix}.$$

Here (x_1, y_1) represents a point in the original image, (x_2, y_2) is the resulting point in the transformed image, θ is the angle of rotation, and t_x and t_y are the x and

y translations respectively.

The general case rigid transformation is the affine transform. Here, a mapping, F , is affine if $F(x) - F(0)$ is linear. For a mapping, M , to be linear, $M(x_1 + x_2) = M(x_1) + M(x_2)$, and for every constant c , $cM(x) = M(cx)$. Affine transformations map straight lines to straight lines and preserve parallelism of lines, however angles and lengths are not preserved. 2-D affine transformations can be formulated as follows:

$$\begin{vmatrix} x_2 \\ y_2 \end{vmatrix} = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \begin{vmatrix} x_1 \\ y_1 \end{vmatrix} + \begin{vmatrix} t_x \\ t_y \end{vmatrix}$$

where

$$\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}$$

is any real-valued matrix. Affine transformations include the class of rigid transformations as well as non-uniform scaling and shearing.

A projective transformation simulates the distortion caused by viewing a 3-D image through an optical system. Again, straight lines are mapped to straight lines, however parallelism of lines is not preserved. A 2-D projective transformation is formulated using a 3-D coordinate system as follows:

$$\begin{vmatrix} x_2 \\ y_2 \end{vmatrix} = \begin{vmatrix} \frac{u}{w} \\ \frac{v}{w} \end{vmatrix}$$

where

$$\begin{vmatrix} u \\ v \\ w \end{vmatrix} = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} \begin{vmatrix} x_1 \\ y_1 \\ 1 \end{vmatrix}.$$

Projective transformations are a superset of the class of affine transformations.

The most general transformation is the curved transformation. A typical formulation of a curved transformation uses polynomial functions. In 2-D this can be represented as follows:

$$x_2 = a_{00} + a_{10}x_1 + a_{01}y_1 + a_{20}x_1^2 + a_{11}x_1y_1 + a_{02}y_1^2 + \dots$$

$$y_2 = b_{00} + b_{10}x_1 + b_{01}y_1 + b_{20}x_1^2 + b_{11}x_1y_1 + b_{02}y_1^2 + \dots$$

Curved transformations can map any straight line onto a curve. They are a super-class of all the previously mentioned transformations, which through inspection can be seen to be polynomial transformations of low degree. Figure 1.1 gives a graphic representation of the effect of the different types of transformations.

Parameter determination is the sixth and last classification we will discuss. It refers to the method used to compute parameters for the image matching transformation, and can be direct or search-oriented. A direct method computes the parameters for the transformation in a straightforward manner whereas a search-oriented method uses an initial guess for parameter values and iteratively searches

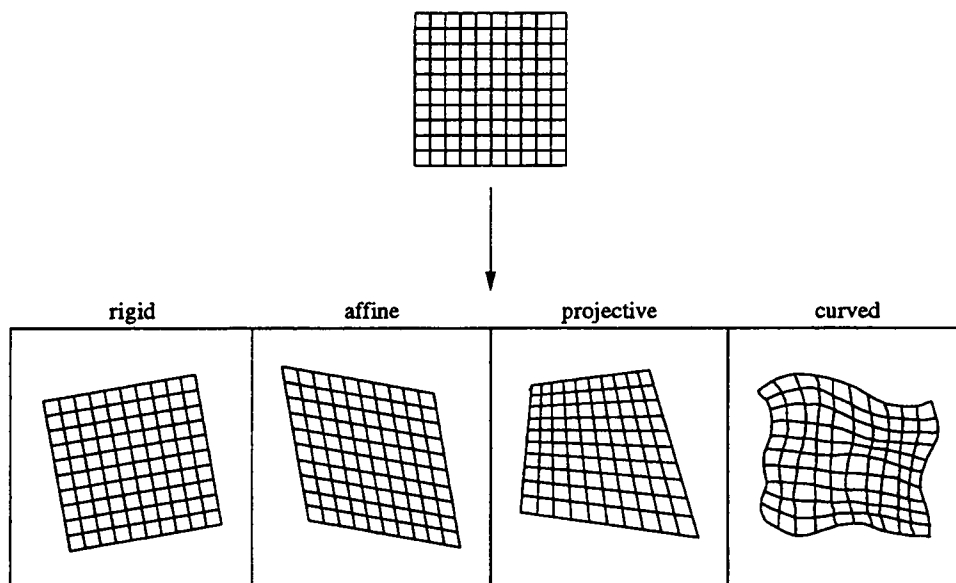


Figure 1.1: Effect of transformations on image.

Adapted from P. A. van den Elsen, E. J. D. Pol, and M. A. Viergever. Medical image matching – a review with classification. *IEEE Engineering in Medicine and Biology*, 12:26–39, March 1993.

the parameter space for the best transformation. A similarity measure is calculated at each iteration to determine the best match.

1.3.2 Registration Algorithms

We now turn our attention to algorithms developed to register medical images. Several methods will be given a brief description and will be classified using the terms defined above. All methods discussed can be categorized as intrinsic and automatic or semi-automatic. We first look at a direct method which computes a projective transform for 2-D medical images. This global method given by Barber [5] utilizes low degree polynomials to approximate a general polynomial

transform. For two images whose intensities are denoted by the functions f and m , both the values of f and m as well as the values of f' and m' are computable for each pixel. Using a set of these values in conjunction with the polynomial transform equations and their derivatives one can create an overdetermined system of linear equations provided that enough independent pixels are used. By solving the linear system, the parameters to the polynomial transform can be determined, and thus the registration mapping is determined directly.

Several methods provide a rigid, global transform for both 2-D and 3-D images using search-oriented techniques. The first method we will discuss, presented by van den Elsen et al. [18], uses the L_{vv} operator to extract ridges from both the source and target image functions. The current estimation of the registration mapping is applied to the source image and the cross-correlation of the result with the target image is computed. If the cross-correlation is maximized within a specified tolerance, the algorithm terminates, otherwise the registration parameters are altered and the similarity is recomputed. In 2-D the L_{vv} operator computes the second derivative of the image intensity in the direction v , where v and w are base vectors which form a local, orthogonal, right-handed coordinate system at the current pixel being evaluated. The effect of this operator is to highlight ridges in the image intensity function. The image correlation formula used to compare the ridgeness of the images is often used in image registration and is worth noting

here. This formula is

$$C = \sum_x f(x)T(g(x)) \quad (1.1)$$

where f and g are image intensity functions, T is the current transform to be tested, and x represents the coordinate vector of a pixel in both images.

Another method designed for globally registering both 2-D and 3-D images using search-oriented techniques and a rigid transform is given by Woods et al. [20]. This method forms equivalence classes from both the source and target images based on pixel intensities or ranges of intensities. For corresponding equivalence classes, the standard deviation of pixel intensities between the source and target image is computed, and then a weighted average of these standard deviations is determined. A search of rigid transformations is conducted until a transform applied to the source minimizes the mean of the standard deviations within a given tolerance.

A similar method within the same category is given by Ardekani et al. [2] which also divides the image pixels into individual classes. This method uses an algorithm called K-means clustering to segment both the target image and the image resulting from applying the current estimate of the registration transform to the source image. The K-means clustering algorithm statistically matches pixels in an image to one of K classes by minimizing the standard deviation of pixel intensities within each class. As above, the transform which minimizes the

standard deviations of pixel intensities between corresponding classes is selected.

Pelizzari et al. [17] have described a method which also employs a search-oriented technique to determine a global, rigid transform for 3-D brain images. To determine the registration transform, one volume is considered to be a 'head' and the other is thought of as a 'hat.' The hat is moved about the head using a rigid transform until the best fit is determined. The head consists of a stack of prisms obtained by determining the exterior boundaries of cross sections of the skull. The hat is determined similarly, however it consists of individual points lying on the surface of the skull. When the mean squared distance between the points in the hat and their nearest neighbors on the head is minimized, the algorithm terminates and the appropriate transform is reported.

A method called, chamfer matching, which computes a global, 2-D, projective transform using a search-oriented technique is presented by Borgefors [7]. This method computes two secondary images based on the actual images to be registered. The first is called a distance image and is created by extracting the edges from the target image and setting each edge pixel to zero. All other pixels in this image are assigned a value which represents their distance from the nearest edge pixel. The second image is formed from the source image by extracting the edges of the image and then selecting pixels along these edges to form a polygon. This image, called the polygon image is superimposed on the distance image. The values of pixels in the distance image that correspond to the polygon are aver-

aged. An average of zero would indicate a perfect match. A search strategy is implemented to determine the projective transform that results in the best match.

Another method in the same category as chamfer matching is given by Maitre and Wu [15] and involves dynamic programming to search for the best registration transform. This method segments edges from both the source and target image. These edges are then broken down into a sequence of primitives consisting of one or more pixels. The sequence resulting from the source image is then matched with the target sequence using dynamic programming. When breaks occur in the sequence, a temporary storage, designated the 'refrigerator,' is used to hold virtual nodes which are used to extend the sequence through the break. Once the two sequences are matched, the registration transform is determined from how the nodes correspond.

Several papers by Bajcsy et al. [8, 4, 3, 11] discuss a method which allows for elastic deformation of an idealized atlas to a source image. This method, called elastic matching minimizes the following cost function:

$$Cost = cost(deformation) - cost(similarity)$$

where the cost of deformation represents the internal forces in an object resisting deformation and are modeled by a set of partial differential equations. The cost of similarity represents the external forces acting on an object computed with a

similarity function. The method models the brain atlas as a 3-D piece of rubber. The similarity function is used to compute forces needed to be applied to the model so that the atlas will match the target image. Using the deformation equations, the model is allowed to reach equilibrium, balancing the external and internal forces. The process is repeated until similarity at equilibrium is within a given tolerance.

The last registration method we discuss is the principal axes method [12] which creates a 2-D or 3-D, rigid, global transform using a direct calculation. This method computes the eigenvectors, or principal axes, of each image's cross correlation matrix and uses these vectors to align and rotate the source image to match the target as shown in Figure 1.2. One problem encountered is that if the volumes to be registered represent slightly different sections of the anatomy then error in registration is likely to occur. In an attempt to solve this problem, Dhawan et al. [10] extended principal axes registration by iteratively reducing the size of the larger volume. At each iteration the volumes are registered and the larger volume is reduced slightly. The algorithm continues until the size of the volumes match.

The principal axes method which will be described in greater detail in Chapter 3 was selected to solve the problem stated in this thesis because it is very fast compared to search oriented techniques and is somewhat tolerant of small inaccuracies between images. Our goal for registration is to quickly find an approximate match between two volumes. Though these volumes might be medical

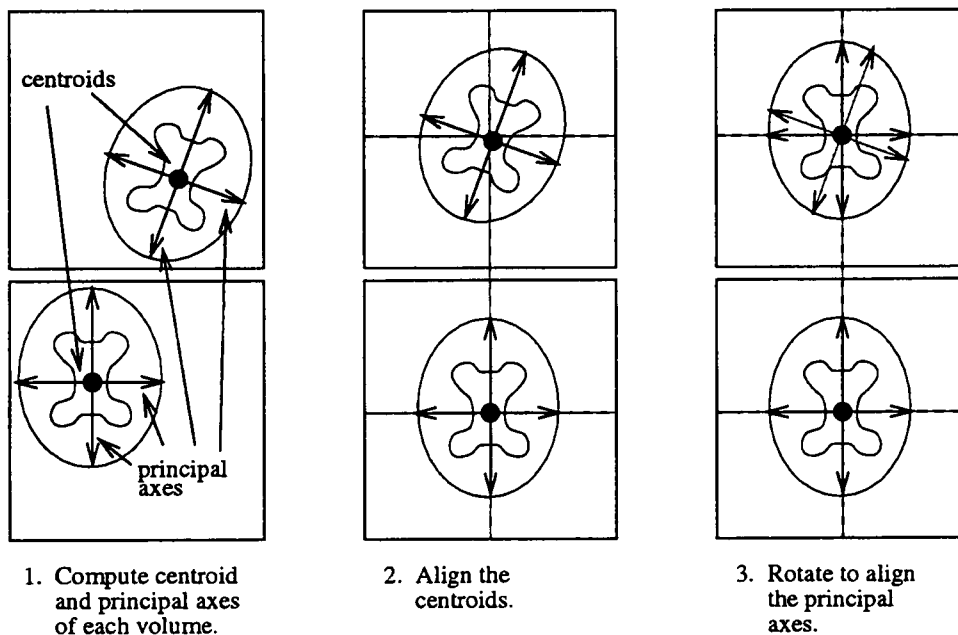


Figure 1.2: Steps in principal axes registration.

scans of the chest or abdomen where movement of the patient or internal organs might preclude an exact match, principal axes registration is not likely to fail completely. On the other hand, the error involved in employing methods using correlation as a similarity measure or computing edges within a volume might prevent a similarity measure from ever being maximized. This would cause a search-based algorithm to fail to converge to a unique transformation. Registration using a polynomial method was avoided because the image inaccuracies would likely create large errors in the registration mapping, and elastic matching was avoided because determining multiple transformations for individual local regions makes it more computationally expensive than other algorithms. There is the possibility of the principal axes method failing to determine a unique transforma-

tion, however this is very unlikely as it would require the data to be completely uncorrelated along more than one set of three orthogonal axes in the volume.

