Registration of Optical Images to 3D Medical Images

Matthew John Clarkson

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
of the
University of London.

June 2000

Computational Imaging Science Group
Division of Radiological Sciences and Medical Engineering
Guy's, King's and St. Thomas' School of Medicine
King's College London

Abstract

The work described in this thesis deals with the registration of single and multiple 2-dimensional (2D) optical images to a single 3-dimensional (3D) medical image such as a magnetic resonance or computed tomography scan. The approach is to develop an intensity based method using an information theoretic framework, as opposed to the more typical feature or surface based methods. Relevant camera calibration and pose estimation literature is reviewed, along with medical 2D-3D image registration. An initial algorithm is developed, which performs registration by iteratively maximising the mutual information of a rendered image and a single optical image. The framework is extended to incorporate information from multiple optical and rendered images which significantly improves registration performance. A tracking algorithm is proposed, which augments this framework with texture mapping as a means of achieving alignment over a sequence of optical images. These methods are tested using images of skull phantoms and volunteers.

A new measure based on the concept of photo-consistency, used in the surface reconstruction literature, is proposed as a measure of image alignment. The relevant theory is developed. This new method is tested using a variety of different photo-consistency based similarity measures, optical images, different numbers of images, images with varying amounts of added noise, different resolutions and different camera positions relative to the object of interest. In almost all cases, similarity measures based on this new framework perform accurately, precisely and robustly. Potential applications will be in radiotherapy patient positioning, image guided craniofacial, skull base and neurosurgery, computer vision and robotics, where the accurate alignment between a 3D image or model and multiple 2D optical images is required.

Acknowledgements

Many thanks to my supervisors Prof. David Hawkes and Dr. Derek Hill for their enthusiasm, inspiration, encouragement and direction. Thanks also to Dr. Daniel Rückert for day to day guidance and support.

Thanks go to everyone in the Computational Imaging Science Group, and in addition, Colin Studholme, John Little, Paul Summers and Calvin Maurer Jr., all of whom have been good friends and fun to work with. A special thanks to Jane Blackall for innumerable cups of tea, coffee and biscuits.

I would also like to thank the Engineering and Physical Sciences Research Council for funding my studentship.

My sincerest, heartfelt thanks are due to Sunny Uberoi, David Bull, Richard Compton-Burnett, Tim Whittome, Andrew Green and Michael Collier. You have supported, encouraged and prayed for me. Many thanks and God bless.

Thanks also to my flatmates, Izzie Carrington, Anne Treiber, Kate Houston, Georgina Baron, Lucy Wilson, Abbey Martin, Gulia Allabergenova and Philippe Batchelor who have provided a place for me to relax and enjoy their company. Thanks also to friends who have made life fun along the way: Kirsty Parsons, Claire Rawson-Mackenzie, Claudia Rodriguez Carranza, Anna Last, Carol Feelie, Wayne Sorensen, Ricky Cauldwell, Paul Kirby, Simon Marsden, Nigel Ward, Dave Betts, Tom Hollins, Gary Skinner, Adam Davies and Tim Stone. A big thanks to the many other people, not listed here, who have been good friends.

Thanks also to mum, dad and my family, who have loved, cared for and encouraged me, without which this thesis would not be finished.

Above all, thanks to God, my Father, my strength and my redeemer.

Contents

List of Figures

List of Tables

List of Abbreviations

2D	2-dimensional
3D	3-dimensional
CAD	computer assisted design
CCD	charge coupled device
CT	computed tomography
DOF	degrees of freedom
ENT	ear, nose and throat
IRED	infra-red light emitting diode
LED	light emitting diode
MI	mutual information
MR(I)	magnetic resonance (imaging)
NCC	normalised cross correlation
NMI	normalised mutual information
PET	positron emission tomography
SPECT	single photon emission computed tomography
SSD	sum of squared differences
SVD	singular value decomposition
VTK	visualization toolkit [Schroeder *et al.*, 1997]
$\mathrm{PC}_{\mathrm{squared}}$	the sum of squared differences of photo-consistency, with a light fixed relative to the object
$\mathrm{PC}_{\mathrm{good}}$	the sum of good, photo-consistent points, with a light fixed relative to the object
$\mathrm{PC}_{\mathrm{inverse}}$	the sum of inverse squared differences of photo-consistency, with a light fixed relative to the object
$\mathrm{PC}'_{\mathrm{squared}}$	as $\mathrm{PC}_{\mathrm{squared}}$ but with a light fixed to each camera
$\mathrm{PC}'_{\mathrm{inverse}}$	as $\mathrm{PC}_{\mathrm{inverse}}$ but with a light fixed to each camera
$\mathrm{PC}_{\mathrm{mutual}}$	photo-consistency, measured with mutual information

Part I

Introduction And Background

Chapter 1

Introduction

There are many different types of clinical, 3-dimensional (3D) radiological imaging modalities available today. Images such as magnetic resonance (MR) and computed tomography (CT) show anatomy, and images such as positron emission tomography (PET) and functional MR imaging (fMRI) show metabolic function. The corresponding imaging devices measure a physical property such as attenuation of X-rays, magnetic properties or emission of photons. Three dimensional medical images represent a regularly sampled set of measurements of some physical property that is not visible to the human eye and these measurements may be internal to an object and hence obscured to the eye. Given that each eye only captures a 2D image, and that the brain is left to reconstruct some 3D representation of the surrounding world, how then can the information contained within 3D radiological images be best utilised by the brain? How can clinicians relate 3D medical image information to the images captured by their eyes?

Devices such as video cameras or endoscopes capture 2-dimensional (2D) optical images, which measure the number of photons of light, incident on a sensor array. Thus optical images look familiar to the human observer, as the retina of the eye also measures the incident light and conveys an image to the brain for subsequent processing.

In order to relate information in a 3D image and a 2D image, the two images must be registered or aligned. To register two images is to compute the mapping between spatial locations in one image and spatial locations in another. Once registered, one can say, "this feature X in the 3D image, must correspond to feature Y in the 2D image". If 2D optical images look familiar to the human, as they represent what the world 'looks like', then the registration of 3D medical and 2D optical images provides a link between the information in a 3D image, and an observation of the physical world around us. This

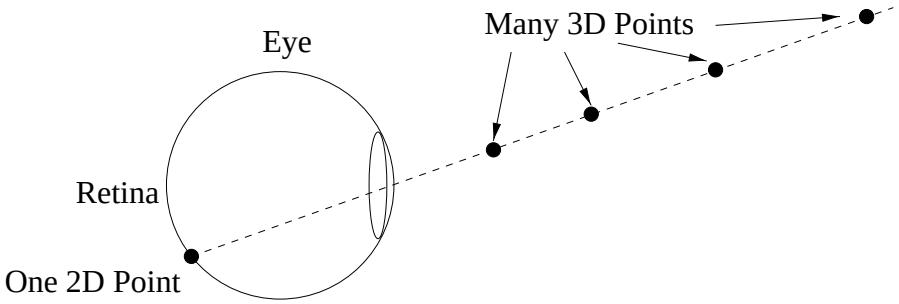


Figure 1.1: In the human eye, many 3D points project to a single 2D point. We speak of a 'line of sight' to describe this.

enables the clinician to examine a 3D image, and identify pathology and to pinpoint its position within a patient. It enables surgeons to take optical images of a patient before them and to say, "from this external viewpoint, and with the knowledge of an accurate registration, I can take my chosen route to the surgical target, confident that I will avoid other critical structures", even before an initial incision has been made. It is accurate registration that provides the link between the 3D medical images and 2D optical images, and it is this registration that relates the 3D images to the more familiar world around us, that otherwise would have to be performed mentally.

This thesis describes methods for the registration of a set of 2D optical images to 3D medical images such as MR or CT scans.

1.1 2D-3D Registration

The term 'registration' can be defined as follows. Registration is the determination of a mapping between coordinates in one space and coordinates in another, such that points which correspond to the same physical location are mapped to each other.

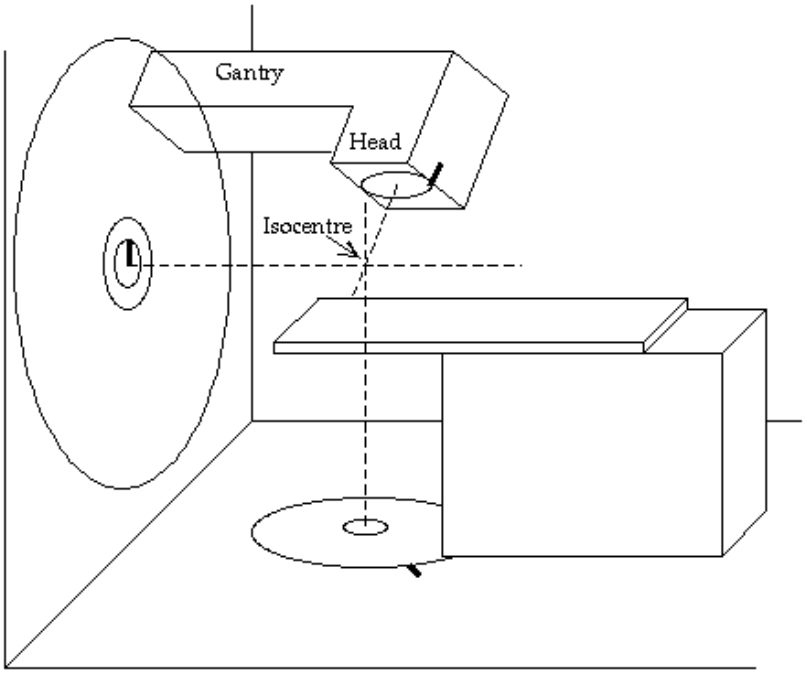
Consider figure 1.1. If the 3D world is imaged, using either the eye or a camera, then the 2D image captures a projection of the 3D world. This means that many 3D points are projected onto the same 2D point. Thus, in this definition of registration, the mapping from 3D to 2D is many-to-one and the mapping from 2D to 3D is one-to-many. The necessary coordinates and equations are described in chapter 2.

1.2 Motivation

The work described in this thesis is primarily motivated by two clinical areas described below. However the problem of 2D-3D registration itself is a general one, applicable to many areas where it is necessary to register or align a 3D image or model with one or more 2D optical images. In this thesis, clinical images are not used. Applying the proposed registration algorithms to clinical applications will form part of the future work. Instead, the thesis focuses primarily on the theoretical and algorithmic developments. Therefore, the descriptions below aim to illustrate and promote several potential applications where such a registration algorithm will find use, rather than be an exact specification of a particular clinical application or scenario that this thesis will address and solve.

1.2.1 Patient Positioning For Radiotherapy

The current clinical procedure at Guy's and St. Thomas' Hospital for positioning patients about to undergo radiotherapy treatment is a rather long process. Consider the cases that have had some 3D scan e.g. CT. With this scan, the proposed radiation treatment plan can be formulated, where treatment may last several days/weeks. For each treatment the patients must be accurately re-positioned on the radiotherapy treatment couch. This is usually achieved using a plastic shell which is custom made to fit each patient, and can be rigidly attached to the treatment couch. The shell is made by placing the patients in a similar position to that in which they will be treated and wrapping them in strips of bandage covered in plaster of Paris. The patients must remain stationary while the plaster of Paris sets. Depending on the patient's condition, this may be uncomfortable or intolerable. A positive head mould is made from the plaster of Paris shell, around which a plastic shell is created by vacuum forming. The patient's shell must then be mounted on a head board in a specific position using plastic struts. The alignment is done by checking that the tragal notch or some other feature is equally high on both sides of the head, and the shell is fixed to these struts. In the treatment planning room, the patients must be aligned with the beam of the linear accelerator. This is achieved using laser guidelines, and raising and lowering the couch until the target of interest lies in the correct place relative to the iso-centre of the radiation beam (see figure 1.2).



(a)

Figure 1.2: (a) Diagram of a radiotherapy linear accelerator, by Dominic Withers. (used with permission).

If the shell is made several weeks prior to treatment, the patient may change weight between shell construction and actual treatment. The skin is inherently deformable, and moves easily relative to underlying tissue. Therefore the shells are unlikely to fit very accurately. In addition, there will be a significant difference in the possible positions that a patient can lie in within the shell. It may also take several minutes for trained personnel to position the patient in the treatment room in a sufficiently similar position to their previous treatment.

In short, shell construction is time consuming and potentially inaccurate. Positioning the patient accurately and reproducibly is difficult. If, however, a patient has already had a CT scan, and cameras can be mounted in the treatment room, then by registering the 3D image to the 2D optical images it will be possible to calculate whether the patient is in the correct position relative to the linear accelerator. This will require that cameras be rigidly attached to the accelerator, and be calibrated such that given a coordinate within the field of view in a video image, then the location of that coordinate relative to the radiation beam is known. If such a registration algorithm is used, and a computer used to verify the patient position, it may be possible to completely avoid mould and shell

making as these are merely fixation devices. It may still be necessary to use some kind of immobilization, e.g. padded head rests, and use the registration algorithm to detect when a patient has moved out of some tolerance region and switch the linear accelerator off, before causing unnecessary damage to the patient. If the registration were performed in real time, with sufficient accuracy, then potentially, the treatment could be performed without immobilization.