1.4 The DICOM 3.0 Standard

The images being utilized for interpolation and registration are stored in the format specified by the Digital Imaging and Communication in Medicine (DICOM) 3.0 standard [13, 1]. This specification was jointly developed by the American College of Radiology (ACR) and the National Electrical Manufacturers Association (NEMA). The purpose of the standard is to establish a uniform standard for the transfer of data definitions and information objects between imaging systems. It allows equipment and applications which conform to the standard to easily exchange information and data and is designed to allow additional services and information classes to be added. In this section, we explore some of the reasons why the standard was created as well as give a brief history of how it was established. We then discuss the organizational structure imposed by the standard and what effects this has on our interpolation/registration implementation.

In the 1980s, CT and MR data were stored on floppy disks and magnetic tape. The data on these mediums was not easily accessible by the customer because of multiple storage formats. Accessing and manipulating the data on different devices became a frustrating and time-consuming task. In order to increase hospital

productivity, a uniform format needed to be established, however, the scope of the task was very large and needed support from industry which manufactured medical imaging devices. Because of this, ACR requested that NEMA act as an industry representative and cooperate with them in development of a communications standard. The first ACR-NEMA standard, Ver. 1.0, was released in 1985. Then, in 1991, NEMA joined the ANSI Healthcare Information Standards Planning Panel and Ver. 2.0 of the standard was designed to be conformant to the Open System Interconnection (OSI) standard developed by the International Standards Organization (ISO). Demonstration software was needed, so the Radiological Society of North America (RSNA) contracted the Washington University, Mallinckrodt Institute of Radiology to develop it. This software was exhibited at the RSNA annual meetings in 1992 through 1995. Numerous companies also developed software and applications conformant to the standard and displayed these at the RSNA meetings. After the second demonstration in 1993, NEMA members voted and passed the DICOM 3.0 standard.

The DICOM standard effects the interface between different devices and medical imaging applications. It has been designed to allow point-to-point connections where all software and hardware is DICOM conformant. Because of the proliferation inter-network communication, the standard also specifies communication using TCP/IP or OSI networks. In a layered model of network communication, the DICOM implementation layer lies directly above the OSI and TCP/IP layers

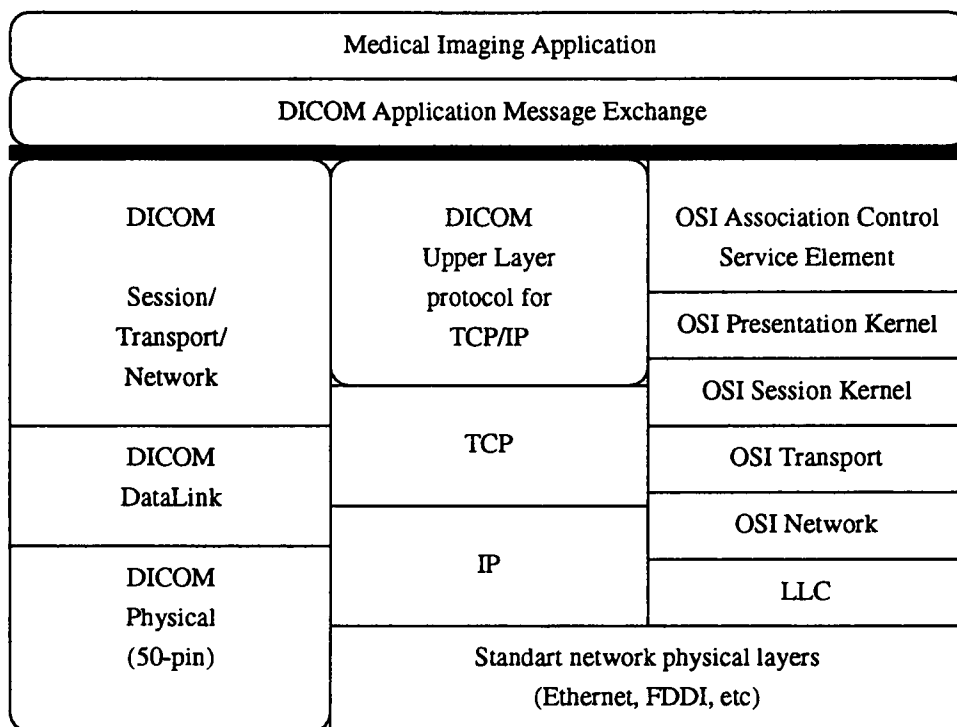


Figure 1.3: Network model of DICOM.

Adapted from S. C. Horiil, F. W. Prior, W. D. Bidgood, Jr., C. Parisot and G. Claeys.
 DICOM: An Introduction to the Standard, http://www.xray.hmc.psu.edu/dicom/dicom_intro/DICOMIntro.html,
 1996

and provides a message exchange system between a medical imaging application and the network layers (Fig. 1.3).

The standard specifies an object oriented implementation and can be divided into information classes and service classes. An information object contains image data as well as attributes such as patient name, physician name, acquisition method, etc., whereas service classes provide network image transfer, image study management, and print management. For the purpose of our interpolation and registration, only the information object attributes were accessed. The specific

attributes obtained from the DICOM image class are the following:

- Slice thickness.
- Image position.
- Image orientation.
- Slice location.
- Pixel spacing.
- Number of rows.
- Number of columns.
- Bits allocated.

These attributes deal specifically with the positioning and characteristics of a CT, MR, or PET image. The image position designates the x , y , and z coordinates of the upper left and corner of the image, and the image orientation specifies the direction cosines, $\cos(\alpha)$, $\cos(\beta)$, and $\cos(\gamma)$, of the first row and first column of the image. Both of these attributes are given with respect to the patient coordinate system. Figure 1.4 shows graphically how these attributes describe the position and orientation of an image.

The attributes that were selected are those which allow one to determine the location and intensity of any pixel in an image with respect to the patient coordinate system. This is necessary in the interpolation step before two volumes

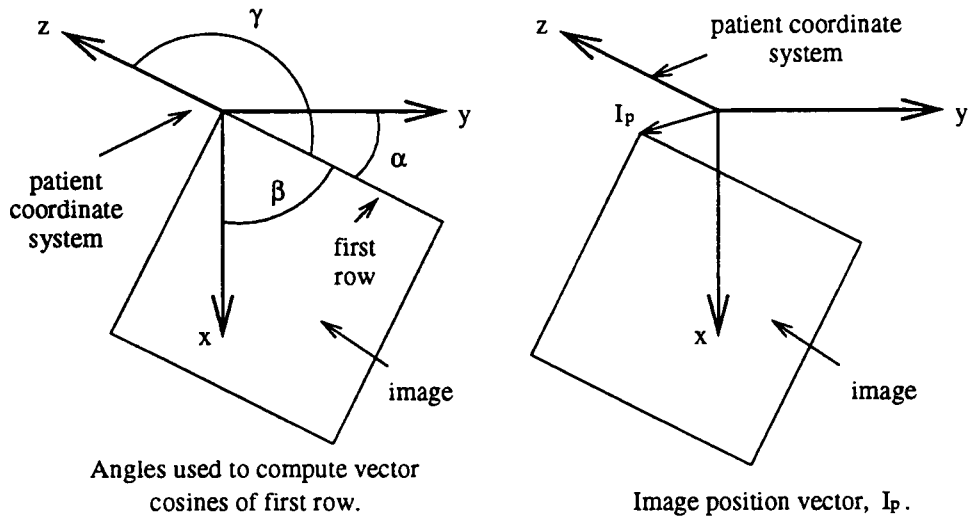


Figure 1.4: Position and orientation of image with respect to the patient.

are registered. After the registration process is complete, the ultimate goal is to output the new images in the DICOM format so that they will be accessible by other applications and storage devices.

1.5 Overview

Several methods of image interpolation have been presented. In Chapter 2 we explore the algorithms in greater detail paying close attention to the Modified Linear Interpolation technique which is our method of choice. We describe the mathematics necessary to implement it and then summarize the procedure itself.

In Chapter 3 we examine image registration. We choose three of the algorithms for further development and then proceed with a discussion of the principal axes method. We explain the computation of the principal axes transform using an

image's covariance matrix and the eigenvectors of this matrix, followed by an outline of the algorithm we have implemented.

The results of the interpolation and registration steps are discussed in Chapter 4. Methods for error analysis are presented followed by an assessment of the accuracy of the principal axes method.

Chapter 5 provides the conclusion for this thesis. The procedures and data are summarized and possible extensions to this work are suggested.

Chapter 2

Volume Interpolation

In Chapter 1, we introduced several methods of interpolating points from surrounding voxels. These methods can be used in two or three dimensions and each has features relevant to our problem. We now proceed to give a more detailed explanation of these algorithms. In particular we will talk about linear, tri-linear, cubic, shape-based, and kriging algorithms. This is then followed by a discussion of how our interpolation is actually implemented.

2.1 Interpolation Methods

2.1.1 Linear and Tri-Linear Interpolation

We begin by first considering linear interpolation in a 2-D scenario. This method of interpolation estimates the value of an unknown pixel, $\hat{p}$, represented by a

vector of the form $[x, y]$. If $\hat{p}$ lies on a straight line between two known pixels, p_1 and p_2 , we can compute the linear function, $\hat{f}$, passing through all three points which approximates the actual intensity function, f . We base $\hat{f}$ on the intensities $\hat{f}(p_1) = f(p_1)$ and $\hat{f}(p_2) = f(p_2)$. Let

$$\hat{f}(p) = ax + by.$$

Substituting in p_1 and p_2 and solving the resulting system for a and b we obtain $\hat{f}$. Then we solve for the intensity level of $\hat{p}$ by calculating $\hat{f}(\hat{p})$. The extension of this equation to 3-D involves adding a cz term to its right side and solving the system of three equations for a , b , and c .

If the distances from p_1 to $\hat{p}$ and p_2 to $\hat{p}$ are d_1 and d_2 respectively, then the linearly interpolated intensity of $\hat{p}$ can be found using the formula

$$\hat{f}(p) = w_1 f(p_1) + w_2 f(p_2) \tag{2.1}$$

where

$$w_1 = \frac{\frac{1}{d_1}}{\frac{1}{d_1} + \frac{1}{d_2}} = \frac{d_2}{d_1 + d_2} \tag{2.2}$$

$$w_2 = \frac{\frac{1}{d_2}}{\frac{1}{d_1} + \frac{1}{d_2}} = \frac{d_1}{d_1 + d_2}. \tag{2.3}$$

This formula can be easily extended to incorporate several voxels and will be

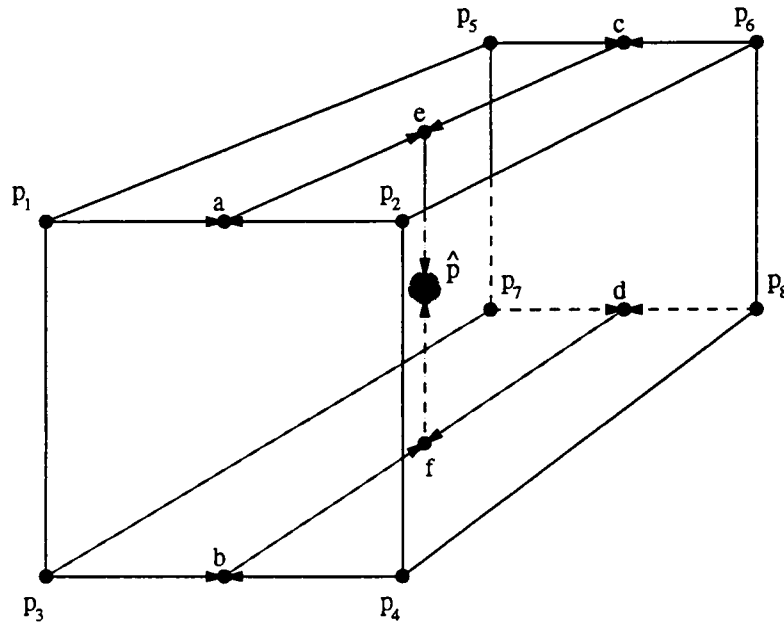


Figure 2.1: Tri-linear interpolation.

Adapted from R. W. Parrott, M. R. Stytz, P. Amburn, and D. Robinson. Towards statistically optimal interpolation for 3-D medical imaging. *IEEE Engineering in Medicine and Biology*, 49–58, September 1993.

used later in this chapter in the description of the interpolation method that was implemented.

A different modification to the simple linear interpolation method is called tri-linear interpolation. This method uses 8 points at the corners of a rectangular prism surrounding the unknown voxel. Let these points be p_1 through p_8 as shown in Figure 2.1. Tri-linear interpolation estimates the values of points a , b , c , and d using p_1 and p_2 , p_3 and p_4 , p_5 and p_6 , and p_7 and p_8 respectively with linear interpolation. Then point e lying halfway between a and c and point f lying halfway between b and d are then similarly determined. Finally $\hat{p}$ is determined by interpolation on points e and f .

Because of the simplicity of the computation, both linear and tri-linear interpolation are very fast. However, linear interpolation uses only two data points to determine its unknown voxel, so little image information is incorporated in the interpolation process. Tri-linear interpolation incorporates eight points but must calculate the values of six additional points to simply determine one voxel value. For our interpolation algorithm we seek a method which applies several points directly to the unknown value.

2.1.2 Cubic Interpolation

Cubic interpolation changes linear interpolation by factoring in the slope of the image intensity function at the known pixels when computing the value of an unknown pixel. This is done by calculating a cubic approximation, $\hat{f}$, at point, $\hat{p}$, instead of a linear function. Again, we are approximating the intensity function, f , between the known points p_1 and p_2 . However, to compute the cubic approximation function, we rely not on the coordinates of p_1 and p_2 , but on the distances, d_1 and d_2 , of these points to $\hat{p}$ respectively. This allows the following derivation to be used in both the 2-D and 3-D cases. All three points must be collinear and we are also required to approximate the slopes $f'(p_1)$ and $f'(p_2)$ in the direction of $\hat{p}$.

Let x represent the distance of a point from p_1 along the line between p_1 and p_2 , and let $x_1 = 0$ (the distance from p_1 to itself) and x_2 be the distance from p_1

to p_2 . Then

$$\hat{f}(x) = a + b(x_2 - x) + c(x_2 - x)^2 + d(x_2 - x)^3 \quad (2.4)$$

where we impose the following two conditions for a cubic spline having clamped, or restricted, boundry conditions with two nodes as presented by Burden [9]:

1. $\hat{f}(x_1) = f(x_1), \hat{f}(x_2) = f(x_2)$
2. $\hat{f}'(x_1) = f'(x_1), \hat{f}'(x_2) = f'(x_2).$

Now, computing the derivative of $\hat{f}(x)$ we get

$$\hat{f}'(x) = -b - 2c(x_2 - x) - 3d(x_2 - x)^2 \quad (2.5)$$

so, letting $x = x_2$, and substituting into Equations 2.4 and 2.5 we have

$$a = \hat{f}(x_2) = f(x_2)$$

$$b = -\hat{f}'(x_2) = -f'(x_2).$$

Letting $x = x_1$ in Equations 2.4 and 2.5, we get

$$\hat{f}(x_1) = f(x_1) = a + bx_2 + cx_2 + dx_2$$

$$\hat{f}'(x_1) = f'(x_1) = -b - 2cx_2 - 3dx_2^2$$

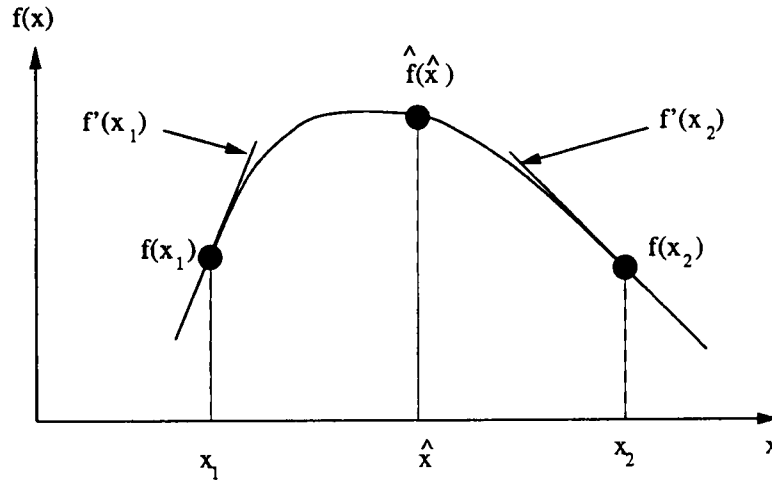


Figure 2.2: Cubic interpolation.

which we solve for c and d to get

$$c = \frac{3f(x_1) - 3a - 2bx_2 + f'(x_1)x_2}{3 - 2x_2^2}$$

$$d = \frac{-f'(x_1) - b - 2cx_2}{3x_2^2}$$

thus obtaining the needed parameters for $\hat{f}(x)$. An example of computing $\hat{f}(x)$ between x_1 and x_2 is shown in Figure 2.2.

Like linear interpolation, cubic interpolation uses only two values to estimate the intensity of an unknown voxel. In the same manner that we obtain tri-linear from linear interpolation, cubic interpolation can be extended to tri-cubic interpolation. However, for the same reasons that we rejected linear and tri-linear interpolation, we also choose not to apply cubic or tri-cubic interpolation to our problem. The reason they are presented here is that cubic interpolation can be

used effectively in a 3-D scenario with several known voxels if sets of collinear points are easy to determine. We will describe this case in Section 2.2.1 when we explain the implementation chosen for our problem.

2.1.3 Shape-Based Interpolation

As was previously mentioned in Chapter 1, shape-based interpolation integrates boundary information within an image to select which known voxels will be used to compute an unknown voxel. In brief, shape based interpolation segments an image into regions inside and outside a chosen boundary structure. When the boundary crosses a region of unknown pixels, the location of the boundary is estimated, and voxels are interpolated from known voxels on the same side of the boundary. We now describe the steps shown in Figure 2.3 required to accomplish this task.

The first step is to segment the original image into a binary image, S_b . This segmentation process must identify a target structure or gray level within the image that has significant meaning. An example of this is to threshold a CT image where the grey levels of different tissue types are known for bone structures. This would create a sharper contrast along bones in the resulting image.

The second step is to map S_b onto a distance data set, S_g , corresponding to the known voxels of S_b . For each known voxel in S_b , the minimum distance to the boundary of the target structure is computed. The corresponding data point in

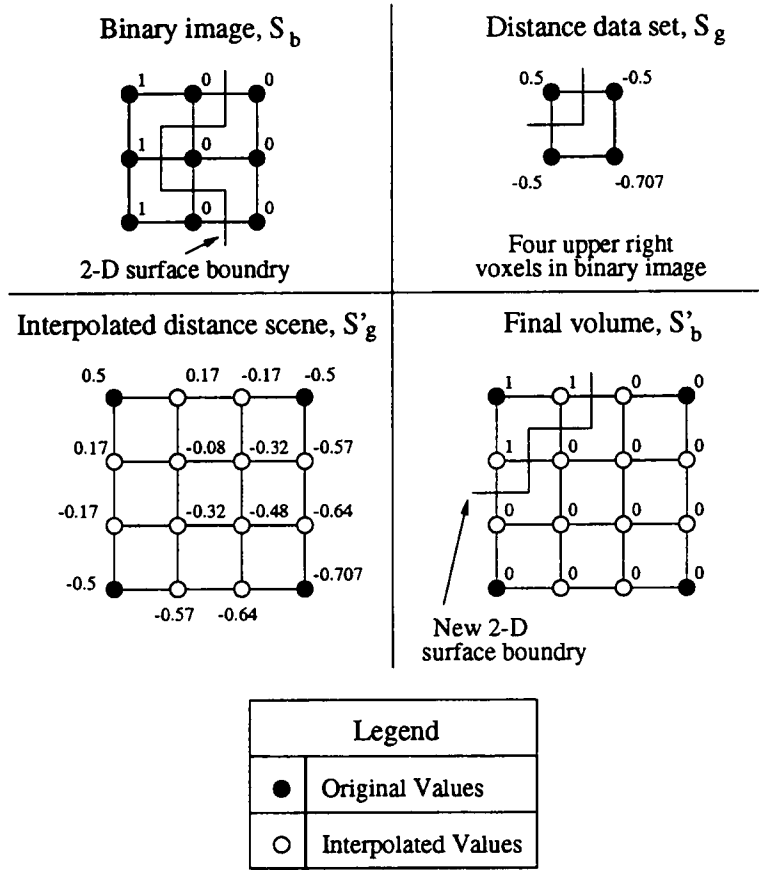


Figure 2.3: Shape-based interpolation.

Adapted from R. W. Parrott, M. R. Stytz, P. Amburn, and D. Robinson. Towards statistically optimal interpolation for 3-D medical imaging. *IEEE Engineering in Medicine and Biology*, 49-58, September 1993.

S_g is given the value of the computed distance, and is made negative if the point lies outside the specified structure, i.e. the pixel value in S_b is 0.

Following the second step, the third step is to create an interpolated distance scene, S'_g , with voxels corresponding to both known and unknown voxels of S_b .

The values of S'_g are interpolated from the known distances in S_g .

The fourth and last step is to create a final binary volume, S'_b , whose values

are determined by the corresponding distances of S'_g . If the distance is positive, the voxel is given the value 1, if negative, it is assigned the value 0. From this binary scene, it can be determined whether a unknown voxel should be given the value of voxels inside or outside the boundary.

This form of interpolation would work well on images with few intensity levels and distinct boundaries. However, in our circumstances we have numerous intensity levels often gradually increasing or decreasing along a given direction in the image. Shape-based interpolation might still be very effective if used in conjunction with a different algorithm. Often, between slices, other methods cause much blurring on the edges of distinct structures such as bones or organs. Shape-based interpolation could be used to sharpen these edges by determining which side of a boundry a voxel should be interpolated from, and then a different method could be used to actually interpolate the value from known voxels on the same side of the boundary.

2.1.4 Kriging

Kriging is an interpolation method which incorporates an arbitrary number of nearby voxels in the calculation of an unknown voxel. The basic kriging equation computes a weighted sum of the neighboring voxels. Given that $Z_i(p_i)$ is the image intensity at voxel p_i ,

$$\hat{Z}(p) = \sum_{i=1}^n w_i Z_i(p_i) \quad (2.6)$$

where $\hat{Z}(p)$ is the estimated value of unknown voxel p , and w_i is the weight applied to the intensity of voxel p_i . In order to make kriging a theoretically unbiased linear estimator, we impose the conditions

$$E(\hat{Z} - Z) = 0 \quad (2.7)$$

$$\min E(\hat{Z} - Z)^2. \quad (2.8)$$

Here E is the expected mean, Z is the actual value at point p , and $\hat{Z} - Z$ represents the estimation error. By substituting Equation (2.6) into Equations (2.7) and (2.8) we have

$$E \left[\sum_i w_i Z_i(p_i) \right] - E(Z) = 0 \quad (2.9)$$

$$\sigma_e^2 = 2 \sum_i w_i \kappa(p_i, p) - \kappa(p, p) - \sum_i \sum_j w_i w_j \kappa(p_i, p_j) \quad (2.10)$$

respectively where σ^2 is the variance and p is the point where the estimated value is to be computed. Also, $\kappa(m, n)$ represents the covariance between m and n which is modeled by a function called a semivariogram, $\gamma(m, n)$. Usually $\gamma(m, n)$ is estimated by structural analysis of the image and can be represented as $\gamma(h_{mn})$. By substituting in γ and a Lagrangian multiplier, η , and then minimizing the system we obtain the equations

$$\sum_j w_j \gamma(h_{ij}) + \eta = \gamma(h_{ip}) \quad (i = 1, \dots, n)$$

$$\sum_j w_j = 1.$$

We obtain the weights for the kriging equation by solving this system.

Kriging is an effective interpolation algorithm because it utilizes several voxels to compute the value of an unknown voxel and because it is able to utilize structural attributes of the image through the semivariograms. However, computation time is large because determining each unknown voxel requires solving the system of equations given by (2.9) and (2.10). We desire a faster and simpler approach, so instead of using the statistical method of generating the weights, our implementation uses the distance technique of Section 2.1.1, combining it with the basic kriging equation which utilizes a neighborhood of points. We describe this method in the next section.