1.2.2 Image Guided Surgery

Consider a patient who has had some pre-operative 3D medical image taken of an area of interest for some surgical procedure. It is clearly vital that a surgeon is able to relate pre-operative information to the current surgical scene before them. This is difficult to do. The success of a procedure is heavily dependent on the surgeon's training and ability to perform mentally the necessary 'registration' from the physical space of the operating room to the pre-operative images. Furthermore, surgeons are unavoidably limited by the fact that some objects are not transparent. There will be structures, within the surgical field that are of critical interest and yet are obscured, e.g. nearby blood vessels. Currently a surgeon must use his/her judgement, experience and prediction to reach a target whilst avoiding other critical structures. The proposed registration algorithms were developed with the motivation of making these tasks simpler for a surgeon.

Image guided surgery (IGS) uses devices such as an optical tracking device e.g. Optotrak (Northern Digital) to track the position of surgical localisers within the surgical field. By registering the coordinate system of the optical tracker to the pre-operative images it is then possible to relate the physical position of a tracked localiser to the corresponding position within the pre-operative images [Maciunas, 1993]. This enables the surgeon to be guided by the pre-operative images, to identify physical structures, to measure the distance to unseen structures and so on. These methods usually require that the surgeon look away at a computer display. In addition the localised position is usually visualised at the intersection of three orthogonal planes through the 3D image.

However, with a 2D-3D registration algorithm it is possible to display the pre-operative information in a more intuitive fashion. An optical image of the current operative scene would capture the same view as the surgeon sees. By registering a 3D image to the optical image, information from the 3D image can be overlaid on the optical image. For instance a graphical model of a tumour present inside the patients head can be drawn on top of the

(a)

Figure 1.3: This picture shows the MAGI system [Edwards *et al.*, 1999b]. The microscope housing holds image injectors that augment the optical image with virtual information from the pre-operative data.

video image. This would provide an 'augmented reality' where information concerning the real scene was augmented with virtual representations of the pre-operative data. It would give the surgeon the ability to visualise the position of a tumour, before making any incisions. An increased awareness of the actual size and location of the tumour may enable the surgeon to reduce the size of the planned craniotomy to the minimum required to successfully complete the planned procedure. In addition, once the optical images and 3D image are registered, renderings of critical structures can be overlaid in the correct position to guide the surgeon around them. The augmented overlays can be achieved on a workstation monitor, or with hardware such as image injectors, to overlay an image within each eye piece of a stereo operating microscope, see figure 1.3 and [Edwards *et al.*, 1999d]. The proposed registration techniques could be applied to ENT surgery, neurosurgery or craniofacial surgery.

Figure 1.4: An example of pre-operative data overlaid on an intra-operative video image. The patient had a petrous apex cyst removed. A rendering of the zygomatic arch and carotid artery (in blue) were overlaid onto the view seen through the operating microscope.

The MAGI (microscope assisted guided interventions) system shown in figure 1.3 provides image guidance using overlays such as that shown in figure 1.4. This patient had bilateral petrous apex cysts. The usual approach is through the cochlea/labrynth, which was inappropriate here as the patient would loose all hearing. Instead, this cyst was approached though the zygomatic arch. The figure shows a video image taken from a camera mounted within the operating microscope. During the operation, renderings of information from the pre-operative MR scan was overlaid onto the video image to provide guidance. In this figure, a rendering of the zygomatic arch and carotid artery were overlaid onto the video image. The registration was achieved using bone implanted markers. Would it be possible to perform this registration without bone implanted markers?

To summarise, 2D-3D image registration can be used to provide image overlays and image guidance to enable a surgeon to make informed decisions and guide them towards targets whilst avoiding other structures. Ultimately this could promote quicker, safer and less invasive surgery.

1.2.3 Computer Vision And Robotics

2D-3D registration can also be used in computer vision or robotics applications. Instead of a 3D medical image, the 3D information could be from a CAD (computer assisted design) model, a laser or patterned light range finder, indeed any 3D model that accurately reflects the shape of an object. The 2D optical images could be from a variety of cameras. If the 3D model described a room or environment, then registering this model to images from cameras mounted in a mobile robot may allow for autonomous robot navigation. If the 3D model described some part of a manufactured object, then registering this model to images from cameras mounted on a production line robot may allow for robot assisted manufacturing. Another application may take video images of people and match them to a database of models for security identification processes. A further potential application of 2D-3D registration may be in computer assisted learning. For instance, video cameras could take images of an aircraft engine and register this to a known model. A computer could then overlay instructions or guidelines on the image to assist an engineer to maintain the engine in some way. This may be done using wearable video cameras, and the overlay performed using a head up display or some augmented reality hardware.

To summarise, there are many applications of a 2D-3D, optical image to 3D model registration algorithm. Chapter 3 reviews the current state of the art with emphasis on clinical applications. The registration framework developed in chapter 7 could potentially be applied to many computer vision or robot based applications.

1.3 Aims And Hypothesis

The aims of this work described in this thesis are as follows. Initially it is necessary to review and study the current state of the art with the aim of exposing the strengths and weaknesses of existing methods. 2D-3D registration occurs in a number of guises in the literature. In the computer vision and photogrammetry literature, the problem is known as pose estimation, the location determination problem and in the camera calibration literature as extrinsic parameter calibration. These terms will be described in chapter 2. After reviewing the literature, an algorithm is developed that is based around existing concepts, although significantly different. The performance is studied with the aim of experimentally determining the limits and breaking points of current ideas. Subsequently,

various algorithms are studied with the aim of improving upon current methods. This includes extending the registration framework to include information from multiple optical images simultaneously, using texture mapping to increase robustness, and finally a whole new registration paradigm is developed. The aim is to study the performance of these different algorithms and to carefully validate the performance against high quality gold standards. In addition, the aim is to utilise and develop the use of intensity based methods for 2D-3D image registration, with the goal of producing a method that performs as well as feature based methods, and yet requires little or no segmentation. The hypothesis of this thesis is stated here:

- It is possible to develop an intensity based algorithm to register multiple video images to 3D models or images, that is sufficiently accurate, precise and robust, to be suitable for applications such as radiotherapy patient positioning and also for image guided ENT (ear, nose and throat) surgery, neurosurgery or craniofacial surgery.

By accurate, it is meant that the registration solution is as close as possible to the true registration. A registration error of around 1mm will generally be 'sufficiently accurate'. Precise means that if the registration is repeated many times, the resultant registrations are very similar. Robust means that the algorithm should be able to register images with different initial conditions, images of different quality and images of different content.

1.3.1 Organisation

This thesis is divided into three parts. The first part introduces the problem and provides necessary background information. Chapter 2 describes coordinate systems, camera models, lighting models and various terminology that will be used throughout the thesis. Chapter 3 contains a literature survey. Topics covered include camera calibration, pose estimation, tracking, a framework for image registration, medical image 2D-3D registration including point, contour, surface and intensity based methods and finally a comparison of the most relevant algorithms. The conclusion to chapter 3 gives a specification for the proposed registration algorithms.

The second part contains the experimental work. Chapter 4 describes a single optical image to 3D medical image registration algorithm based on using mutual information. In chapter 5 this mono view algorithm is extended to register multiple optical images to a

3D image. Chapter 6 describes a novel multiple view tracking algorithm which, given an initial, accurate registration, utilises texture mapping to update registration over a series of optical image frames. Chapter 7 introduces a novel framework for multiple optical image to 3D model registration.

The third part contains the conclusions. Chapter 8 summarises the main findings of this thesis, and proposes interesting areas for future research.

The software used in this thesis came from different sources. The main machine used was a Sun Sparc 10, with Elite3D graphics card, 128Mb RAM, running SunOS 5.6 (UNIX). Any interactive segmentation was performed using ANALYZE (Biomedical Imaging Resource, Mayo Foundation, Rochester, MN, USA.). Viewing images was performed using `xv`, written by John Bradley or `rview` written by Colin Studholme. Tsai's camera calibration method was performed using the software written by Reg Wilson, freely available at `http://www.cs.cmu.edu/~rgw/`. The registration software was developed using the Visualization Tool Kit (VTK) [Schroeder *et al.*, 1997]. VTK provides basic image processing functions such as Gaussian filtering of an image, and also many graphical functions, such as surface extraction and surface and volume rendering. All the registration algorithms were implemented by the author by adding similarity measures and a search strategy to VTK, and writing scripts to connect all the necessary components together and coordinate the sequence of events. Adding components to the VTK framework was done using C++, and the scripts were written using Tcl [Ousterhout, 1996]. In addition, the validation and error analysis software was written by the author using a variety of Tcl scripts, C++ and C.

1.4 Contribution And Overview

The main contributions of the work described in this thesis are as follows.

- Chapter 4 describes an algorithm to register a single optical image to a 3D image such as an MR or CT scan. This chapter represents an implementation of an algorithm based on already existing work in the literature [Viola and Wells, 1995]. However, this chapter establishes the limitations of such an algorithm.

- Chapter 5 demonstrates two new, simple ways for registering multiple optical images to a single 3D image using an information theoretic framework. This chapter represents an incremental improvement to the mono view algorithm. The performance of the multiple view algorithm is carefully assessed.

- Chapter 6 describes a new method for updating the registration between sequences of multiple optical images and and a surface derived from an MR or CT scan. The algorithm is a tracking algorithm, utilising texture mapping in an information theoretic framework.

- Chapter 7 introduces a novel framework for registering multiple optical images, and a single 3D scan, based on photo-consistency. Photo-consistency has previously been used for shape reconstruction [Kutulakos and Seitz, 1998]. However, applying this concept to image registration in a novel fashion has led to an accurate, precise and robust algorithm. The algorithm performs well using different numbers of cameras, with significant optical image degradation, for images of different people, for different subsampled images and also lends itself to an efficient implementation. This new framework appears suitable for many applications. It is in this chapter, that the most significant and novel research of this thesis is described.

Chapter 2

Background For 2D-3D Registration

This chapter introduces some key concepts, mathematical notation and terminology used throughout the remainder of the thesis. First, the mathematics used to represent the 2D-3D registration problem is described.

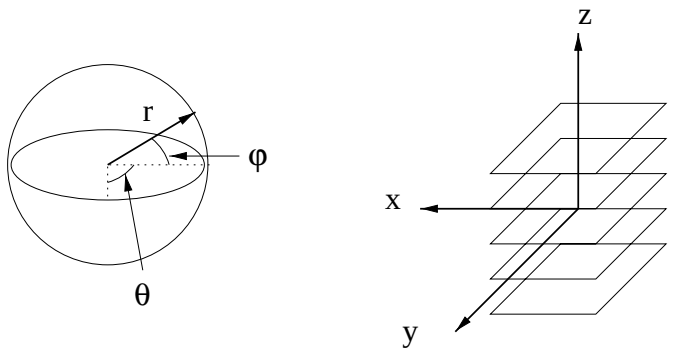
2.1 The 2D-3D Registration Transformation

The task of registration is to find a mapping from spatial locations in one image to the corresponding spatial locations in another. As introduced in chapter 1, this thesis describes algorithms to register a 3D medical image to one or more 2D images. Each image has a coordinate system which defines the spatial locations within that image.

Let coordinates in the 3D image (also called the model) be denoted by $\mathbf{m} = (m_x, m_y, m_z, 1)^T$ and those in a 2D image by $\mathbf{p} = (p_x, p_y, 1)^T$, using homogeneous coordinates. Homogeneous coordinates are used to enable this projection to be represented with a linear transformation as is common in computer vision textbooks [Duda and Hart, 1973; Trucco and Verri, 1998]. The registration problem is then to find a 3×4 transformation matrix $\mathbf{M}$ such that

$$k\,\mathbf{p} = \mathbf{M}\,\mathbf{m} \tag{2.1}$$

where k is a homogeneous coordinate scale factor. The use of equation (2.1) assumes that there is no geometric image distortion, i.e.. the projection geometry of the camera is perfect, and also that there is no deformation between the 3D image, and the 2D image. The validity of these assumptions will be discussed in chapter 3.



(a) (b)

Figure 2.1: The model coordinate systems can be for example (a) spherical or (b) Cartesian.

2.2 Coordinate Systems

The coordinate systems necessary to describe the registration problem are the *model*, *world*, *camera* and *pixel* coordinate systems. These coordinate systems are now described in detail.

2.2.1 The Model Coordinate System

One of the inputs of the registration algorithm is a 3D medical image e.g. MR/CT. The 3D image defines a model of an object of interest e.g. a patient's head. The model can have any coordinate system, for example, a sphere is naturally represented by a spherical coordinate system (r, θ, ϕ), while 3D image slices are naturally represented by a Cartesian coordinate system (x, y, z) as shown in figure 2.1. As the models used in this thesis are derived from a 3D image, the coordinate system of choice is a homogeneous Cartesian coordinate system using millimetres as units. Thus model points are denoted by $\mathbf{m}$, where

$$\mathbf{m} = (m_x, m_y, m_z, 1)^T \tag{2.2}$$

The term model is used to refer to the 3D image being registered, and for compatibility with computer graphics literature is analogous to the term *actor*, used to represent an object being rendered [Schroeder *et al.*, 1997; Foley *et al.*, 1990].