2.2 Modified Linear Interpolation

In Section 2.1.1, Equations (2.1), (2.2), and (2.3) show that the distances of known voxels from an unknown voxel can be used to determine weights for linear interpolation. As was noted at the end of Section 2.1.4, these weights can then be used in the basic kriging equation instead of the statistically generated weights involved in kriging. This idea can be realized so that an unspecified number of voxels can be used in the interpolation with the ability to add additional voxels at a later time. This is the method chosen for our implementation and we now

describe 1) how to perform this interpolation method, 2) how to compute voxel positions from information given in the DICOM image class, and 3) the overall implementation strategy utilized.

2.2.1 Voxel Intensity Computation

The modified linear interpolation method in a fashion similar to numerical quadrature computes an estimated image intensity, $\hat{f}(\hat{p})$ at an unknown point $\hat{p}$ using the equation

$$\hat{f}(\hat{p}) = \frac{\sum_{i=1}^n w_i f(p_i)}{\sum_{i=1}^n w_i}$$

where $f(p_i)$ is the value of the image intensity function at point p_i , n is the number of known points in the given neighborhood, and w_i is simply the reciprocal of the distance from point p_i to $\hat{p}$.

The following steps describe the application of one known value to the estimated sum, $\hat{f}(\hat{p})$. These are repeated for each known point within the neighborhood, and at any iteration the current value of $\hat{f}(\hat{p})$ reflects the modified linear average of all applied voxels.

Algorithm to add $f(p_i)$ to the weighted sum, $\hat{f}(\hat{p})$:

1. Determine if voxel p_i is within the specified neighborhood surrounding $\hat{p}$. If so, continue, otherwise end.

2. Calculate the distance, d_i , from the center of voxel p_i to the center of p .
3. Compute the new value of $\hat{f}(\hat{p})$ using

$$w_i = \frac{1}{d_i}$$

$$\hat{f}(\hat{p}) = \frac{w_c p_c + w_i p_i}{w_c + w_i}$$

where p_c is the value of $\hat{f}(\hat{p})$ from the previous iteration or 0 if this is the first voxel applied, and w_c is the value of w calculated from step 4 of the previous iteration or 0 if this is the first voxel applied.

4. Compute w , the current sum of the weights of all voxels currently applied to $\hat{p}$ using

$$w = w_c + w_i.$$

This will be used in the next iteration as the new value for w_c .

One benefit of using this form of linear interpolation is that it uses information from several surrounding voxels to calculate an unknown voxel. Because little computation is needed, a second benefit is the speed with which the algorithm can be performed. A third benefit which greatly simplifies the use of this algorithm is its ability to maintain a current weighted average so that the number of pixels applied can be arbitrary. The result of this benefit will be seen in Section 2.2.3, where the overall implementation strategy is discussed.

2.2.2 Computing Voxel Positions

Computing the weighted linear sum is a relatively easy task, however, to do so, one must be able to calculate the distance between a pixel in the source data image and a voxel in the target volume. To simplify this calculation, the distance is computed from the centers of the pixel and voxel. The positions of the pixel and voxel are computed with respect to the patient coordinate system and then the distance formula is applied. Because the position of the voxel with coordinates $(0,0,0)$ in the target volume is known with respect to the patient, it is a very simple task to compute the position of an arbitrary volume voxel. However, the source data slices have different orientations so it is more difficult to calculate the positions of individual pixels in these images. In this section, we describe how to calculate the positions of image pixels using DICOM image attributes so that the distances needed for linear interpolation can be computed.

There are two attributes from the DICOM image class which are needed to determine the position of a pixel belonging to an image of that class. The first is the image position attribute, $I_p = [x, y, z]$, which is a vector from the origin of the patient based coordinate system to the image pixel with indices $(0,0)$. The second attribute is a pair of vectors, $R = [a_r, b_r, g_r]$ and $C = [a_c, b_c, g_c]$, which represent the direction cosines for the first row and first column of the image respectively. Let p_{ij} represent the pixel with coordinates (i, j) . Also, let P_{ij} be

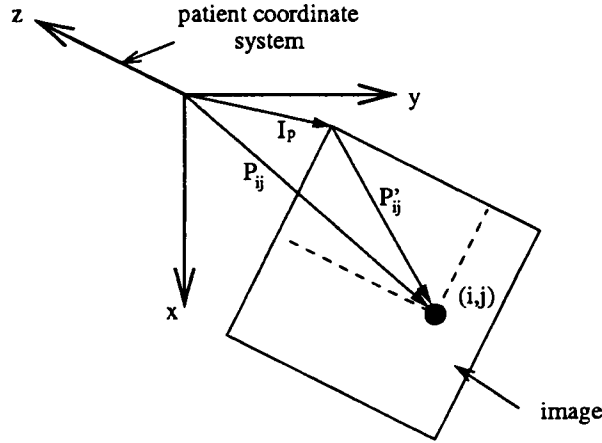


Figure 2.4: Image position vectors.

a vector representing the position of p_{ij} with respect to the patient coordinate system, and P'_{ij} be a vector representing the position of p_{ij} with respect to p_{00} . Then Figure 2.4 shows the relation of the image position vector to the patient coordinate system and to the position vectors of p_{ij} .

Let the image be translated so that the pixel p_{00} is located at the patient origin. Then Figure 2.5 represents an image with the angles, α_r , β_r , and γ_r , used to compute the direction cosines, a_r , b_r , and g_r , for its first row as indicated. The direction cosines can be used to calculate P'_{ij} , of pixel p_{ij} , using

$$P'_{ij} = \begin{bmatrix} d_i a_r + d_j a_c \\ d_i b_r + d_j b_c \\ d_i g_r + d_j g_c \end{bmatrix}.$$

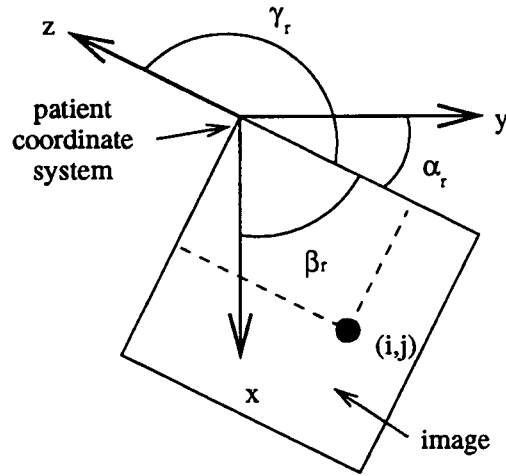


Figure 2.5: Image orientation angles.

Then P_{ij} can be computed as

$$P_{ij} = I_p + P'_{ij}.$$

The distance between P_{ij} and the position vector of nearest the target volume voxel can be now be calculated.

2.2.3 Implementation Strategy

The interpolation of a complete volume consists of two entirely separate interpolation passes. Both passes utilize the strategy presented in Section 2.1.4, however only the first pass uses the DICOM attributes explained in Section 2.2.2. We now proceed to describe these passes.

The first pass interpolates data from the source images onto the 3-D target

volume. This pass examines each pixel in a source image to determine within which target voxel its center resides. Usually the target volume has a lower resolution than the source, therefore many pixel centers will fall within the bounds of one voxel. Each pixel is added to the weighted sum of only that voxel. The weights for each voxel are maintained in a corresponding array with the same dimensions as the volume. Because of the order that the source pixels are applied to the target, a search to find all pixels applicable to one voxel does not need to be executed. Instead, the image pixels can be traversed in any order desired. The result of the first pass is a target volume that has the image slices filled in, but has gaps corresponding to the empty spaces between the source image scans.

The second pass interpolates values to fill in the gaps in the target volume. Because the necessary data for this pass is already in the volume, the source images do not need to be consulted. Instead, voxels that have not been set are determined from neighboring voxels whose values are already known. By doing this, the computation time required to determine the locations of source image pixels is eliminated. Since the voxels are in a fixed grid, it is much simpler to determine their distances with respect to each other. In fact, this can be accomplished before beginning the interpolation process.

This implementation has a computational complexity of $O(v(n + 1))$ where v represents the number of voxels in a volume and n is the size of the neighborhood used to interpolate one voxel. Because our procedure is linear with respect to

the number of voxels, it is highly efficient. However, the number of voxels in an interpolation neighborhood varies greatly, from 27 voxels to easily over 500. This parameter increases in a cubic fashion when the gaps between medical scans increase or when volume resolution is made finer. With this growth rate, the interpolation procedure can take a very long time when the original data is sparse, and the desired resolution is high.

It is possible to use different methods of interpolation in the two passes. As was mentioned in Section 2.1.1, cubic interpolation can be used in a 3-D scenario if sets of collinear can be easily determined. In the second pass of our implementation, the fact that we use data from only the target volume would allow us to extract sets of collinear points along the three axes as well as diagonally through a unknown voxel. Each set could be used to determine a estimated value for the unknown voxel using cubic interpolation, and then all the values for a given voxel could be averaged to determine the final intensity level.

Now that the interpolation process has been explained, we are ready to examine registration algorithms that can be applied. Chapter 3 first discusses three different registration methods, two which provide global registration examples and one which shows a possible local extension to the method we have selected. Following this discussion is a description of the theory and implementation of the Hotelling transform.

Chapter 3

Volume Registration

There are many registration methods that can be used to align 3-D images. In Chapter 1, we listed several different methods and categorized them according to the scheme presented by van den Elsen [19]. We now limit the scope of our discussion to only four different algorithms that we have selected which demonstrate the wide range of registration techniques. We provide greater detail and are able to highlight features common to many registration methods. In this chapter we will discuss registration by polynomial approximation, by feature extraction and cross-correlation, and by elastic matching. Following this, we will develop the principal axes or Hotelling transform which is the technique we have chosen to implement. The theory on which this transform is based will be given, followed by a discussion of our implementation which registers two 3-D volumes.

3.1 Registration Methods

We begin this chapter with a detailed discussion of registration by polynomial approximation, by feature extraction and cross-correlation, and by elastic matching. These methods show the large scope that registration techniques span. Though we cannot give a complete analysis of these methods, we will introduce the theory on which each is based, as well as discuss applicability and attributes of each. We now begin with polynomial approximation.

3.1.1 Registration by Polynomial Approximation

Determining a registration transformation by using polynomial approximation is a direct method of registering two images. Barber [5] describes the theory of this method as well as an iterative approach to its implementation. He computes a projective, global, transform on 2-D images. In this section we summarize this method and give a brief discussion of its merits.

Barber refers to the registration mapping as a coordinate transfer function (CTF). To approximate the CTF which maps an image, $m(x, y)$, onto a second image, $f(x, y)$, the polynomials

$$r = a_{11} + a_{12}x + a_{13}y + a_{14}xy \quad (3.1)$$

$$s = a_{21} + a_{22}x + a_{23}y + a_{24}xy \quad (3.2)$$

are used where the coefficients, a_{ij} , are determined through Taylor series approximation.

The first step is to extend the problem to three dimensions by considering an image function, $g(x, y, z)$, where z adds an additional dimension such that if the parameters to g are (x_1, y_1, z_1) , then the current image function is spatially identical to $f(x_1, y_1)$ for all values of x_1 and y_1 . However if the z parameter is $z_1 + \delta z$, the image function is identical to $m(x_1, y_1)$. We also have $g(x, y, z) = g(x + \delta x, y + \delta y, z + \delta z)$ where δx , δy , and δz specify the change in x , y , and z . By computing the first Taylor series approximation of g , we get

$$g(x + \delta x, y + \delta y, z + \delta z) = g(x, y, z) + \delta x \frac{dg}{dx} + \delta y \frac{dg}{dy} + \delta z \frac{dg}{dz}$$

or

$$0 = \delta x \frac{dg}{dx} + \delta y \frac{dg}{dy} + \delta z \frac{dg}{dz}.$$

Now by identifying $g(x, y, z)$ with $f(x, y)$ and $g(x, y, z + \delta z)$ with $m(x, y)$, and by rewriting these using Equations (3.1) and (3.2) as examples we get the equation

$$\begin{aligned} f(x, y) = & m(x, y) + (a_{11} + (a_{12} - 1)x + a_{13}y + a_{14}xy) \frac{df}{dx} \\ & + (a_{21} + a_{22}x + (a_{23} - 1)y + a_{24}xy) \frac{df}{dy}. \end{aligned} \quad (3.3)$$

The values of f and m as well as $\frac{df}{dx}$ and $\frac{df}{dy}$ can be determined for arbitrary

coordinates, x and y , within the images. Also, equations of the form shown in Equation (3.3) are linear in the parameters a_{ij} , so given at least eight independent pixels from f and m , we can solve the resulting system. When this linear system has been solved and values substituted into Equation (3.3), the resulting equation expresses the image f in terms of m .