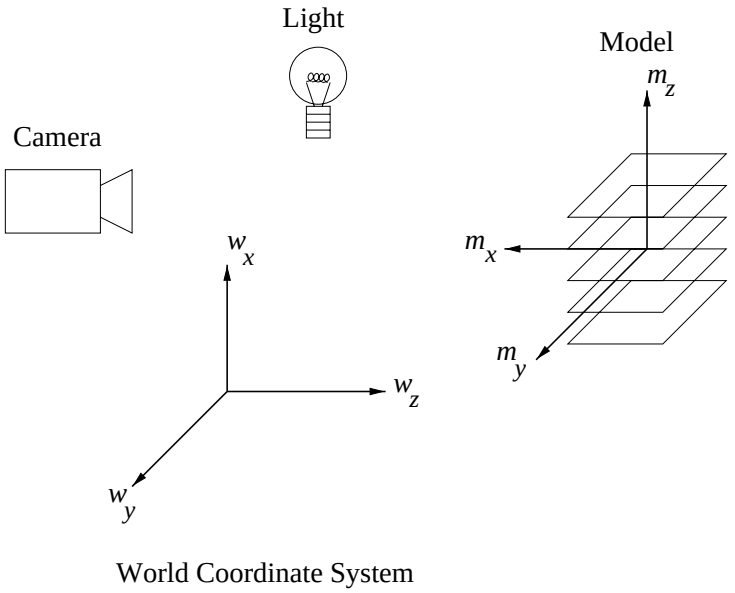


Figure 2.2: The world coordinate system, in which the position of a camera, light source and model can be defined.

2.2.2 The World Coordinate System

In computer graphics, the world coordinate system refers to the virtual world. Figure 2.2 shows a possible computer graphics setup. The model is represented as a set of image slices, however a surface could be constructed using the marching cubes algorithm [Lorensen and Cline, 1987], or the model could be volume rendered directly from the image data. A virtual camera and light source are placed in the virtual world. A rendered image is a picture of what the model 'looks like' as seen from the camera's viewpoint, given the simulated lighting conditions and a reflectance model. A coordinate system is needed to define the position of the camera, light source and model relative to each other. This is the world coordinate system. The world coordinate system in computer vision terms often refers to the real world, where the origin and axis of the world coordinate system are defined by a calibration object, by marks on an object of interest, or by a tracking device. In terms of a 2D-3D registration problem, the world coordinate system serves as a common frame of reference, within which to describe the relative position of the camera, model and light. World coordinates are denoted by $\mathbf{w}$ where

$$\mathbf{w} = (w_x, w_y, w_z, 1)^T \tag{2.3}$$

and the units are millimetres.

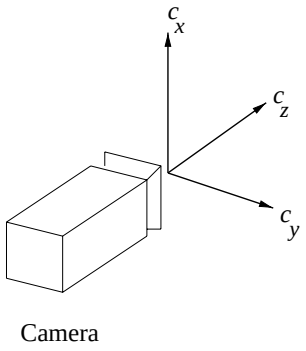


Figure 2.3: The camera coordinate system. The c_z axis is the camera's optical axis.

2.2.3 The Camera Coordinate System

In the camera coordinate system, the orientation of the c_z axis is defined by the camera's optical axes. The orientation of the c_x and c_y axes is defined by the axis of the sensor array within the camera, as shown in figure 2.3. The origin of the camera coordinate system is the centre of projection. The camera coordinates are denoted by $\mathbf{c}$, where

$$\mathbf{c} = (c_x, c_y, c_z, 1)^T \tag{2.4}$$

Camera coordinates have the same units as the world coordinates, (millimetres), but points are measured relative to the camera, i.e. the c_z coordinate describes how far away a point is from the camera imaging plane, in a direction parallel to the camera's optical axis.

2.2.4 The Pixel Coordinate System

The pixel coordinate system refers to the coordinates in the 2D video image. Pixel coordinates are denoted by $\mathbf{p}$ where

$$\mathbf{p} = (p_x, p_y, 1)^T \tag{2.5}$$

The units of $\mathbf{p}$ are pixels. The transformations between model, world, camera and pixel coordinate systems are now described in detail.

2.2.5 The Model To World Coordinate Transformation

Model coordinates are represented by 3D coordinates, with millimetre units, and are denoted by $\mathbf{m}$ (see section 2.2.1). World coordinates are represented by 3D coordinates, with millimetre units, and are denoted by $\mathbf{w}$ (see section 2.2.2). The transformation from model coordinates in millimetres to world coordinates in millimetres can be represented by a 4×4 rigid body transformation matrix ${}^{w}\mathbf{Q}^{m}$. The right superscript m denotes model coordinates and the left superscript w denotes world coordinates. i.e. the matrix ${}^{w}\mathbf{Q}^{m}$ transforms from model to world coordinates:

$$\mathbf{w} = {}^{w}\mathbf{Q}^{m}\,\mathbf{m} \tag{2.6}$$

2.2.6 The World To Camera Coordinate Transformation

The transformation from world to camera coordinates is represented by a 4×4 rigid body transformation matrix ${}^{c}\mathbf{Q}^{w}$. The right superscript w denotes world coordinates and the left superscript c denotes camera coordinates. i.e. the matrix ${}^{c}\mathbf{Q}^{w}$ transforms from world to camera coordinates:

$$\mathbf{c} = {}^{c}\mathbf{Q}^{w}\,\mathbf{w} \tag{2.7}$$

2.2.7 The Degrees Of Freedom Of A Rigid Body Transformation

The matrices ${}^{w}\mathbf{Q}^{m}$ and ${}^{c}\mathbf{Q}^{w}$ are both rigid body transformations. A rigid body transformation is a transformation comprising only rotations and translations. Let t_x, t_y and t_z denote translations in millimetres parallel to the x, y and z axis of some orthogonal coordinate system and r_x, r_y and r_z denote rotations in degrees about the x, y and z axis. Let $\mathbf{R}_x$, $\mathbf{R}_y$ and $\mathbf{R}_z$ be 4×4 matrices to represent the rotations r_x, r_y and r_z respectively and $\mathbf{T}_{xyz}$ be a 4×4 matrix to represent the translations t_x, t_y and t_z respectively. The matrices $\mathbf{R}_x$, $\mathbf{R}_y$, $\mathbf{R}_z$ and $\mathbf{T}_{xyz}$ can be defined as:

$$\mathbf{R}_x = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(r_x) & -\sin(r_x) & 0 \\ 0 & \sin(r_x) & \cos(r_x) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \mathbf{R}_y = \begin{pmatrix} \cos(r_y) & 0 & \sin(r_y) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(r_y) & 0 & \cos(r_y) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.8}$$

$$\mathbf{R}_z = \begin{pmatrix} \cos(r_z) & -\sin(r_z) & 0 & 0 \\ \sin(r_z) & \cos(r_z) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \mathbf{T}_{xyz} = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.9}$$

If these matrices are multiplied together to form a single 4×4 matrix $\mathbf{Q}$, where

$$\mathbf{Q} = \mathbf{T}_{xyz}\, \mathbf{R}_x\, \mathbf{R}_y\, \mathbf{R}_z \tag{2.10}$$

then the resultant matrix $\mathbf{Q}$ is determined by the 6 parameters t_x, t_y, t_z, r_x, r_y and r_z. Thus it is said to have 6 degrees of freedom (DOF). In addition, let $\mathbf{R}_{xyz} = \mathbf{R}_x\, \mathbf{R}_y\, \mathbf{R}_z$, then

$$\mathbf{Q} = \mathbf{T}_{xyz}\, \mathbf{R}_{xyz} \tag{2.11}$$

Thus $\mathbf{Q}$ is constructed from a 4×4 rotation matrix followed by a 4×4 translation matrix. $\mathbf{R}_{xyz}$ and $\mathbf{T}_{xyz}$ each have 3 degrees of freedom.

2.2.8 The Extrinsic Camera Parameters

The transformation from model coordinates to camera coordinates has been defined in sections 2.2.5 and 2.2.6. The previous section described how a rigid body transformation can be defined from six parameters. Therefore the model to world transformation matrix ${}^{w}\mathbf{Q}^{m}$ can be defined using six parameters, and the world to camera transformation matrix ${}^{c}\mathbf{Q}^{w}$ can also be defined using six parameters. The transformations ${}^{w}\mathbf{Q}^{m}$, ${}^{c}\mathbf{Q}^{w}$ were defined as they represent a typical computer graphics framework. However, for the purpose of 2D-3D registration it is possible to make the following simplification.

The composition of two rigid body transformations is itself a rigid body transformation as both are distance preserving transformations. This means the two rigid body transformations ${}^{w}\mathbf{Q}^{m}$, ${}^{c}\mathbf{Q}^{w}$ can be represented by 6 parameters in total. In section 2.2.7, the parameters t_x, t_y, t_z, r_x, r_y and r_z, and matrices $\mathbf{R}_x$, $\mathbf{R}_y$, $\mathbf{R}_z$, $\mathbf{T}_{xyz}$, $\mathbf{R}_{xyz}$ and $\mathbf{Q}$ were used to describe how a general rigid body transformation was formed from 6 parameters, and hence had 6 DOF. The order of matrix multiplication was defined in equations (2.8)-(2.11). Therefore, to simplify notation, let

$$\mathbf{Q} = \mathbf{T}_{xyz}\, \mathbf{R}_{xyz} = \mathbf{T}_{xyz}\, \mathbf{R}_x\, \mathbf{R}_y\, \mathbf{R}_z = {}^{c}\mathbf{Q}^{w}\, {}^{w}\mathbf{Q}^{m} \tag{2.12}$$

To summarise, the transformation from model coordinates to camera coordinates is given by equation 2.12 and hence the parameters t_x, t_y, t_z, r_x, r_y, and r_z. The parameters

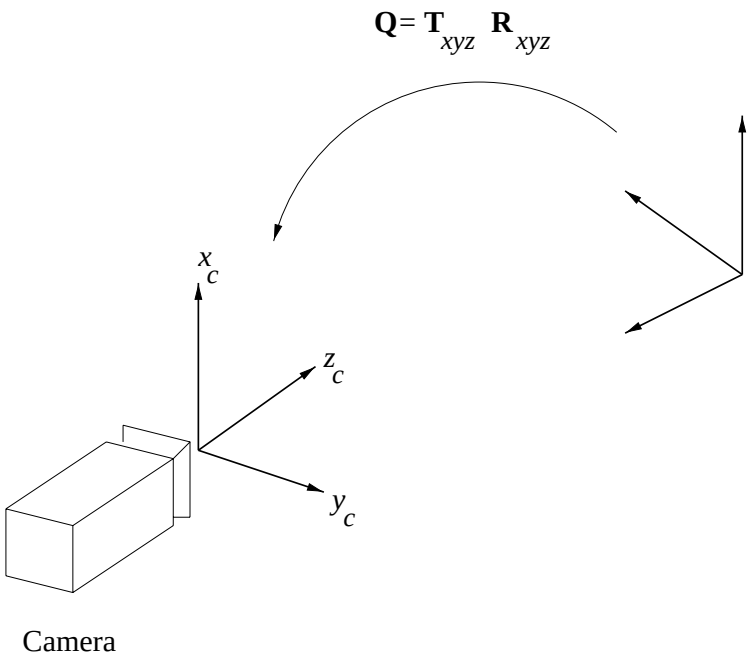


Figure 2.4: The extrinsic camera parameters define the position and orientation (pose) of the camera with respect to another known coordinate system. The camera coordinate system is labelled x_c, y_c and z_c, the pose is determined by the rigid body transformation matrix $\mathbf{Q}$ which comprises of a rotation matrix $\mathbf{R}_{xyz}$ followed by a translation matrix $\mathbf{T}_{xyz}$.

$t_x \ldots r_z$ are called the *extrinsic* camera parameters. The extrinsic camera parameters define the position and orientation (pose) of the camera with respect to another known coordinate system [Trucco and Verri, 1998]. (see figure 2.4).

2.2.9 Camera Models

An image of the 3D world within the field of view of a video camera is formed by projection onto a 2D image plane. Several geometric models for modelling the projection process have been proposed. These are the perspective or pinhole model, the weak perspective or scaled orthographic model [Trucco and Verri, 1998], the para-perspective model [Aloimonos, 1990], the ortho-perspective model [DeMenthon and Davis, 1992a] and the parallel or orthographic projection model [Trucco and Verri, 1998].

Aloimonos [Aloimonos, 1990] characterises the projection process as having the following effects: (a) The distance effect. Objects appear larger when they are closer to the image plane. (b) The position effect. A pattern on an object's surface is distorted by an amount relative to the angle between the line of sight and the image plane. (c) The foreshortening effect. A pattern on an object's surface is also distorted depending on the angle between the surface normal of the surface and the line of sight of the camera.