As has been previously mentioned, this method is a good example of one of the few direct interpolation algorithms and is flexible in the following aspects. It is possible to iterate this procedure to obtain better results. Barber does so by repeating this technique while combining CTF's to increase the accuracy of the final mapping. This method can also be limited to an affine transform by simply dropping the xy terms from Equations (3.1) and (3.2), or it can be extended to provide a curved mapping by using polynomials of greater degree.

The similarity between the polynomial method and the algorithm actually implemented is that they both are examples of direct registration. Direct methods are often faster than search-based methods because they do not need to iteratively search for an optimal set of transformation parameters. However, with this procedure, image noise can cause a large degree of error in the transformation. One possible way to reduce this error would be to repeat the procedure with multiple sets of independent pixels so that an average set of coefficients can be determined, but doing so would increase processing time.

We now proceed with a discussion of a typical registration technique using a

search-based algorithm which compares features between images and computes the similarities of these features to select a optimal transformation.

3.1.2 Registration by Feature-Extraction

Several registration methods are global, search-based techniques, and many of these use feature extraction with cross-correlation to compute the similarity of two images after a test transformation has been applied. The method we will now discuss is described by van den Elsen [18] and uses what is called the L_{vv} operator to implement its feature extraction. We first give a brief outline of the structure of a generalized search-based registration procedure, followed by a description of the L_{vv} operator for two images, and then conclude with a summary of some features in this method common to many registration algorithms.

Most search-based algorithms follow these three steps: 1) extract features from both images, 2) select and apply the next test transformation, 3) compute similarity using cross-correlation. These steps are repeated until an acceptable transform is found. Several feature extractors have been used for step one including gradient filters, 2nd derivative filters, and basis functions. The L_{vv} operator given by van den Elsen is a type of 2nd derivative filter. Step two involves choosing the next available set of parameters in a transform's parameter space. Because this space can be quite large, it can take a long time to perform an exhaustive search for the optimal parameters, therefore van den Elsen uses what is known

as a multi-resolution pyramid. This technique involves performing registration on images whose resolution have been greatly reduced, thus eliminating much of the parameter space. Once an optimal transform has been determined, the resolution is increased, and the previous transform is used as an initial guess for the new search. Finally, in step three, the similarity between the transformed image and its target is calculated using the cross-correlation formula given as Equation 1.1.

The L_{vv} operator for 2-D extracts information about ridges and troughs in an image. It can be computed by finding the 2nd derivative of the image intensity function in a direction specified by a vector, v . Vectors v and w form base vectors to a local, orthogonal, right-handed coordinate system where w is the local gradient of the image intensity function as shown in Figure 3.1. When a pixel being considered lies on a slope, w will point up the slope in the same direction as the gradient. As a ridge in the intensity function is crossed, w changes direction to point along the same line as the ridge. When a pixel is located on the opposite side of a ridge, w then reverses its original direction. Thus, on a ridge, w lies parallel to the ridge whereas v lies in a direction perpendicular to the ridge. Because of this, the image intensity function in the v direction at a ridge pixel is generally very concave so L_{vv} has a large negative value. The opposite is true at the bottom of a trough where L_{vv} has a large positive value. For all other pixels, L_{vv} is generally close to zero, therefore it highlights ridges and troughs in an image.

The procedure described by van den Elsen has at least three features that are

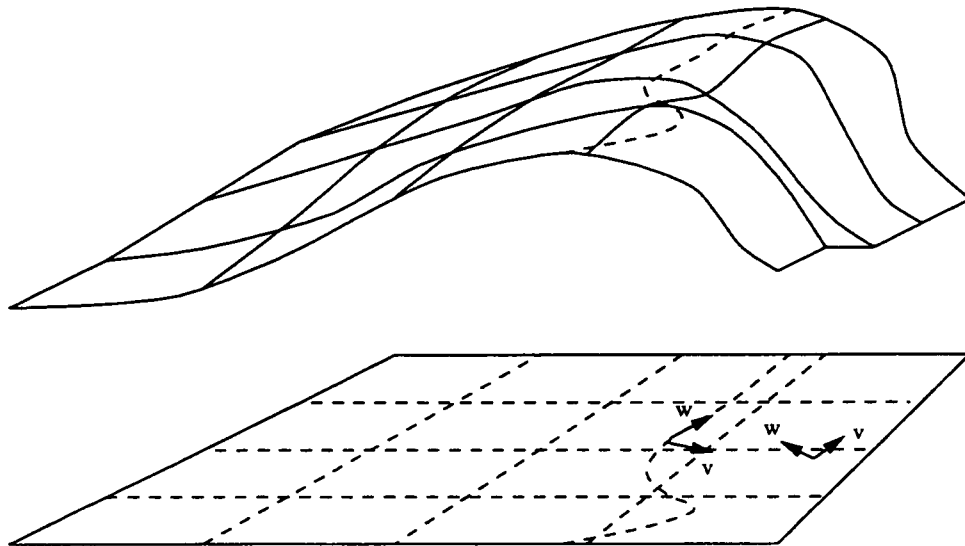


Figure 3.1: Orthogonal vectors, w and v , in computation of the L_{vv} operator. Adapted from P. A. van den Elsen, J. B. A. Maintz, E. J. D. Pol, and M. A. Viergever. Automatic registration of ct and mr brain images using correlation of geometrical features. *IEEE Transactions on Medical Imaging*, 14(2):384–396, June 1995.

common among image registration algorithms. The first is its use of a feature extractor to isolate properties that are common between the two images to be matched. This is a requirement for most image registration, and ranges from simple thresholding to much more complicated techniques such as applying the L_{vv} operator. The second commonality seen in this algorithm is the use of a multi-resolution technique which can be applied in many situations and can greatly reduce computation time by limiting the search space. The third feature common to many registration techniques is using cross-correlation as a similarity measure to compare registration results.

The next registration method differs from the previous two because it is an example of a local technique. It allows for greater flexibility in the registration

transform and can overcome problems such as warped or distorted images which cannot be easily registered with a global method.

3.1.3 Elastic Matching

Elastic matching can best be thought of as a rubber sheet or block which contains an image. Forces can be applied to the sheet or block to stretch or warp it so that it matches a second image. This method has been employed by Bajcsy et al. [8, 4, 3, 11] to match an idealized brain atlas to a specific brain image. The atlas can then be superimposed on corresponding images of different modalities to help identify specific cranial structures and regions. The method is local, therefore subsections of the atlas are modified to match local regions in the target image. In order to obtain an optimal matching, the procedure seeks to minimize the following cost equation:

$$cost = cost(deformation) - cost(similarity).$$

This shows that as energy is spent to deform the atlas the cost of deformation increases. However, as similarity increases because of the deforming process, the cost of similarity also increases and balances the deformation cost. In this section, we discuss the method used to compute the deformation costs as well as similarity costs, and then give comment regarding the relevance of this method to our own

registration implementation.

In order to compute deformation costs, one must be able to model the external forces that need to be applied to a localized region to obtain a minimized cost. The Navier Stokes equations are used by Bajcsy to accomplish this. These equations are

$$\mu \nabla u_i + (\lambda + \mu) \frac{\partial \theta}{\partial x_i} + F_i = 0 \quad (i = 1, 2, 3)$$

where θ is the cubical dilation given by

$$\theta = \frac{\partial u_1}{\partial x_1} + \frac{\partial u_2}{\partial x_2} + \frac{\partial u_3}{\partial x_3}.$$

In these equations, $x = [x_1, x_2, x_3]^T$ refers to the coordinate system of the local region before deformation, $F = [F_1, F_2, F_3]^T$ represents the external forces distributed throughout the elastic region that we wish to compute, μ and λ are Lamé's elastic constants which determine the elasticity of the body, and $u = [u_1, u_2, u_3]^T$ is the resulting displacement from applying the forces. The elastic constants allow us to specify the amount of deformation an object has when a given force is applied. With larger constants an object becomes stiffer thus resulting in a local transform similar to a rigid transform. With smaller constants the effect is the reverse and the resulting curved transform introduces greater deformation.

The cost of similarity is computed using a normalized cross-correlation func-

tion similar to that used in the previous section, and is used to find the optimal transformation. If forces proportional to the gradient vector of the similarity function are applied to the system, the similarity will increase. By applying greater or lesser forces, one can compute the local minimum of the cost function thus determining the desired transformation. To select specific attributes from the images for use in computing the cross-correlation, Bajcsy decomposes the image functions using an orthonormal system of hermite polynomials. Relevant components are then used to compute similarity. As with the matching scheme described in Section 3.1.2, a multi-resolution pyramid can be used to reduce the search space and computation time.

Elastic matching is not a viable alternative to achieve the goals in this paper because it requires a large amount of computation time to match all local regions at different resolution levels. We discuss it here because it illustrates a possible extension to the current implementation. As has been previously mentioned, some of the images that we are matching are not fixed over time, e.g. chest images which change due to movement of internal organs and respiration. Because of this, the global method we have selected cannot perfectly match some image sets. Elastic matching could be used as an additional method after global registration when a closer matching is desired.

3.2 Principal Axes Method

The algorithm we have chosen to implement is called the principal axes method [12]. It is a global and direct method, therefore there is no search for the optimal transformation parameters. Instead, they are directly computed from the image data. This method is a modification of the Hotelling transform which can be computed for a population of random vectors of the form $\vec{x} = [x_1, x_2, \dots, x_n]^T$ using

$$y = A(\vec{x} - \vec{m}_x)$$

where A is a matrix representing a linear transformation, and $\vec{m}_x$ is the mean vector $E(\vec{x})$ of the population. To give a detailed description of the theory for this transform, we first present in this section a discussion of linear transforms and eigenvector computation. Following this is a description of the determination of an image's covariance matrix. Finally, we show how this matrix is combined with eigenvector computation to derive the Hotelling transform and discuss how this transform is implemented to register the images.

3.2.1 Eigenvector Computation

Eigenvectors of specialized matrices have been discussed by Kolman [14] and can be computed as follows. An $n \times n$ matrix, A , represents a linear transformation with eigenvectors of A being defined as all vectors α such that $A[\alpha]_s = c[\alpha]_s$ where

c is a scalar and is the corresponding eigenvalue of $[\alpha]_s$. We can say that a second matrix, D , is similar to A if

$$A = P^{-1}DP \quad (3.4)$$

where P is a nonsingular matrix. Basically, D represents the same linear transformation as A but with respect to a different basis.

If D is a diagonal matrix, that is, all the entries of D are zero except on the diagonal, we say that A is diagonalizable. In this case, each of the nonzero entries of D are eigenvalues of A , and the corresponding rows of P are the related eigenvectors which form a basis for D .

It can be shown that if A is real and symmetric, then there exists an orthonormal set of eigenvectors of A such that Equation (3.4) holds. These eigenvectors (i) form an alternate basis for A , and (ii) when used as the rows of P , define a transform to convert between this basis and the original basis of A . In other words, if A is real and symmetric, it can be diagonalized with an orthonormal basis of eigenvectors. We now discuss how these eigenvectors can be computed.

Given 3×3 matrix A , the eigenvalues of A are the real roots of the characteristic polynomial of A represented by the equation

$$f(t) = \det(tI_3 - A) = 0$$

where $\det$ specifies the determinant, and I_3 is the 3×3 identity matrix. Because

A is a 3×3 matrix, the degree of f is 3, and f has the form

$$t^3 + pt^2 + qt + r = 0$$

where p , q , and r are real coefficients. We can directly solve for the roots of f in the manner given by Beyer [6] thereby obtaining the means to compute the eigenvectors as follows. We substitute $x - \frac{p}{3}$ for t to obtain

$$x^3 + ax + b = 0 \tag{3.5}$$

for f , where

$$a = \frac{1}{3}(3q - p^2)$$

and

$$b = \frac{1}{27}(2p^3 - 9pq + 27r).$$

For there to be three real and unequal roots, the condition

$$\frac{b^2}{4} + \frac{a^3}{27} < 0$$

must be satisfied. We now can let $x = m \cos(\theta)$, and f becomes

$$m^3 \cos^3(\theta) + am \cos(\theta) + b = 0. \tag{3.6}$$

By the trigonometric identity

$$4 \cos^3(\theta) - 3 \cos(\theta) - \cos(3\theta) = 0$$

it follows from Equation (3.6) that

$$\frac{4}{m^3} = \frac{-3}{am} = \frac{-\cos(3\theta)}{b}$$

from which we get

$$m = 2\sqrt{\frac{-a}{3}}$$

and

$$\cos(3\theta) = \frac{3b}{am}. \quad (3.7)$$

Now, any solution θ_1 satisfying Equation (3.7) also has the solutions $\theta_1 + \frac{2\pi}{3}$ and $\theta_1 + \frac{4\pi}{3}$. Thus the roots of Equation (3.5) are $2\sqrt{\frac{-a}{3}} \cos(\theta_1)$, $2\sqrt{\frac{-a}{3}} \cos(\theta_1 + \frac{2\pi}{3})$, and $2\sqrt{\frac{-a}{3}} \cos(\theta_1 + \frac{4\pi}{3})$.