Assuming that the camera has no geometric distortion, the perspective model is the geometrically correct model and captures the distance, position and foreshortening effects. However, the equations to describe it are non-linear. The other four models can be described with simpler equations. The weak perspective model captures only the distance and foreshortening effects, and the parallel projection model only captures the foreshortening effect. The para-perspective and ortho-perspective models capture the distance, position and foreshortening effects and provide simple equations. Figure 2.5 illustrates the perspective, weak perspective, para-perspective, ortho-perspective and orthographic projection models and these are described below:

(a) **The Perspective Camera Model** With no geometrical image distortion, the perspective or pinhole model is the geometrically correct camera model of those listed above, and is illustrated in figure 2.5(a) and figure 2.6. A point in 3D camera coordinates $\mathbf{c} = (c_x, c_y, c_z, 1)^T$ is projected onto an image plane I_1 at $\mathbf{c}' = (c'_x, c'_y, c'_z)^T$. If the pinhole model were physically constructed, the pinhole would define the optical centre O, the optical axis is OZ and the image of point $\mathbf{c}$ would appear inverted at $\mathbf{c}'$ on I_2. Usually in diagrams such as figure 2.5, the image plane is placed in front of the optical centre. The point $\mathbf{c}'$ on plane I_1 is equivalent to point $\mathbf{c}'$ on plane I_2. f is the camera's focal length.

(b) **The Weak Perspective Camera Model** The weak perspective or scaled orthographic projection model is illustrated in figure 2.5(b). All points are projected orthogonally along rays parallel to the optical axis OZ onto an auxiliary plane, and then projected perspectively. The auxiliary plane should pass through the centre of mass of the point set, but is shown displaced to the left for clarity.

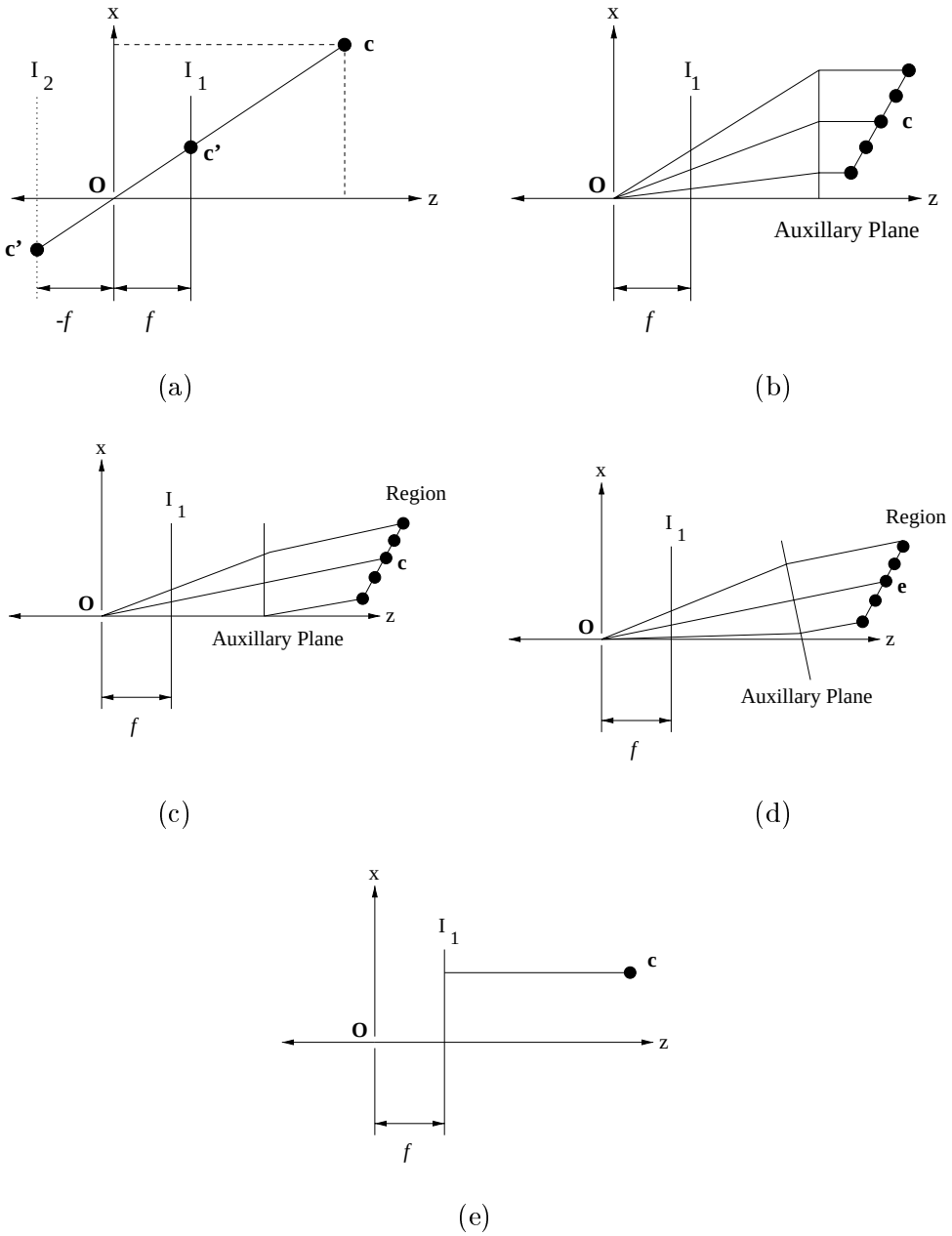


Figure 2.5: 5 different camera projection models. (a) perspective or pinhole model, (b) weak perspective or scaled orthographic model, (c) para-perspective model, (d) ortho-perspective model, (e) parallel or orthographic model. (see text, section 2.2.9).

(c) **The Para-Perspective Camera Model** The para-perspective projection model is illustrated in figure 2.5(c). The points are projected onto an auxiliary plane using rays that are parallel to the ray from the optical centre O to the centre of mass C and then projected perspectively. The auxiliary plane should again pass through the centre of mass of the point set, but is shown displaced to the left for clarity.

(d) **The Ortho-Perspective Camera Model** The ortho-perspective projection model is illustrated in figure 2.5(d). The points are projected onto an auxiliary plane using rays that are parallel to the ray from the optical centre O to the centre of mass C. In this case the auxiliary plane is perpendicular to OC, and should pass through the centre of mass of the point set, but is again shown displaced to the left for clarity. The points are then projected perspectively.

(e) **The Parallel Projection Camera Model** The parallel or orthographic projection model simply projects points onto the image plane I_1 using rays that are parallel to the optical axis and is illustrated in figure 2.5(e).

2.2.9.1 Choice Of Camera Model

The perspective camera model was chosen for the remainder of this thesis. This was because, if geometric distortion is negligible, then the perspective model is the most accurate, and geometrically correct [Aloimonos, 1990]. The algorithms described in this thesis were implemented using VTK [Schroeder *et al.*, 1997] and OpenGL. These libraries provide perspective and parallel camera models, not para-perspective, orthoperspective or weak perspective. Thus perspective projection is readily available in standard graphics implementations.

The weak-, ortho-, and para-perspective and parallel projection camera models are more useful for problems such as surface reconstruction [Trucco and Verri, 1998; Aloimonos, 1990]. The simpler formulations enable simpler numerical algorithms to be developed, but do not offer any advantage for the registration problem.

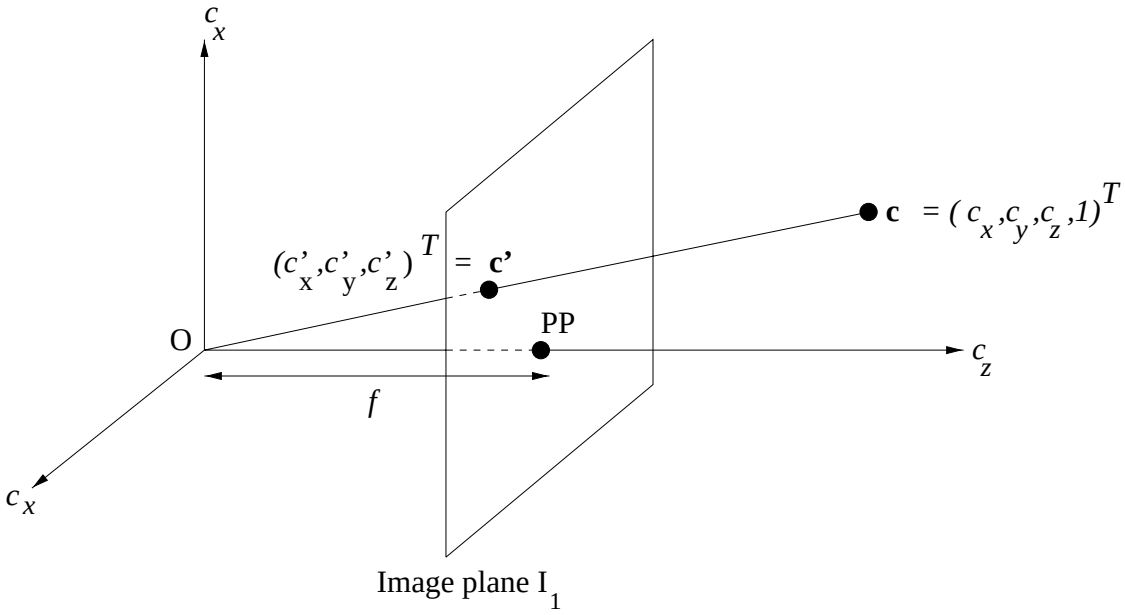


Figure 2.6: The perspective projection camera model. The camera coordinate system is labelled c_x, c_y and c_z. The optical centre is O and the optical axis is Oc_z. The intersection of the optical axis with the image plane is called the *ray piercing point* and is labelled PP. The camera coordinate $\mathbf{c} = (c_x, c_y, c_z, 1)^T$ is projected to $\mathbf{c}' = (c'_x, c'_y, c'_z, 1)^T$. f is the focal length of the camera.

2.2.10 The Camera To Pixel Coordinate Transformation

The camera to pixel coordinate transformation is a projection from the 3D camera coordinates $\mathbf{c}$ measured in millimetres to the 2D video image coordinates $\mathbf{p}$ measure in pixels. The necessary mathematical notation for the perspective projection model will now be developed in detail. The perspective projection model can be found in any graphics or computer vision textbook [Duda and Hart, 1973; Foley *et al.*, 1990; Trucco and Verri, 1998]. As shown in section 2.2.9, the perspective transformation maps camera coordinates $\mathbf{c} = (c_x, c_y, c_z, 1)^T$ onto the image plane I_1 at $\mathbf{c}' = (c'_x, c'_y, c'_z, 1)^T$ according to

$$\begin{aligned} c'_x &= f\frac{c_x}{c_z} \\ c'_y &= f\frac{c_y}{c_z} \\ c'_z &= f\frac{c_z}{c_z} = f \end{aligned} \tag{2.13}$$

The image plane is placed at a distance f from the optical centre. Thus the z-component of the point $\mathbf{c}$ after the perspective transformation is always f and hence is usually ignored. The coordinate $\mathbf{c}'$ is scaled according to a perspective transformation, but it's units are still millimetres. To convert to pixels, two scale factors k_x, k_y and two offsets

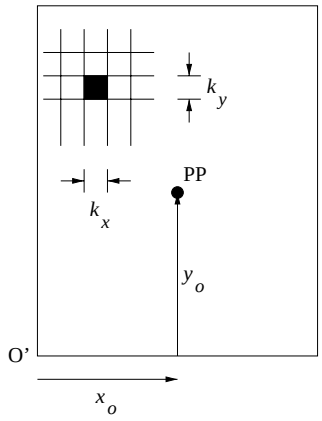


Figure 2.7: To convert coordinates in millimetres to pixels, two pixel scale factors k_x and k_y are required. These scale factors describe how many millimetres each pixel represents. In addition, the intersection of the optical axis is likely to be near the image centre, thus offsets x_o and y_o are required.

x_o and y_o are required, as shown in figure 2.7. These are combined thus;

$$p_x = k_x c'_x + x_o \quad p_y = k_y c'_y + y_o \tag{2.14}$$

So the transformation from 3D camera coordinates $\mathbf{c} = (c_x, c_y, c_z, 1)^T$ to 2D pixel coordinates $\mathbf{p} = (p_x, p_y, 1)^T$ is given by

$$p_x = x_o + \frac{k_x\, c_x\, f}{c_z} \quad p_y = y_o + \frac{k_y\, c_y\, f}{c_z} \tag{2.15}$$

The perspective transformation can be written in matrix form as

$$k\,\mathbf{p} = \mathbf{P}\,\mathbf{c} \tag{2.16}$$

where

$$\mathbf{P} = \begin{pmatrix} k_1 & 0 & x_o & 0 \\ 0 & k_2 & y_o & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{2.17}$$

and $k_1 = k_x f$, $k_2 = k_y f$ and k denotes the homogeneous scale factor.

2.2.11 The Intrinsic Camera Parameters

Section 2.2.10 described the projection from camera coordinates $\mathbf{c}$ to 2D pixel coordinates $\mathbf{p}$. Specifically, the perspective or pinhole camera model was parameterised using k_1, k_2, x_o and y_o which are the x and y pixel scale factors and the intercept of the optical

axis with the image plane. These parameters describe an ideal projection process which occurs within a perfect pinhole camera, and are called the *intrinsic* camera parameters. In the computer vision and photogrammetry literature, camera models to account for various forms of lens distortion have been developed. These models require more parameters, and are discussed in detail in chapter 3. For now, the *intrinsic* parameters are defined as those necessary to link the coordinates in the camera coordinate system with the corresponding pixel coordinates.