By substituting back $x = t + \frac{p}{3}$ we obtain the eigenvalues of A . We now simply replace c with the eigenvalues in the matrix equation

$$(cI_3 - A)\alpha = 0$$

and solve the resulting linear system for the corresponding eigenvectors α . As

was mentioned previously, these eigenvectors form an orthonormal basis which can be applied to diagonalize A . We will use this fact in Section 3.2.3 along with the covariance matrix, which will be discussed next, to derive the Hotelling transform.

3.2.2 Computing the Covariance Matrix

If we have a set, X , of n dimensional random vectors of the form

$$\vec{x} = [x_1, x_2, \dots, x_n]^T$$

where the x_i are real numbers, then we can define the covariance matrix of X as

$$C_x = E[(\vec{x} - \vec{m}_x)(\vec{x} - \vec{m}_x)^T] \quad (3.8)$$

where E represents expected value and $\vec{m}_x$ is the expected value of X . Each element of C_x , c_{ij} , represents the covariance of the elements x_i and x_j from vectors in X . Because of this, $c_{ij} = c_{ji}$, and therefore C_x is a real and symmetric matrix.

We estimate the expected value of M samples of X using

$$\vec{m}_x = \frac{1}{M} \sum_{k=1}^M \vec{x}_k$$

and we compute the covariance matrix with

$$C_x = \frac{1}{M} \sum_{k=1}^M \vec{x}_k \vec{x}_k^T - \vec{m}_x \vec{m}_x^T$$

where the $\vec{x}_k$ represent individual vectors in the set X .

In our imaging application, the vectors for X will simply be the coordinates of pixels in a specified region in the image. These are obtained through image segmentation such as thresholding, and represent an object in the image which will be matched to a similar object in the target image using the Hotelling transform. We now turn our attention to deriving this transform and describing how this is used to implement principal axes registration.

3.2.3 Derivation of the Hotelling Transform

In this section we combine information presented in Sections 3.2.1 and 3.2.2 to derive the Hotelling transformation by showing that the diagonalization of the covariance matrix of a random set of vectors yields a new basis where vectors represented by this basis are uncorrelated. This implies that the new basis vectors lie along the principal axes of the random set.

From Section 3.2.2 we know that the covariance matrix, C_x , of a set of M random vectors, X , can be determined and that it is a real and symmetric matrix. We can find an orthonormal basis of eigenvectors for C_x and use them as rows in

a matrix, P . In this way we create a transform which can be used to map the vectors in X to a new orthonormal basis. A diagonal matrix, D , such that

$$D = PC_xP^{-1}$$

corresponds to the linear transform represented by C_x but with respect to the basis of eigenvectors. We now show that D is actually the covariance matrix, C_y , for the set of vectors Y where

$$\vec{y}_k = P(\vec{x}_k - \vec{m}_x) \quad (i = 1, \dots, M)$$

and where Y has a zero mean, $\vec{m}_y = 0$, because we subtract $\vec{m}_x$ before applying P . To compute C_y , we use Equation (3.8) from Section 3.2.2 and show that it is equivalent to D as follows:

$$\begin{aligned} C_y &= E [(\vec{y} - \vec{m}_y)(\vec{y} - \vec{m}_y)^T] \\ &= E [\vec{y}\vec{y}^T] \\ &= E [P(\vec{x} - \vec{m}_x)(P(\vec{x} - \vec{m}_x))^T] \\ &= PE [(\vec{x} - \vec{m}_x)(\vec{x} - \vec{m}_x)^T] P^T \\ &= PC_xP^{-1} \\ &= D. \end{aligned}$$

Here we use the fact that the columns of P form an orthonormal set. So P is orthogonal and $P^{-1} = P^T$.

Because C_y is diagonal, the covariance of any two elements of Y , $\vec{y}_i$ and $\vec{y}_j$ where $i \neq j$, is equal to zero, so the elements of Y are uncorrelated. This means that the basis vectors of Y represent the principal axes of the vectors in Y . We now show how to use this fact to register two 3-D images.

3.2.4 Principal Axes Registration

We now discuss how the Hotelling transform is modified in our implementation so that it can be used to register two 3-D images. When dealing with image volumes, we can consider the coordinates of each voxel within a specified structure in the volume as a three dimensional vector. These vectors form the sets X and Y of voxel coordinates from the two volumes to be registered. From these sets, we compute the covariance matrices, C_x and C_y , and means, $\vec{m}_x$ and $\vec{m}_y$, of the volumes. In this case, the mean of a set is simply the centroid of its corresponding volume. From the covariance matrices, we then compute the principal axes (eigenvectors) of the structure in the volume. The principal axes of X and Y are used as the rows for the transform matrices P_x and P_y respectively. These are ordered so that angles between corresponding axes are minimized.

Instead of using the actual Hotelling transform, our implementation creates a transform which aligns the principal axes of volume 1 to the nearest respective principal axes of volume 2. In this we are making the assumption that the volumes to be registered differ by a rotational angle of less than 45 degrees in any direction. We can make this assumption because the patient oriented coordinate system in the DICOM image class is essentially the same for different images but may differ by a translation and small rotation. The transformation mapping the vectors in X to X' so that X' is registered to Y is

$$\vec{x}'_k = \vec{m}_y + P_y^T P_x (\vec{x}_k - \vec{m}_x)$$

This transform first moves the centroid of X to the patient origin. It then rotates the vectors in X using P_x so that their principal axes lie along the patient coordinate system's axes. These vectors are again rotated by P_y^T so that their principal axes align with the principal axes of the vectors in Y . The centroid of Y is then added so that the patient coordinate systems of X' and Y are approximately identical as is shown in Figure 3.2.

The principal axes transform is very efficient, having a computational complexity of $O(n)$ where n is the number of voxels in the volumes to be registered. It offers an approximate registration solution which can be followed by a slower, more accurate method. Because it does not utilize pixel intensity information in

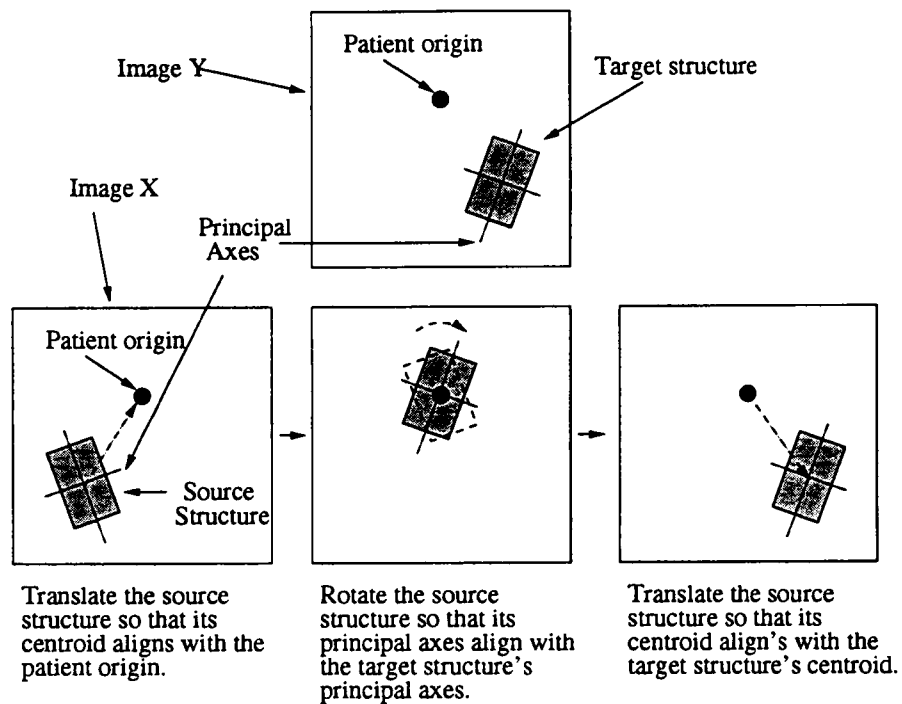


Figure 3.2: Registration using centroids and principal axes.

the calculation of the registration transform it can be used for images from multiple modalities, eg. CT, MR, and PET. However, whether the the images being registered are from the same modality or not, they must be properly segmented to obtain nearly identical structural information as it is the geometric structure of an image that is registered. In Chapter 4 we will see examples of the registration of MR anatomical scans that have been both properly and improperly segmented as well as the registration of idealized phantom images.

Chapter 4

Results

In Chapters 2 and 3, we discussed the interpolation and registration of medical image volumes. This process involves error which can lead to inaccurate results. Three possible sources of error are 1) incomplete data due to gaps between images and discrete image registration, 2) rounding and averaging during interpolation, and 3) noise or inaccuracies in acquisition. In this chapter we describe methods of assessing the accuracy of registration techniques, followed by an error analysis obtained by applying one of these methods to our registration algorithm. We then conclude with examples of phantom slices and anatomical MRI volume slices that have been extracted from an interpolated and registered set of volumes.

4.1 Accuracy Verification Methods

In the literature, several methods of assessing the accuracy of a registration match have been described. Because different forms of registration have unique characteristics, not all verification methods are applicable to a given technique. For example, the estimated transformation angles of a global rigid transform can be compared to the true transformation angles if they are known, however in a local elastic method, the curved nature of the transform precludes this type of comparison. We will now proceed to discuss methodology and applicability of accuracy verification techniques, namely 1) comparison of artificial versus automatic transform parameters, 2) determination of the amount of overlap between structures in images, 3) qualitative verification, and 4) comparison of the angles between principal axes.

The first verification method we discuss involves comparing the parameters generated by the registration algorithm to parameters determined by some artificial means. This method was employed by van den Elsen et al. [18] to compare matching using the L_{vv} operator of cranial images and is effective for registration methods that provide a global rigid transform. In this case, the automatically generated transform is uniquely decomposed into its constituent parts, i.e. translations and rotations, and these are compared to artificial transform parameters obtained by physically aligning the images. Though this method works well, ar-

tificial parameters obtained in this manner may contain error because they are based on human judgment.

In our case, we compare identical test volumes that have been artificially translated and rotated before registration. Because the inverse of this transform is known in advance, there is no need to physically align the volumes at a later time, so the error in human judgment is avoided. Unfortunately, comparing transformation parameters is not a viable method to analyze our registration process because the calculated transformation matrix is not unique. When the automatically generated matrix is decomposed into translations and rotations about the x, y, and z axes, the parameters determined may vary greatly from the artificial parameters and yet effect the same overall transform. Though we do not utilize this verification procedure, it is similar to the method we employed to determine the accuracy of our registration algorithm.

When unique parameters cannot be determined, or when a transformation is local so that there is more than one set of parameters, a second verification method involving computing the overlap of structures in an image may be useful. Pellizari et al. [17] registers two phantom cranial volumes obtained from a plastic skull with geometric markers placed in it. The markers in the target volume are compared with the markers in the registered volume to ascertain error. Gee et al. [11] also uses a comparison of this sort. A slice from an elastically registered brain atlas is compared to a slice from the target image. Actual structures within

the cranium are examined, and the error is computed by

$$error = \frac{(volume\ of\ intersection\ of\ structures)}{(volume\ of\ union\ of\ structures)}.$$

Though this method is very useful when transform parameters cannot be determined, low resolution may decrease its accuracy. For this reason, comparing the overlap of structures was not used in our analysis.

The third verification method is suggested by Woods [20] and is a qualitative evaluation of the images by a human observer. One simply obtains images resulting from the registration process and compares them visually. Obviously this method involves the likelihood of judgment error, but it is a good way to check the aesthetic properties of a registration algorithm. Later in this chapter we use this method to compare individual MRI volume slices generated by our interpolation and registration implementation. These slices have subtle differences which make a non-qualitative comparison of the images difficult and error prone.

The final method we discuss for verifying the accuracy of a registration procedure is to compare the differences in the principal axes and centroids of the automatically registered volume to the axes and centroids of the target volume. This is the method we have chosen to implement using mathematically generated phantom images. Our procedure is to first define a geometric object, r_s , for which the principal axes and centroid are known in advance. The source volume, v_s , is

interpolated from slices extracted from r_s . A second object, r_t , of identical size and shape is defined with the desired transform applied to it. We then interpolate the target volume, v_t from slices extracted from r_t . Because r_s and r_t are mathematically defined, it is simple to determine their exact centroids and principal axes which we label as sets P_s and P_t respectively. Following this, v_s is automatically registered to v_t using the principal axes method, and the calculated transform is then decomposed into its constituent translations and rotations. These translations and rotations are applied to P_s , the original principal axes and centroid of v_s , to get P'_s . Finally, P'_s is compared to P_t to determine translation and rotation error.

By using the centroid and principal axes to determine error, the results of the transform and not the transform parameters are being compared, therefore it does not matter if the transform is not unique. Also, this method utilizes the image attributes on which our registration algorithm is based, and so it is especially applicable to our situation. We now review the application of this method to a series of phantom image sets.

4.2 Error Analysis

In this section we discuss the results obtained from registering a source volume containing a mathematically defined geometric object to a second target volume

containing an identical object. The objects are $30 \times 40 \times 60$ mm rectangular prisms. The principal axes of the first rectangle lie along the coordinate axes in which it is defined, and its centroid lies at the origin. The second rectangle has been rotated and translated through application of a global rigid transform. The definitions for these rectangles are used to create sets of two dimensional DICOM images which are in turn used to interpolate the source and target volumes. The principal axes and centroids of these volumes are then compared using the fourth scheme described in Section 4.1. The aspects from these comparisons which we will concern ourselves with are 1) registration error with respect to volume resolution given a single set of transformation parameters, 2) registration error resulting from differing rotation angles about one axis, and 3) average registration error with respect to volume resolution given several different sets of transformation parameters.

We first look at the result of comparing the amount of registration error to the resolution of the registered volumes given a fixed rotation. In Figure 4.1 the y axis depicts the rotational error in degrees of one of the volume's principal axes with respect to the voxel size which is depicted along the x axis. From this graph we see that the rotational error starts very small at a voxel size of less than 1.0 mm and gets progressively greater with larger voxel sizes. Resolution is inversely related to voxel size, so as might be expected, error increases as resolution decreases. This fact will be seen more clearly in a later graph and it results from the fact that

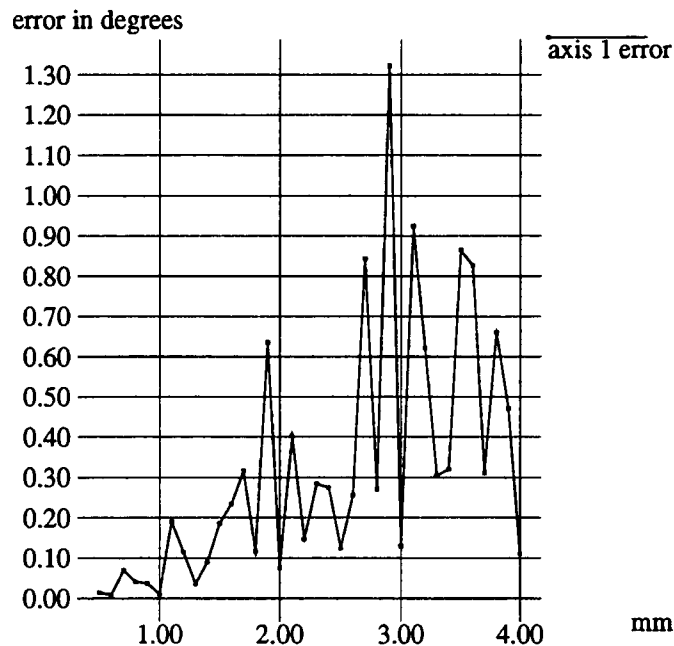


Figure 4.1: Comparison of rotation versus voxel size for a phantom image with an initial rotation of 10 degrees about the y and z axes and an initial translation of 10 mm along the x, y, and z axes.

lower resolution involves reduced image accuracy. When we compare translational error to accuracy, we see a similar response as is shown in Figure 4.2.

In both of these figures one can see an oscillation between large and small amounts of error as the resolution decreases. This can be accounted for by the fact that the images are represented in a discrete form. Given a very small change in resolution, the representation of the image remains the same. However, the principal axes and centroid will change somewhat to accommodate the new resolution. This causes a small increase in the amount of error. At some point, if the resolution continues to decrease, the discrete representation of the image will

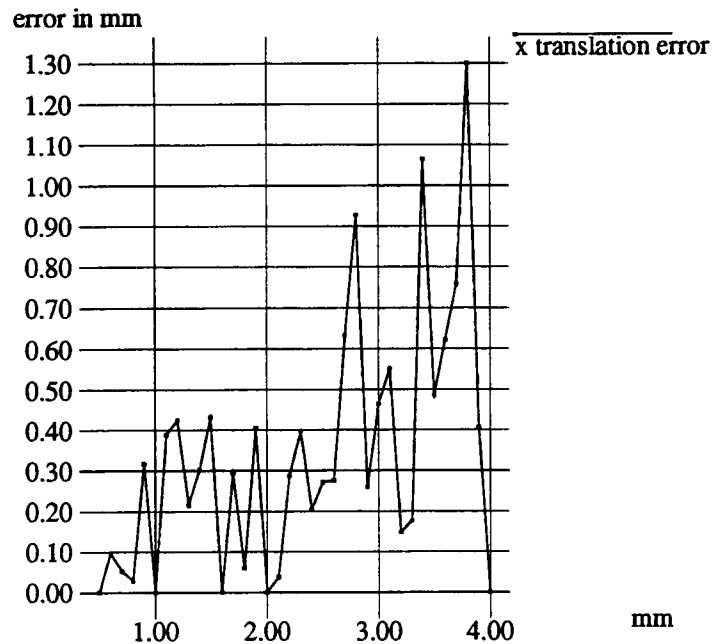


Figure 4.2: Comparison of translation versus voxel size for a phantom image with an initial rotation of 10 degrees about the y and z axes and an initial translation of 10 mm along the x, y, and z axes.

adjust due to rounding. At this point, the error is locally maximum. Then, as the resolution continues decreasing, the error will decrease until the principal axes and centroid again align with the new discrete representation of the image.

The second comparison made using the results obtained from the phantom images shows the rotational error in thousandths of degrees with respect to a series of rotations about the x axis. This is represented by Figure 4.3 with rotations about the x axis varying from 0 to 20 degrees. The error alternates between high and low values in the same manner as the oscillation found when changing the volumes' voxel sizes. This is similarly a result of the images being represented

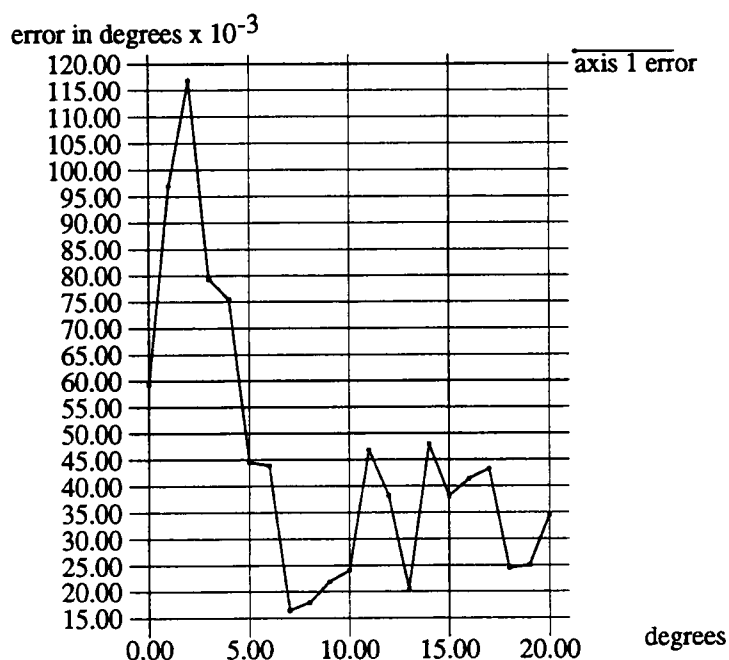


Figure 4.3: Comparison of rotation versus voxel size for a phantom image with an initial rotation of 0 and 10 degrees about the y and z axes respectively, and an initial translation of 10 mm along the x, y, and z axes.

in a discrete form. The spike observed in the graph at a two degree rotation is notable and can possibly be accounted for by realizing that the rectangles are originally aligned with the x, z plane of the volume's coordinate system. Since a line or plane represented in discrete form is greatly distorted when rotated at small angles relative to its coordinate axes, this behavior can be expected.

The third result which we consider is a comparison between the average rotational error in degrees versus volume voxel sizes when the rotation parameters for the initial transform are allowed to take several values. The rotation about the x axis varies in one degree increments from 0 to 20 degrees and the rotations about

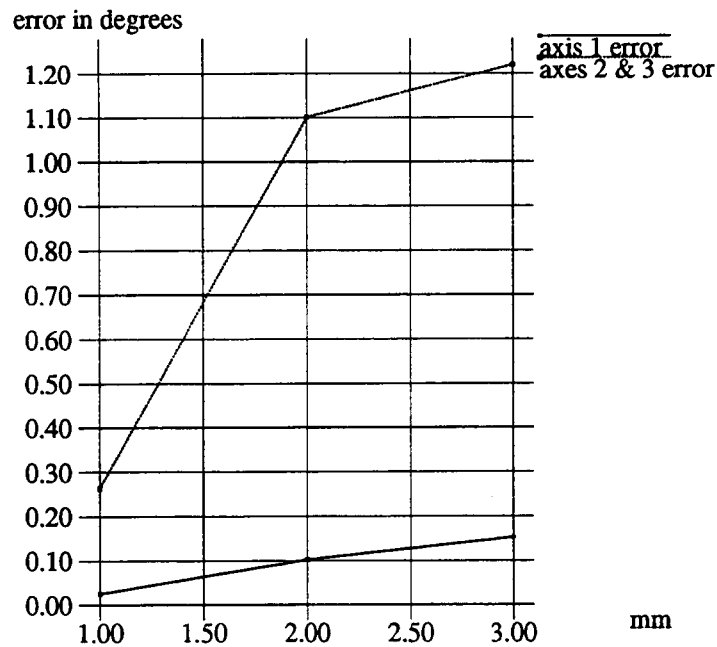


Figure 4.4: Comparison of average rotation versus voxel size given initial rotations about the y and z axes of both 0 and 10 degrees.

the y and z axes vary between either 0 or 10 degrees. Figure 4.4 shows the results of these comparisons. As was mentioned previously, it is clear that the average error increases as the resolution decreases. However, there is a great difference between the amount of error between the first axis of the volumes and the second and third axes. This can be accounted for by noting that the rotation around the x axis varies from 0 to 20 degrees which apparently has an averaging effect on the rotation error for the first axis of the registered volumes. The rotations of only 0 or 10 degrees about the y and z axes apparently cause larger errors in the second and third axes of the registered volumes. It is expected that if the rotations about the y and z axes were allowed to vary between 0 and 20 degrees, then the resulting

error would be similar to that of the first axis.

Overall, the registration results are very good. The largest registered rotational error is 1.32 degrees which is quite small. Also, the translational error in millimeters never exceeds half the value of the voxel size. However, these results do not show the ability of the principal axes transform to register anatomical MRI, CT, or PET image volumes. In the next section, we display actual medical volume slices that have been registered using our implementation.

4.3 Registration of Anatomical Data

Perhaps the final test to apply to any registration algorithm is a qualitative assessment. Ultimately, it is expected that a registration algorithm will be used so that images can be compared visually, therefore it is definitely the case that registered images should appear to match each other. We were able to obtain MR anatomical scans; in this section we compare corresponding slices obtained from the source, target, and registered source volumes which were interpolated from those scans. We examine four volume sets; the first of which is an example of the phantom volumes described in Section 4.2, the second being a cranial and chest scan series, the third is a cranial series only, and the fourth is a cranial set showing the effects of improper segmentation.