2.2.12 The Complete Model To Video Coordinate Transformation

In summary, the transformation from model coordinates $\mathbf{m} = (m_x, m_y, m_z, 1)^T$ to pixel coordinates $\mathbf{p} = (p_x, p_y, 1)^T$ can be represented by

$$k\,\mathbf{p} = \mathbf{P}\,\mathbf{Q}\,\mathbf{m} \tag{2.18}$$

Comparing equation (2.1) with equation (2.18) reveals that

$$\mathbf{M} = \mathbf{P}\,\mathbf{Q} \tag{2.19}$$

2.3 Camera Calibration, Pose Estimation And 2D-3D Registration

The terms camera calibration, pose estimation and 2D-3D registration need to be clarified before the literature review in chapter 3. In section 2.2.8, the camera extrinsic parameters were defined as the three rotations and three translations that relate the camera coordinate system to some other known coordinate system [Trucco and Verri, 1998]. In section 2.2.11, the intrinsic parameters were defined as those necessary to link the coordinates in the camera coordinate system with the corresponding pixel coordinates. The term pose means position and orientation. Thus pose estimation is defined to be the process of computing the camera's extrinsic parameters, i.e. the position and orientation of the camera with respect to another known coordinate system. Consequently pose estimation algorithms usually assume that the camera's intrinsic parameters are known [Lowe, 1987; Yuan, 1989; Phong *et al.*, 1995]. The term camera calibration may refer to the process of computing a camera's intrinsic, extrinsic or both intrinsic and extrinsic parameters [Nomura *et al.*, 1992; Lowe, 1987; Tsai, 1987]. The term 2D-3D registration in the context of this thesis is to find a transformation from 3D image coordinates to the corresponding 2D image coordinates.

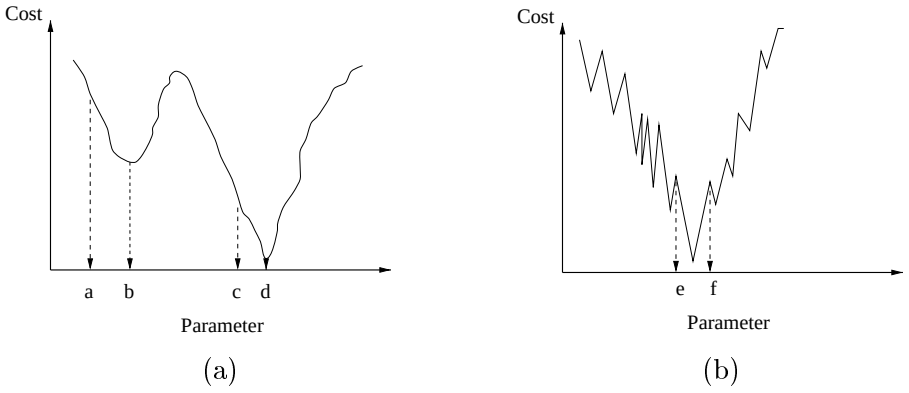


Figure 2.8: (a) A smooth and (b) rough 'parameter space'. See text section 2.4.

Mathematically, a camera calibration procedure and a 2D-3D registration procedure that both calculate intrinsic parameters and extrinsic parameters are equivalent. However, in practice these methods are often combined for a given application. Consider guiding a mobile robot using a video camera. If the focal length and zoom of the camera are known to be constant, the intrinsic parameters could be calculated to a high degree of accuracy off line, using a camera calibration procedure. The extrinsic parameters, i.e. the position and orientation with respect to the robots environment, could then be calculated as the robot goes about its task, using a pose estimation procedure. Likewise for 2D-3D registration. If the application is to register optical to 3D medical images and it is known that the intrinsic parameters will remain fixed, then these can be calculated using a calibration procedure, thereby reducing the registration task to that of pose estimation.

2.4 Search Space

The term search space is used to refer to the n dimensional space of an n dimensional optimisation problem. An optimisation problem would be to maximise (or minimise) some function with respect to a set of n parameters. 2D-3D registration can be viewed as an optimisation problem. Assume that for the camera model, the intrinsic parameters are known, and that some function F is defined such that for a given set of extrinsic parameters (i.e. $n = 6$), F measures the cost of the registration. The registration problem is then to adjust the six parameters to minimise F. The search space can be plotted on an $n+1$ dimensional graph. Figures 2.8 (a) and (b) show two 1D search spaces. The cost function has been plotted on the vertical axis against the parameter on the horizontal

axis. Consider the graph in figure 2.8(a). For this example, the global minimum and the correct solution is at d. A typical gradient based optimisation strategy would be to pick a starting point e.g. c, calculate the gradient and make repeated steps downhill, until the minimum is reached. However if the starting position was a, this algorithm would finish at b i.e. a local minimum and the incorrect solution. The range of capture refers to the distance in search space from the correct solution within which the algorithm will converge to the correct solution. In optimisation, it is important to know how smooth the search space is. In figure 2.8, graph (a) is much smoother (simpler) than graph (b). Consequently graph (b) has a much smaller capture range (e $\leq$ parameter $\leq$ f). A search space like graph (a) makes the optimisation more robust as it makes it easier for an algorithm to find the correct minimum. In some cases however, the correct solution may correspond to a local minimum. A discussion of optimisation strategies and the issues involved can be found in [Press *et al.*, 1992], and [Maes, 1998] for a medical 3D-3D registration example.

2.5 Surface Reflectance And Reflection Models

The computer graphics community is interested in producing realistic images using computers. Thus, different lighting models have been studied which, when used to render a computer image, provide varying levels of realism. Below are brief descriptions of some of the terms and models used to describe different lighting effects. Sections 2.5.1 to 2.5.4 are a summary of [Foley *et al.*, 1990] pages 722 to 731. See [Foley *et al.*, 1990] for much more detail on lighting models.

2.5.1 Ambient Reflection

Consider an object that is lit by a non-directional light source. For example, in a room with many light sources and light inter-reflecting off surrounding objects there appears to be a general overall illumination which can not be attributed to a given light source. This is called ambient illumination. The object will be illuminated from all directions, and if it reflects equally in all directions, then the observed intensity I can be described by

$$I = I_a k_a \tag{2.20}$$

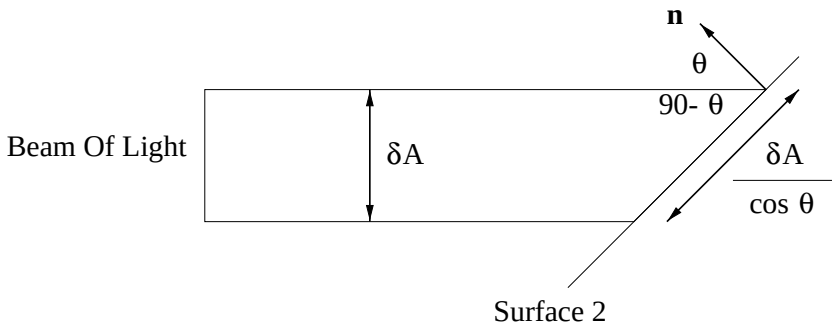


Figure 2.9: The Lambertian reflection model, reproduced from [Foley *et al.*, 1990]. Incident light varies with $\cos\theta$ and reflected light varies inversely with $\cos\theta$. See text section 2.5.2.

where I_a is the intensity of the ambient light and k_a is the coefficient of ambient reflection where $0 \leq k_a \leq 1$ The coefficient k_a is a material property, characterising the intrinsic colour or surface type from which an object is made. With this model, no shading is apparent.

2.5.2 Diffuse Reflection

Now consider illuminating an object by a single point light source, where the source emits light equally in all directions. Diffuse reflection, also called Lambertian reflection, is the type of reflection exhibited by dull matte surfaces, e.g. chalk. The mechanism which causes diffuse reflection is internal scattering of light in the microscopic inhomogeneities in the surface medium. Some light is absorbed, and due to the random nature of the scattering, the rays that are reflected are done so in a variety of directions, resulting in diffuse reflection [Nayar *et al.*, 1991], where light is reflected on average, equally in all directions.

The observed intensity at a surface point is dependent on the angle between the surface normal and the direction from the point to the light source. Consider figure 2.9 where a point light source emits a beam of light of width δA which intercepts an area of $\delta A/\cos\theta$ on surface 2. The area $\delta A/\cos\theta$ is inversely proportional to $\cos\theta$. Thus, incident light energy per unit area is proportional to $\cos\theta$. According to Lambert's law however, the light reflected towards a viewer is proportional to the cosine of the angle between the viewer and the surface normal $\mathbf{n}$. However, the amount of surface area seen by the viewer is inversely proportional to the cosine of the angle between the viewer and the surface

normal **n**. Thus the two cosine terms cancel out and the light energy observed by the viewer is independent of viewer direction and proportional to $\cos\theta$, the angle between the incident light and the surface normal. Thus, for diffuse reflection the observed intensity I is

$$I = I_p k_d \cos\theta \tag{2.21}$$

where k_d, $0 \leq k_d \leq 1$ is the diffuse reflection coefficient and I_p is the intensity of a point light source.

2.5.3 Specular Reflection

Specular reflection describes the reflection seen from a shiny surface. Surfaces such as plastics, metals and varnished ceramics will appear with bright highlights caused by surrounding light sources. Figure 2.10 shows a sphere, with surface normal **n**, a vector pointing towards the light source **l**, a vector pointing towards the viewer **v**, and the reflection of the light vector about the surface normal **r**. Specular reflection is observed when the angle α is small, i.e. near zero for metal, and zero for a perfect mirror.

2.5.4 The Phong Lighting Model

The Phong lighting model is a popular model in computer graphics, [Foley *et al.*, 1990]. The Phong model can be seen as a combination of ambient, diffuse and specular terms. The intensity at a point is given by

$$I = I_a k_a O_d + f_{att} I_p [k_d O_d (\mathbf{n} \cdot \mathbf{l}) + k_s (\mathbf{r} \cdot \mathbf{v})^n] \tag{2.22}$$

where I_a is the ambient light intensity, O_d is the object's diffuse colour, k_a, k_d and k_s are the coefficients of ambient, diffuse and specular reflection respectively, f_{att} is an atmospheric attenuation coefficient, I_p is the point light source intensity and $\mathbf{n}, \mathbf{l}, \mathbf{r}$ and $\mathbf{v}$ are the vectors representing the surface normal, the direction towards the light, the reflection vector and the direction towards the viewer respectively (see figure 2.10). The parameter n controls the radius of the observed specular highlights. The light source attenuation factor f_{att} makes objects further from the light source appear dimmer, i.e. $f_{att} = 1/d^2$ where d is the distance from a surface point to the light source. Equation (2.22) simply describes intensity as a single valued quantity, i.e. grey scale intensity. For colour images, the equation can be evaluated for each component of colour, such as red, green and blue (RGB) and then the intensities combined.

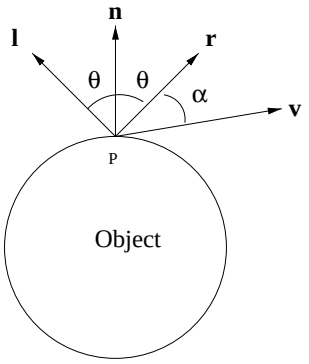


Figure 2.10: Angles and vectors for the Phong lighting model. The vector **l** points to the light source, **n** is the surface normal, **r** is the light source vector, reflected about **n** and **v** is the vector pointing to the viewer.

Furthermore, equation (2.22) is often implemented using a separate specular colour O_s, so that

$$I = I_a k_a O_d + f_{att} I_p [k_d O_d (\mathbf{n} \cdot \mathbf{l}) + k_s O_s (\mathbf{r} \cdot \mathbf{v})^n] \tag{2.23}$$

which can be used to describe specular highlights that are not the same colour as the diffuse colour e.g. a diffuse red sphere with green specular highlights. Figure 2.11 shows three red spheres rendered with (a) purely ambient light, (b) purely diffuse light and (c) a mixture of diffuse and specular light. In these images, the sphere position, light position and viewing direction are identical. In image (a) the ambient reflection gives no shading information. In (c), 50% diffuse lighting is used to illuminate half the sphere, whilst the specular reflection is demonstrated by the white highlight.

2.5.5 Relevance Of Lighting Models To This Thesis

The notation in the previous sections gives us terminology to describe what is observed in the video images. For instance consider the images in figure 2.12. The skull phantom in image (a) appears to have dull, matte reflection, which would seem to fit the description of the Lambertian surface. However, the skull phantom has painted black spherical fiducials attached to it, which have white dots at the centre of each fiducial. These white dots can be described as specular reflection. In image (b), the skin texture of the volunteer appears to be reasonably diffuse, but there is specular reflection at the tip of the nose, and at

(a) (b) (c)

Figure 2.11: Three images of a rendered white sphere. (a) Ambient reflection. (b) Diffuse reflection. (c) Diffuse reflection with specular highlight. i.e. $I_a = 0, k_a = 0, I_p = 1, k_d = 0.5, I_s = 1, k_s = 1, n = 50$.

(a) (b)

Figure 2.12: Two images which exhibit diffuse and specular reflection. See text. (a) Skull Phantom, (b) An image from dataset 'matt'

the centre of the eyes. A video image is an image of light reflected from surfaces within the scene. For registration purposes, it is important to consider how surface reflection will affect the accuracy of alignment. If features are extracted from the video image, it is important that the accuracy of the feature localisation is unaffected by different reflections. If intensities are used directly to perform the alignment, then it is important that the resultant registration is not affected if the overall scene illumination changes. Nayar [Nayar *et al.*, 1991] describes two, more complex reflectance models, the Torrance-Sparrow model and the Beckman-Spizzichino model of surface reflectance. The Torrance-Sparrow model is a geometric model. Assuming that the wavelength of incident light is small relative to the surface irregularities, geometric arguments are used to develop a

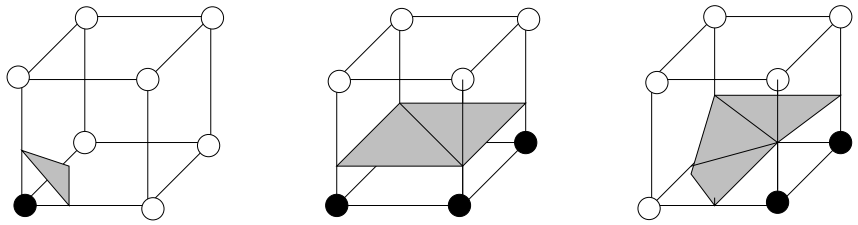


Figure 2.13: Three cases to demonstrate the marching cubes algorithm. Black spheres represent voxels whose intensity is below a threshold, white spheres represent voxels whose intensity is above a threshold. The shaded polygons represent the polygons formed as a result of linearly interpolating the position of the iso-surface.

model for specular reflection, which can be combined with the familiar Lambertian model to give more realistic effects than the Phong model. The Beckman-Spizzichino model used Maxwell's wave equations to derive a physics based model of reflectance, which models two types of specular reflection. These models are more complicated than Phong's [Foley *et al.*, 1990] but also more realistic. Chapter 3 reviews the current literature in camera calibration, and pose estimation from a computer vision, and medical imaging viewpoint. The interesting question is how existing algorithms have treated these issues of changing reflectance and illumination, and whether complicated modelling of reflectance properties is indeed necessary for accurate registration. It will be seen that in general, computer vision algorithms do not use these complicated, but more realistic lighting models as they work sufficiently well with the simpler models.

2.6 Surface Models

The work described later in this thesis registers a 2D video image to a 3D image. The video image will show a specific surface present in the world scene. The 3D image however is represented initially by simple voxel intensity data. To perform the registration, a surface model is extracted from the 3D image. This is performed using standard computer graphics techniques. The surface is extracted using the marching cubes algorithm [Lorensen and Cline, 1987] implemented in Visualization Toolkit (VTK) [Schroeder *et al.*, 1997]. The marching cubes algorithm extracts an iso-intensity surface from a volume data set. This is performed by processing through each voxel in turn, and looking at the neighbouring voxels. If some of the surrounding voxels cross

(a) (b) (c)

Figure 2.14: (a) A surface model is represented by points, (b) which are connected together with lines to form polygons, (c) and each polygon is rendered using a shading model.

the intensity threshold, the surface is defined using a plane which linearly interpolates the position of the surface between the voxels. For a voxel, there are 256 permutations of the 8 corners being above or below an intensity threshold, which due to symmetry reduces to 16 cases of interest for where a surface could be placed. Figure 2.13 illustrates 3 such cases. Black spheres represent voxels whose intensity is below a threshold, and white spheres above. The three cases show in shaded grey the polygons that the marching cubes algorithm would form. For further details see [Schroeder *et al.*, 1997; Foley *et al.*, 1990]. For image points in a regular grid, the end result is a set of points and lines defining polygons that fit an iso-intensity surface between voxels. This can then be rendered by drawing each polygon using standard graphics techniques. A similar example is illustrated in figure 2.14. A surface is represented as (a) a set of points, (b) which are joined together with edges to define the connectivity of the points, and (c) each polygon is rendered according to a lighting model.

Chapter 3

Review Of 2D-3D Image Registration

The previous chapter described camera models, and defined the camera intrinsic and extrinsic parameters. In section 2.3 it was noted that pose estimation involves determining the extrinsic parameters, camera calibration involves determining the intrinsic and/or extrinsic parameters, and that 2D-3D registration involves finding both the intrinsic and extrinsic parameters. This chapter reviews techniques used to register 2D and 3D images or models and is organised as follows.

Camera calibration procedures are reviewed, followed by pose estimation algorithms from the computer vision literature. Subsequently, tracking algorithms are reviewed. Tracking is the process of registering a sequence of images, taken over time, with the knowledge that the change in the registration transformation between each image in the sequence is likely to be small. These algorithms are assessed and it is summarised that they are not applicable to the optical image to 3D medical image registration task. Thus they are not implemented in this thesis. A framework for classifying registration algorithms [Brown, 1992] is introduced, and the terminology is used throughout the thesis. Then the current state of the art in terms of medical 2D-3D image registration is reviewed, where the 2D image can be either X-ray, fluoroscopy, or video. Finally, a comparison of methods and a brief specification are provided.

Brown provides a thorough review of image registration [Brown, 1992], van den Elsen [van den Elsen *et al.*, 1993], Maurer [Maurer Jr. and Fitzpatrick, 1993] and chapter 3 of Maintz [Maintz, 1996] all review medical image registration. Lavallee specifically reviews registration for image guided surgery [Lavallee, 1996]. The aim of this chapter is to concentrate on relevant 2D-3D registration material where the 2D image is a projection of the 3D volume of interest.

3.1 Camera Calibration

Camera calibration has long been an important issue in the field of photogrammetry [Weng *et al.*, 1992]. Photogrammetry is the process of making measurements from photographs, where accuracy is of extreme importance. In computer vision, camera calibration procedures enable one to relate image measurements to the spatial structure of the observed scene or infer 2D image coordinates from 3D information. The former finds application in surface reconstruction, tracking and robot vehicle guidance, and the latter in mechanical part inspection amongst other things. In computer vision, it can be more important for a calibration algorithm to be fast and robust, rather than being highly accurate.

Tsai provides a detailed survey of camera calibration techniques [Tsai, 1987]. Weng categorises camera calibration methods into three categories, namely, closed form solutions, two stage methods and nonlinear minimisation [Weng *et al.*, 1992].

3.1.1 Closed Form Solutions

Closed form solutions compute camera parameters analytically. The transformation from 3D-2D coordinates is non-linear with respect to the camera parameters. Weng defines intermediate linear equations to solve for both the intrinsic and extrinsic camera parameters. He states that in general, algorithms that compute a closed form solution are fast, cannot incorporate any camera distortion parameters and in the presence of noise the accuracy is poor. Weng therefore uses his closed form solution to initialise a non-linear minimisation [Weng *et al.*, 1992].

3.1.2 Two Stage Methods

The transformation from 3D world coordinates $\mathbf{w} = (w_x, w_y, w_z, 1)^T$, to 2D pixel coordinates $\mathbf{p} = (p_x, p_y, 1)^T$ is given by

$$k\,\mathbf{p} = \mathbf{M}\,\mathbf{w} \tag{3.1}$$

where k is the homogeneous scale factor and the matrix $\mathbf{M}$ is a 3×4 transformation matrix where

$$\mathbf{M} = \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{pmatrix} \tag{3.2}$$

The matrix **M** is defined up to an arbitrary scale factor and therefore has 11 unknowns [Trucco and Verri, 1998] i.e. the whole matrix **M** can be normalised by dividing through by m_{34} so that the resultant matrix always has $m_{34} = 1$, leaving 11 remaining numbers and hence 11 DOF. Though equation 3.1 is non-linear with respect to the underlying intrinsic and extrinsic camera parameters, it is linear with respect to the elements of matrix **M**. Given at least six, but in practice many more, pairs of corresponding 2D and 3D points, the matrix **M** can be calculated in the least squares sense [Ballard and Brown, 1982; Gonzalez and Woods, 1992; Trucco and Verri, 1998], using singular value decomposition (SVD). If all that is required is to project 3D points to corresponding 2D points, then the matrix **M** can be considered to represent a calibrated camera.

Rougee makes the assumption that the 2D pixel size is square and formulates a pinhole camera model using six extrinsic and three intrinsic camera parameters [Rougee *et al.*, 1993]. The classic pinhole camera model described in chapter 2 uses six extrinsic and four intrinsic camera parameters [Ganapathy, 1984; Faugeras, 1993; Trucco and Verri, 1998]. This means that the matrix **M** has 11 unknown parameters, whereas the underlying camera model may have 9 or 10. So performing the calibration by calculating the matrix **M** may provide unstable results as the matrix **M** can describe arbitrary linear transformations, and not only projections [Rougee *et al.*, 1993].

Two stage methods then proceed to calculate the camera parameters from the matrix **M**. Ganapathy [Ganapathy, 1984], Strat [Strat, 1984] and Faugeras [Faugeras, 1993] describe methods for extracting 10 camera parameters. The extracted parameters can be unstable depending on the number of points used, the point configuration and the accuracy with which the points are determined. Faugeras adds another parameter θ, the angle between the image plane axes, and provides a method for extracting 11 parameters from the matrix **M**. However, this method will still suffer due to noise on the image points, and whether the underlying matrix **M** does describe a projection or an arbitrary linear transformation.

Image distortion is modelled using extra intrinsic parameters. Figure 3.1 illustrates radial, tangential, barrel and pincushion distortion. Tsai proposed a two stage technique to determine a camera model which incorporates first order radial lens distortion requiring a single extra intrinsic parameter [Tsai, 1987]. Tsai's method uses a closed form solution to derive the extrinsic parameters and an initial estimate of the intrinsic focal length and radial distortion coefficient. This is followed by an iterative update of the translation

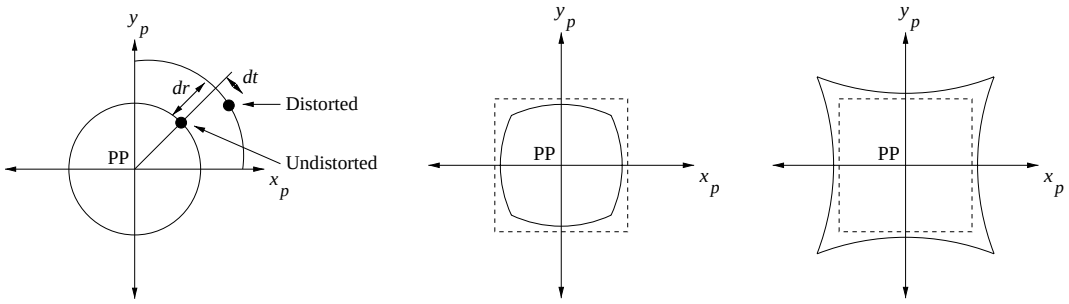
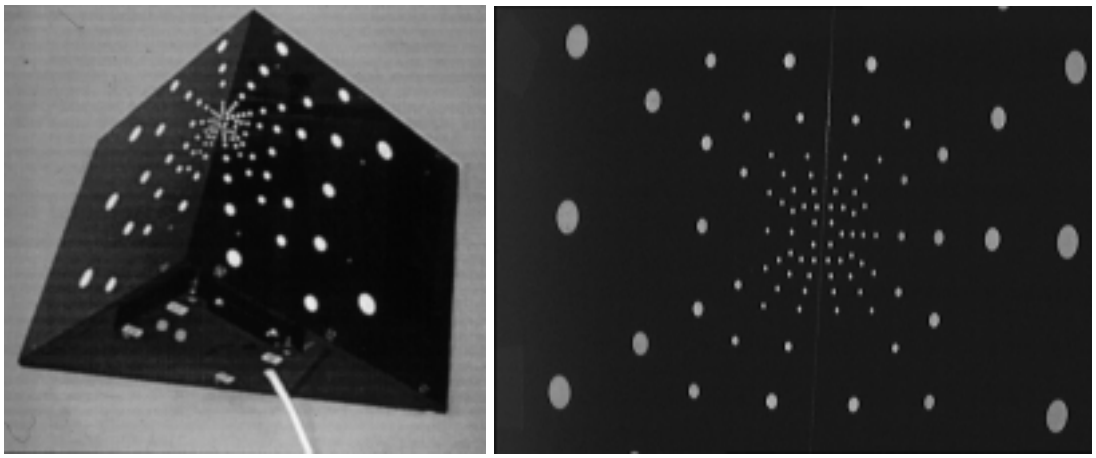


Figure 3.1: (a) Radial distortion *dr* and tangential distortion *dt*. (b) Barrel distortion - a square is distorted inwards i.e. negative radial distortion. (c) Pincushion distortion - a square is distorted outwards, i.e. positive radial distortion.

with respect to the optical axis, the focal length and the radial distortion coefficient. The original method did not include calibration of the image centre, which Tsai claims does not affect the accuracy of 3D measurement. The advantage of Tsai's method is the closed form solution for most of the parameters, and an iterative solution is only required for three parameters which should therefore perform well. Further work by Wilson [Wilson, 1994] which includes freely available software has extended Tsai's method to optimise all 11 parameters. A disadvantage is that Tsai's method only calculates radial distortion, and cannot be easily extended to calculate further types of distortion.