The first set of images display the accuracy of our registration procedure when

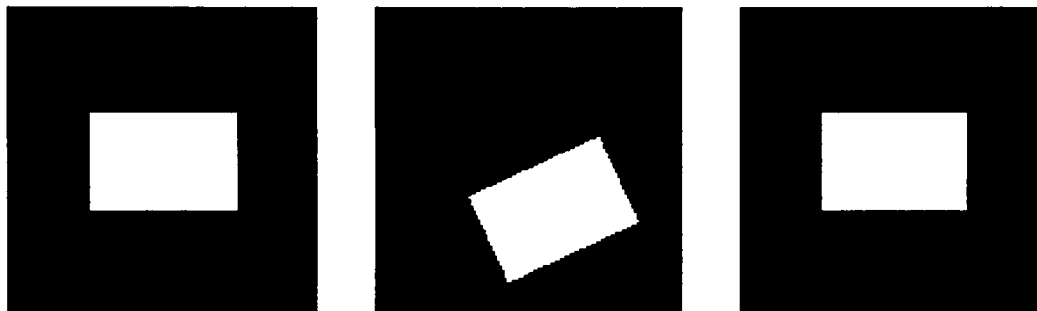


Figure 4.5: Slices obtained from source, target, and registered source phantom volumes.

the volumes to be registered are virtually identical. We have registered two rectangular prisms and give sample slices of the source, target, and registered source volumes in Figure 4.5 in that order. These slices all correspond to identical locations in the volumes with respect to the patient origin. As can be seen, there is no noticeable error.

The second image set represents slices of the cranium and left shoulder. These images were obtained by registering a source volume which was interpolated from sagittal scans through the patient to a target volume derived from a set of oblique scans. Corresponding slices extracted from the source, target, and registered source volumes can be seen in Figure 4.6 respectively. It is plain that though the source image represents the slice with the same position as the target image, they are definitely not visually identical. From this we see that the volumes' patient origins are different. However, the target slice compared with the registered source slice shows the effectiveness of the Principal Axes Method as these slices are approximately the same. Both the source slice and the registered source slice

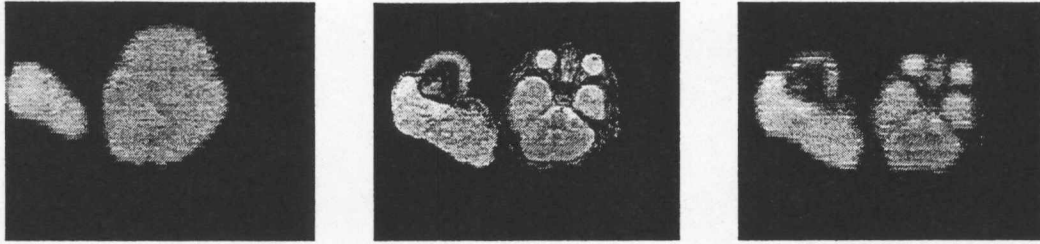


Figure 4.6: MRI slices of cranium and shoulder obtained from source, target, and registered source volumes.

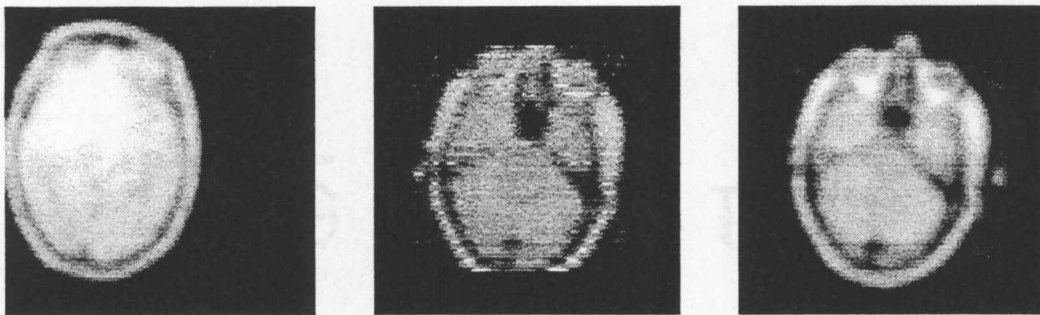


Figure 4.7: MRI Cranial slices obtained from source, target, and registered source volumes.

display the banding effect mentioned in Chapter 2.

The third image set represents cranial slices only. These images were obtained in a similar fashion as the second set except that oblique scans were used for the source volume and sagittal scans for the target volume. Figure 4.7 shows the corresponding slices obtained from the source, target, and registered source volumes. Again, we observe the difference between the source and target slices, and the similarity between the target and registered source slices.

The final image set in Figure 4.8 displays the error that can occur when volumes are not segmented properly. It can be seen that the source and target images are nearly identical before registration, but that the registered source image is rotated

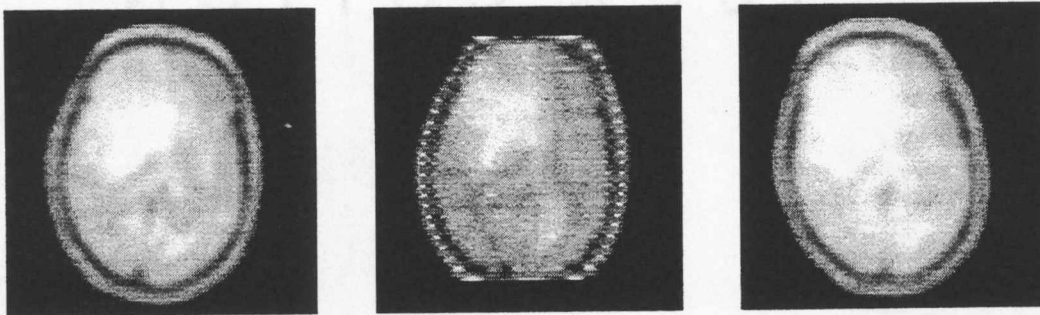


Figure 4.8: MRI lices displaying misregistration due to improper segmentation.

by approximately 10 degrees about the z axis. This is due to thresholding the volumes at a level which allowed noise in the original scans to be included in the registration process. From this we see the importance of correct segmentation of volumes before registration takes place.

The results of the implemented registration process have been seen to be successful. An SGI Indigo 2 workstation was used to perform the interpolations and registrations on images conforming to the DICOM image class. The registration procedure itself took under 10 seconds to register medical scans with volume dimensions of $143 \times 91 \times 143$ and $134 \times 134 \times 108$ respectively where each voxel was 2.1mm cubed. However, the interpolation of the source and target volumes in addition to the reinterpolation of the newly registered volume required approximately 33 minutes due to large gaps between original MR scans..

Chapter 5

Conclusion

In this thesis we have studied medical image interpolation and registration. We have considered several forms of interpolation including linear, cubic, shape-based, and kriging. The method chosen to be implemented was a form of linear interpolation which we have designated as modified linear interpolation. This form combines speed as well as the ability to consider several voxels when computing an unknown value. Its primary computation utilizes a formula similar to the basic kriging equation but does not include a specialized method for determining voxel weights based on an image's structural attributes. Instead, these weights are computed using the reciprocal of the distance between voxel centers. The method is very effective when computing volumes based on complete data sets, however, in many cases gaps exist between medical scans which ultimately create a banding effect in the computed volume. This effect can be reduced using blurring

or lowpass filtering methods. The difficulties experienced with this interpolation procedure are common to any method that is applied to incomplete data.

Several registration algorithms have been presented among which are polynomial approximation, cross-correlation of features, elastic matching, and principal axes registration. We have discussed in detail and have chosen to implement the principal axes method. This method is a modified form of the Hotelling algorithm. It registers images by 1) computing the centroids and principal axes of two volumes, 2) translating the centroid of the source volume to the patient origin, 3) aligning the source volume's principal axes to the target volume's principal axes, and 4) translating the centroid of the source volume to the centroid of the target volume. This type of registration utilizes a global and rigid transform and so provides a coarse registration of two volumes. It is relatively fast compared to other registration methods and provides the approximate solution that is desired.

Because the principal axes method is fast and provides a coarse registration, it is an excellent way to begin a more detailed registration process. It can be used to approximately register two volumes so that a further registration step, such as polynomial approximation or elastic matching, can fine tune the results. The initial principal axes registration is very useful before attempting a curved or local registration method because it greatly reduces the amount of time spent in these computationally intensive procedures.

The results of the chosen interpolation and registration processes are seen to

be highly accurate under ideal circumstances. With lower resolution, registration error increases but is still within an acceptable range. The ability of the principal axes method to register actual anatomical images is also displayed. Though these results often depend highly on the image segmentation process, every registration procedure requires accurate extraction of image properties. Compared to others this method excels in its simplicity and efficiency

Bibliography

Bibliography

- [1] ACR-NEMA digital imaging and communications in medicine (DICOM v3.0) faq. <http://www.xray.hmc.psu.edu/dicom/faq.html>.
- [2] B. A. Ardekani, M. Braun, I. Kanno B. F. Hutton, and H. Iida. A fully automatic multimodality image registration algorithm. *Journal of Computer Assisted Tomography*, 19(4):615–623, July/August 1995.
- [3] R. Bajcsy and S. Kovacic. Multiresolution elastic matching. *Computer Vision, Graphics and Image Processing*, 46:1–21, 1989.
- [4] R. Bajcsy, R. Lieberman, and M. Reivich. A computerized system for the elastic matching of deformed radiographic images to idealized atlas images. *Journal of Computer Assisted Tomography*, 7(4):618–625, August 1983.
- [5] D. C. Barber. Registration of low resolution medical images. *Phys. Med. Biol.*, 37(7):1485–1498, 1992.

- [6] W. H. Beyer, editor. *CRC Standard Mathematical Tables*. CRC Press, Inc., 27th edition, 1984.
- [7] G. Borgefors. Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):849–865, November 1988.
- [8] C. Broit. *Optimal Registration of Deformed Images*. PhD thesis, University of Pennsylvania, 1981.
- [9] R. L. Burden and J. D. Faires, editors. *Numerical Analysis*. PWS Publishing Company, fifth edition, 1993.
- [10] A. P. Dhawan, L. K. Arata, A. V. Levy, and J. Mantil. Iterative principal axes registration method for analysis of MR-PET brain images. *IEEE Transactions on Biomedical Engineering*, 42(11):1079–1087, November 1993.
- [11] J. C. Gee, M. Reivich, and R. Bajcsy. Elastically deforming 3D atlas to match anatomical brain images. *Journal of Computer Assisted Tomography*, 17(2):225–236, March/April 1993.
- [12] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley Publishing Company, Inc., 1992.

- [13] S. C. Horii, F. W. Prior, D. W. Bidgood, Jr., C. Parisot, and G. Claeys. DICOM: An introduction to the standard. http://www.xray.hmc.psu.edu/dicom/dicom_intro/DICOMIntro.html, 1996.
- [14] B. Kolman. *Elementary Linear Algebra*. Macmillan Publishing Company, fourth edition, 1986.
- [15] H. Maitre and Y. Wu. Improving dynamic programming to solve image registration. *Pattern Recognition*, 20(4):443–462, 1987.
- [16] R. W. Parrott, M. R. Stytz, P. Amburn, and D. Robinson. Towards statistically optimal interpolation for 3-D medical imaging. *IEEE Engineering in Medicine and Biology*, pages 49–58, September 1993.
- [17] C. A. Pelizzari, G. T. Y. Chen, D. R. Spelbring, R. R. Weichselbaum, and C. T. Chen. Accurate three-dimensional registration of CT, PET, and/or MR images of the brain. *Journal of Computer Assisted Tomography*, 13(1):20–26, January/February 1989.
- [18] P. A. van den El, J. B. A. Maintz, E. J. D. Pol, and M. A. Viergever. Automatic registration of CT and MR brain images using correlation of geometrical features. *IEEE Transactions on Medical Imaging*, 14(2):384–396, June 1995.

- [19] P. A. van den Elsen, E. J. D. Pol, and M. A. Viergever. Medical image matching – a review with classification. *IEEE Engineering in Medicine and Biology*, 12:26–39, March 1993.
- [20] R. P. Woods, J. C. Mazziotta, and S. R. Cherry. MRI-PET registration with automated algorithm. *Journal of Computer Assisted Tomography*, 17(4):536–546, July/August 1993.

Vita

Gregory Wrobel was born in South Bend, Indiana on August 23, 1966. He attended schools in the South Bend Community school system and graduated from Clay Senior High School in May, 1984. Following this, he attended Rose-Hulman Institute of Technology in Terre Haute, Indiana during 1984 and 1985. He then entered the University of Texas at Austin in January of 1987 and graduated in August, 1990 with a Bachelor of Science degree in Mathematics and a teaching certificate in secondary education. After teaching four years, he entered the Computer Science Department at the University of Tennessee, Knoxville as a teaching assistant and student in August, 1994. He received his Master of Science degree in August of 1996 after working as a graduate research assistant for the University of Tennessee Medical Center.