3.1.3 Non-Linear Methods

Non-linear optimisation methods have been used to calibrate pinhole camera models without modelling camera distortion effects. The cost function minimised is usually the squared Euclidean distance between 2D calibration points and the corresponding 3D calibration points projected onto the 2D image plane using the current estimate of the camera parameters. Fleig *et al.* use a Levenberg-Marquardt optimisation [Press *et al.*, 1992] to calibrate the optics of a stereo operating microscope (LEICA M695) where distortion effects were negligible [Fleig *et al.*, 1998]. Rougee used a conjugate gradient descent algorithm [Press *et al.*, 1992] to calibrate an X-ray set, which is also a perspective projection problem [Rougee *et al.*, 1993]. Both Fleig and and Rougee comment on the improvement of the non-linear method over the classical least squares type linear solution mentioned in the previous section. Fleig studies camera calibration for varying zoom and focus settings using a specially designed calibration pattern (see figure 3.2).



(a) (b)

Figure 3.2: Two views of a calibration object used for calibrating multiple zoom and focus settings.

Robert implements a calibration method that does not explicitly require 2D point locations [Robert, 1996]. Tsai's method projects 3D calibration points to 2D and measures the distance between the projected 3D point and it's corresponding 2D calibration point. The cost function minimised is the sum of the squared 2D distances for each calibration point. Robert's method however, takes 3D calibration points and projects them to find the 2D image location. The quantity optimised is the sum of the gradient (maximised) or the sum of the Laplacian values (minimised) of the image intensities at each 2D point. Robert claims that the method is easier to use than classical point matching methods, and exhibits good convergence to the solution, but requires a close initial estimate. In principle this could be extended to also optimise image distortion parameters.

When considering image distortion effects, opinions differ as to what types of distortion are significant. In the photogrammetry literature, where accuracy is very important, Faig's method optimises 17 parameters [Faig, 1975]. Faig models radial lens distortion, decentering, film deformation, affinity and non-perpendicularity of comparator (image) axis. This contrasts with Tsai's method where his "experience shows that for industrial machine vision applications, only radial distortion needs to be considered, and only one term is needed" [Tsai, 1987].

Weng performs nonlinear minimisation to optimise the extrinsic and intrinsic parameters in a camera model which accounts for radial distortion, decentering and thin prism distortion. Decentering occurs when the optical centres of lens elements are not exactly

collinear and thin prism distortion arises from imperfect lens design and camera assembly e.g. a slight tilt of the sensor array. In total, Weng uses five distortion coefficients and demonstrates an improved performance over Tsai's method. [Weng *et al.*, 1992].

3.1.4 Other Methods

Instead of using a geometrical model such as the perspective camera model and then adding distortion parameters to improve accuracy, Champleboux uses a purely interpolative model to characterise cameras and range imaging sensors [Champleboux *et al.*, 1992]. The method, called N-Planes B-Splines (NPBS) takes several (at least 2) images of a calibration pattern and associates 2D image points with a line of sight in space using B-Splines. This means that a distortion field that varies in an arbitrary pattern over the 2D image plane can be accommodated within this model.

Methods exist which calibrate only a subset of the possible parameters in a camera model. Penna describes a method for determining the ratio of pixel dimensions in the horizontal and vertical directions [Penna, 1991]. Nomura describes a method for calibrating the internal parameters focal length, one pixel width (scale factor), image distortion centre and distortion coefficient [Nomura *et al.*, 1992]. Beardsley uses more modern projective geometry methods to demonstrate a solution for the intrinsic parameters of a distortion free model [Beardsley *et al.*, 1992]. Wang calculates the orientation, position (extrinsic parameters) and focal length using vanishing lines and a distortion free model [Wang and Tsai, 1991]. Abidi also computes the extrinsic parameters and focal length using an efficient analytic solution derived specifically for quadrangular targets [Abidi and Chandra, 1995].

Mellor proposed a simple method to calculate the perspective projection matrix $\mathbf{M}$ directly [Mellor, 1995]. For augmented reality applications this can be sufficient. Figure 3.3 shows some of his images. A single video view is used, and circular fiducials are tracked. These fiducials of known size, enable a perspective projection to be calculated and then virtual graphics overlaid.

One of the problems of the methods mentioned thus far is that calibration must be done off line. Accurately constructed calibration objects or calibration hardware is used to perform the calibration before the calibrated camera is applied to some task. Recent interest in computer vision is in self-calibration where calibration is performed using points and landmarks within the field of view during the task of interest. Faugeras

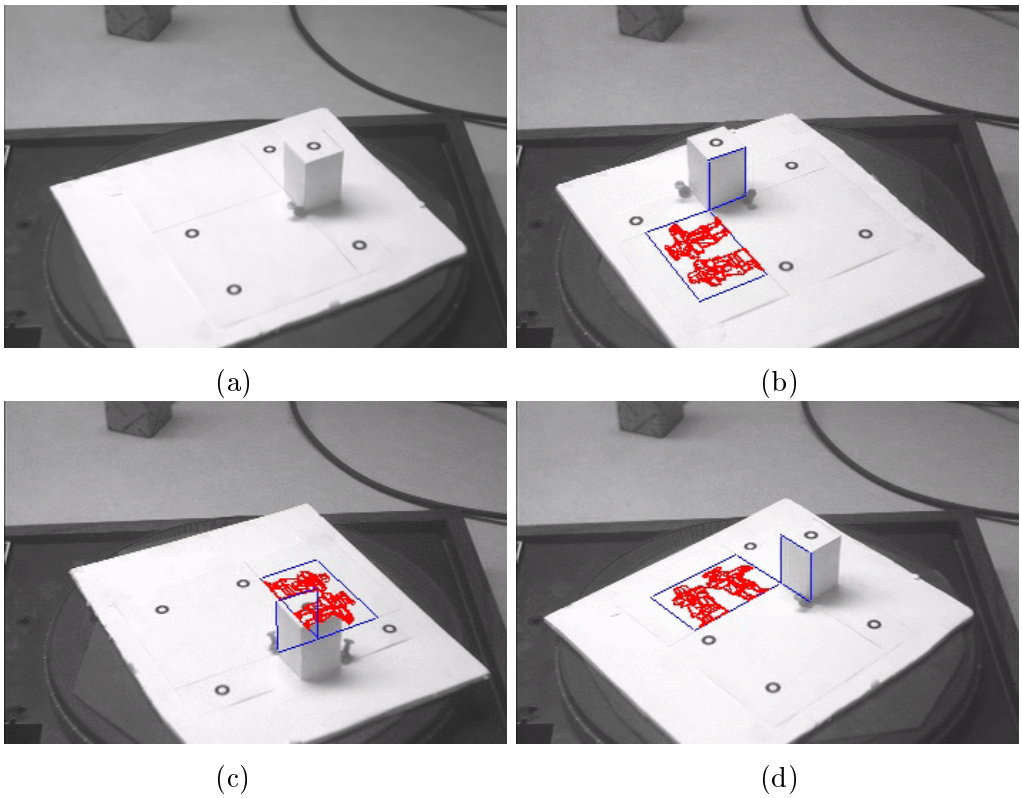
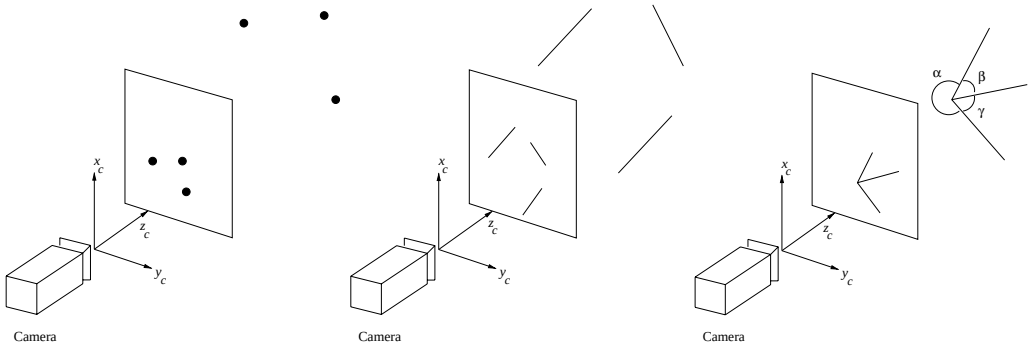


Figure 3.3: (a) Example video image from sequence. The fiducials are tracked. (b), (c) and (d) show blue and red rendered virtual overlays. Images used with permission, thanks to J. P. Mellor.

et al recover intrinsic camera parameters from sequences of images by tracking points over three frames without knowing the motion of the camera [Faugeras *et al.*, 1992]. In addition, performing visual tasks without camera calibration at all [Faugeras, 1992], has been proposed and is based around fundamental matrix theory. The fundamental matrix is a matrix which describes the transformation between pixels in two images, and thus contains all necessary information regarding the intrinsic parameters. Identifying point correspondences between two images is sufficient to construct a projective representation of the environment. This may be sufficient for various robot vision tasks, eliminating the need for full calibration. However, results seem to be dependent on identifying extremely accurate corresponding points in sets of images [Hartley, 1997]. Subsequent papers have used this un-calibrated paradigm for point correspondence in two views [Pilu and Lorusso, 1997], trinocular reconstruction [Ayache and Lustman, 1991; Faugeras and Robert, 1994] and augmented reality [Kutulakos and Vallino, 1998].



(a) (b) (c)

Figure 3.4: Pose can be obtained from points, lines and angles (a) The Perspective-3-Point (P3P) problem (b) The Perspective-3-Line (P3L) problem (c) The Perspective-3-Angle (P3A) problem.

3.2 Pose Estimation

As mentioned in section 2.3, pose estimation refers to finding the position and orientation of a camera with respect to a known coordinate system, or in other words, finding the camera's extrinsic parameters. Pose estimation is often performed after a recognition process. In a car tracking experiment it may be necessary to identify from a video image whether a car is present in the image before trying to estimate its pose with respect to a video camera. If an explicit model of the object of interest is available, e.g. a CAD model of a car, then this problem is known as model based object recognition. If a 3D model is not present, recognition can still be performed by comparing a test image against a database of previous 2D images. This is known as view or appearance based object recognition. However, for the purpose of this literature review, the emphasis is on pose estimation as opposed to object recognition.

3.2.1 Model Based Pose Estimation

Model based pose estimation assumes that a model of an object of interest, e.g. a CAD model of a car or a set of points on a 3D object, is known. Thus pose estimation relates the model coordinate system to the camera coordinate system. Given the intrinsic camera parameters, Fishler and Bolles showed that a solution can be obtained with only three non-collinear pairs of corresponding 3D and 2D points [Fishler and Bolles, 1981]. Fishler and Bolles coined the term Perspective-N-Point problem (PNP) to refer to pose estimation solved using N pairs of corresponding 3D and 2D points. e.g. the perspective-3-point problem is called "the P3P problem" and is illustrated in figure 3.4(a). They

also showed that using three points can result in up to four solutions. Wolfe provides geometric justification that most of the time, only two solutions will be found [Wolfe *et al.*, 1991]. Furthermore, Fishler and Bolles provide a solution to the P4P problem, showing that with 4 coplanar points, a unique solution exists, but with 4 non-coplanar points, a unique solution can not be guaranteed.

Haralick provides a review of six major direct P3P solutions, starting with a solution from Grunert in 1841, and also notes that when comparing these methods, the relative error observed between these 6 algorithms can change by over a thousand to one [Haralick *et al.*, 1994].

The P3P problem has also been addressed using weak perspective [Alter, 1994], paraperspective and orthoperspective [DeMenthon and Davis, 1992a] projection. Weak perspective projection is known to approximate true perspective projection well if the size of the model in depth is small compared to the depth of the object centroid, i.e. the depth of the model is approximately 1/20 of the depth of the object from the camera [Trucco and Verri, 1998]. DeMenthon and Davis claim that ortho-perspective and paraperspective projection produce lower errors for points that are off centre than when using weak perspective projection to solve the P3P problem [DeMenthon and Davis, 1992a].

Wu uses angles to compute pose, calling this the perspective angle to angle problem, and analytically solves for 3 angles i.e. the P3A problem [Wu *et al.*, 1994] as illustrated in figure 3.4(c). This is typically performed by extracting the edges corresponding to the corner of a cube or tetrahedron. Madsen studies the stability of this and a similar algorithm and states that pose estimation is inherently unstable [Madsen, 1997] for certain poses. He demonstrates that the stability of the recovered pose varies significantly with camera viewpoint, and varies in a predictable manner independent of object geometry. From all possible viewpoints, there are eight maximally stable viewpoints. For three edges meeting at a point, three corresponding planes can be defined and the surface normals calculated. The maximally stable viewpoints occur when the angle of the optical axis with each surface normal is equal.

DeMenthon, Davis and Oberkampf use four or more non-coplanar points, start with a weak perspective model and iterate towards a full perspective model to arrive at a solution [DeMenthon and Davis, 1992b; Oberkampf *et al.*, 1996]. Yuan derives an iterative solution for any number of point pairs, but says that no more than 5 point pairs are

necessary and typically 3 or 4 will suffice. Furthermore, non-coplanar data consistently outperforms coplanar data in terms of accuracy and robustness [Yuan, 1989]. Haralick provides an iterative point based technique that appears to be globally convergent and uses robust methods to overcome incorrect point matches [Haralick *et al.*, 1989].

Liu decouples the rotation and translation parameters to reduce computational cost as they are computed separately [Liu *et al.*, 1990]. In general, iterative methods minimise a mean squared error function, which can be sensitive to outliers. Trying to improve robustness, one can minimise the median of a squared error function for a set of correspondences, or rank point sets in order of the squared error and use a lower rank than median [Rosin, 1999].

Pose estimation has been performed by matching models to grey value gradients. The gradient magnitude image is compared with an image formed by taking edges from a model, projecting them onto an image plane and blurring the edge with a Gaussian function perpendicular to the edge profile. For each edge, the cost function minimises the squared difference of the gradient magnitude and Gaussian blurred edge [Kollnig and Nagel, 1997]. Pose estimation can be computed by matching the contours of an object [Ito *et al.*, 1998] to a 3D model. Ito takes a 2D contour, and minimises the distance between lines projected through 2D contour points and the closest point on the 3D model. A similar method by Lavallee [Lavallee and Szeliski, 1995] is performed using medical examples and is discussed in more detail in section 3.5.3.

Nayar proposed a reflectance based object recognition and pose estimation system [Nayar and Bolle, 1996]. For 3D objects, a range scan is acquired of an object. The object is segmented into regions and the region's reflectance ratio is computed. The reflectance ratio is invariant to illumination (light source direction, number of sources) and imaging (viewing direction, aperture setting, magnification and defocus) parameters. For a given triplet of surface patches, the reflectance ratio of each patch is used to key into a database of stored surface patches and corresponding reflectance ratios. For a test image, regions are extracted, reflectance ratios computed and a triplet of possible points selected. If the triplet exists in the database, then the pose is recovered from the image points being tested, and the stored surface points in the database, using a weak perspective pose estimation procedure. Additional points in the database can be used to verify the match.

Shekhar [Shekhar *et al.*, 1999] proposed an approach for multisensor registration based on feature consensus. With widely differing sensor types it will be difficult to design a system to match features. However, regions that appear homogeneous to one sensor are likely to appear homogeneous to another. Thus to estimate the transformation relating two images, a set of features is extracted from each image. The features must be dependent on the form of the transformation parameters. For instance if a rotation parameter is to be recovered, line orientation will provide useful information. Every combination of pairs of lines from the two images can then vote for a value of the rotation parameter. The value that receives the greatest number of votes is chosen [Shekhar *et al.*, 1999]. A similar voting procedure is shown in [Barequet and Sharir, 1997].

3.2.2 View Or Appearance Based Pose Estimation

In view based pose estimation, the aim is to deduce the pose of an object with respect to a camera by comparing a given video image with a database of 2D images. The advantage of such a method is that the object of interest does not have to be explicitly modelled, it only has to be imaged. A test image is compared with a database of stored images. The disadvantage is that the object must be available beforehand to build up the database of views, containing sufficient numbers of views to describe all possible views and illuminations. Ullman and Basri proved that using edges as a description of objects of interest, an object can be represented using a linear combination of a small number of views [Ullman and Basri, 1991]. For an object with sharp contours, imaged under orthographic projection, two images are sufficient to represent general linear transformations, and three images are required to represent rigid transformations in 3-D space. For an object with smooth contours, imaged under orthographic projection, three images are needed to represent an object undergoing linear transformations and six images for rigid transformations plus scaling [Ullman and Basri, 1991].

Another approach to view based recognition is that of using a parametric eigenspace to reduce the amount of image data that needs to be stored [Murase and Nayar, 1995a; Murase and Nayar, 1995b; McKenna and Gong, 1998]. An object is imaged under many poses and illuminations. The intensities are normalised and the region corresponding to the object of interest is identified. Each $M \times N$ image, where M is the number of rows and N is the number of columns, can be plotted in an $M \times N$ dimensional space. Each pixel in the image corresponds to one dimension of this high dimensional space,

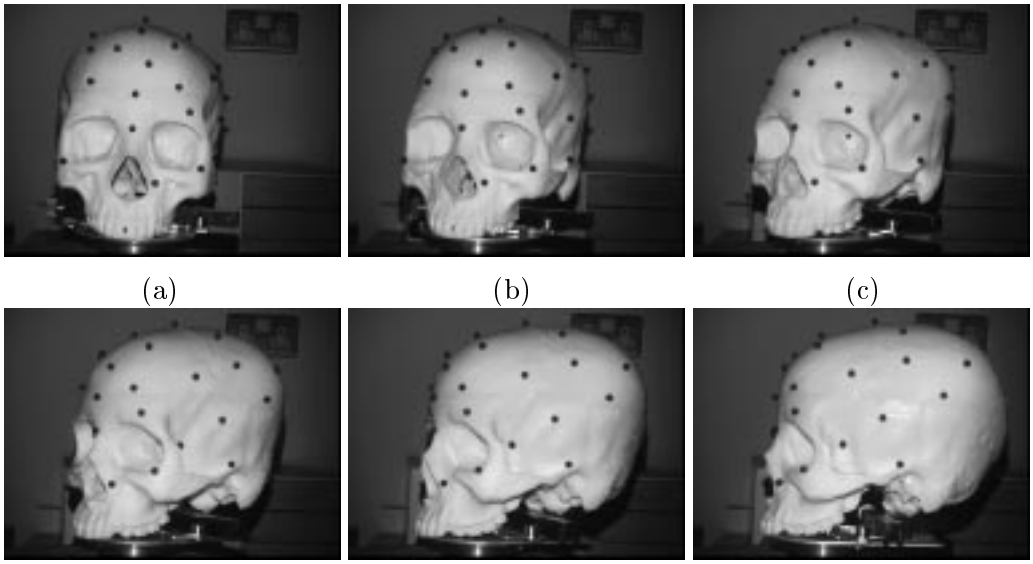


Figure 3.5: For view based recognition the object of interest must be imaged in a wide variety of known poses and illuminations. Here a skull phantom is imaged as it rotates.

and the scalar value at each pixel determines the coordinate for each dimension. Many images are taken and plotted in this high dimensional space. Images that are similar are likely to be plotted near each other. Eigenvector or principal component analysis of this space determines the most important types of variation in intensity. The most important information can be kept, and the minor modes of variation discarded, resulting in a significant amount of data compression. If training images are taken with known pose and illumination parameters, the points plotted in the high dimensional space will form a low dimensional manifold parameterised by the pose and illumination parameters. If a test image is taken, it can be recognised as being similar to the training set if it is close enough to the training points, and its pose can be determined by interpolating the closest point on the surface defined by the training points.

3.3 Tracking

Tracking is the process of computing the pose of a camera with respect to a model coordinate system, for a series of images taken over time. The computer vision literature abounds with many algorithms for the analysis of visual motion from image sequences. In addition, there are several computer vision based processes that are similar to tracking. The 'structure-from-motion' problem is to determine the shape and movement of an

object relative to a camera by tracking corresponding points in an image sequence. Very low bitrate image coding methods aim to track the movement of objects within an image sequence in order to provide high data compression for the communication industry. The structure from motion and image encoding algorithms are briefly discussed before moving on to the more typical tracking algorithms.

3.3.1 Structure From Motion

In the last 20 years, a large body of research has been performed on the 'structure-from-motion' problem. The aim is to deduce the structure and motion parameters of an object using a sequence of images [Trucco and Verri, 1998; Tomasi and Kanade, 1992].

Tomasi's method [Tomasi and Kanade, 1992] (also reviewed in Trucco and Verri's textbook [Trucco and Verri, 1998] chapter 8) assumes an orthographic projection model and that corresponding feature points have been identified. Then the structure and motion parts are separated (factorised) to enable a simple stable solution. Due to the orthographic projection, the full perspective transformation is not recovered. Furthermore, the entire image sequence is processed at once. This adds stability to the calculation, but means that the processing must be done after the images are acquired.

Optical flow, developed by Horn and Schunck in 1980, and described in [Ballard and Brown, 1982], is the apparent movement of image intensities between successive images. A tracking method described by Trucco and Verri computes structure and motion using an optical flow algorithm. This gives a dense flow field (1 flow vector per pixel) but can produce unreliable estimates of the motion field. However, it can be calculated on a frame by frame basis. The translation component is recovered using approximate motion parallax, the rotational component recovered using a least squares procedure and then the rotational component used to calculate depth and hence object coordinates [Trucco and Verri, 1998].

It will be assumed that if a tracking experiment is performed, i.e. a series of video images is taken, then the pose must be computed on an image by image basis. Therefore, the first of these methods requires a complete image sequence, which makes it inapplicable. The second method reconstructs the object in the coordinate system of the camera. For the work described in this thesis, a 3D image of the object already exists, and must be registered to the 2D optical image, i.e. the 3D structure of the object is known. Thus if the second method were performed, a registration process between camera coordinates

and 3D image coordinates would still be necessary. In summary, even though structure-from-motion algorithms do produce an estimate of how an object moves relative to a camera, these methods are an over complication of the pose estimation task, because in the case of 2D-3D registration, a 3D model already exists. Thus this class of algorithm is not considered further in this thesis, as there are more direct ways of computing the pose.

3.3.2 Model Based Image Coding

There also exists a large body of literature concerned with model-based image coding [Li *et al.*, 1994; Steinbach *et al.*, 1998]. These methods aim to segment a 2D image in terms of the objects within it, and then provide parameterisation of how the object moves over time. This is used to provide very low bitrate image coding/compression. Rather than transmit many images of similar objects, the object descriptions are transmitted and then information concerning how the object moves from frame to frame. This avoids redundant transmissions and results in high levels of compression. An example application is that of transmitting video information for video phones. Some methods [Koch, 1993] do use a 'model' of the object i.e. a wire frame model of the head and shoulders, but the emphasis is on providing a simple 2D segmentation rather than accurate pose estimation. Steinbach describes a motion analysis and segmentation algorithm of video images for model-less based image coding [Steinbach *et al.*, 1998]. The algorithm uses two successive video frames and reconstructs an unstructured dense set of points using a structure from motion algorithm. The image texture is then mapped onto the points, and used to track these points in subsequent video frames.

LaCascia also develops a method for tracking using texture mapping [LaCascia *et al.*, 1998] for possible video conferencing or image coding applications i.e. tracking faces. LaCascia however can only approximate the shape of the face using a cylinder. Again, no careful validation was performed. The first video image is aligned with the cylinder model, and then a texture map generated by unwarping the cylinder model and storing the resultant image. Subsequent video images are applied to the model using an estimate of the extrinsic camera parameters, and then the model is unwarped and compared to the previous texture map. This means that both images being registered are warped by an unrealistically simplified transformation.

3.3.3 Region Based Tracking

Another type of tracking algorithm exists, where the aim is to keep track of a specific region within a 2D image e.g. face tracking for use in a video conferencing application [Birchfield, 1997; Birchfield, 1998]. These algorithms often use a simple 2D model to parameterise the region of interest. Birchfield tracks a head in an image sequence by defining an ellipse to model the head position, and optimises the x and y coordinate of the centre of the ellipse, and the scale of the ellipse [Birchfield, 1997; Birchfield, 1998]. The aim of the tracking is to keep the ellipse encircling the head. This is achieved by moving the ellipse such that the interior of the ellipse contains image pixels whose histogram is similar to a known face distribution, and the edge of the ellipse contains the edge of the face.

Hager assumes that the region of interest can be approximated by a plane undergoing affine transformation [Hager and Belhumeur, 1998]. Hager performs frame rate tracking of the face by updating the position of a rectangular region of interest, even in the presence of illumination changes. Bascle uses a deformable contour to track outlines of humans and cars in image sequences [Bascle and Deriche, 1994] and Ivins uses conveniently colour coded regions to perform region based tracking of a mechanical arm [Ivins and Porrill, 1998]. In these region based methods, no explicit model of the 3D object or a camera projection model is used. The tracking is done purely on 2D information. Therefore these algorithms have not been considered further in this thesis.

3.3.4 Feature Based Tracking

Feature based tracking refers to the process of estimating the pose parameters by using point or line correspondences, and has much in common with the pose estimation algorithms described in section 3.2.

A good example of the overall study of pose estimation and tracking can be seen in the work of Lowe [Lowe, 1987; Lowe, 1991; Lowe, 1992]. Edges in the images are detected, and a model is matched to the edges using a Newton type least squares optimisation. False line matches are rejected by considering line proximity, parallelism and collinearity. This improves the robustness and range of capture [Lowe, 1987]. Lowe also extends this to considering models with variable internal parameters. The pose of a hand operated drill was found, where one of the parameters determined the rotation of the handle. Again a least squares minimisation is performed and the object is tracked over long

image sequences [Lowe, 1991]. Lowe subsequently improves the ability of the tracking algorithm to cope with larger frame to frame motions [Lowe, 1992]

Kalman filtering is a widely used method for tracking a small number of features over time [Trucco and Verri, 1998]. The Kalman filter estimates the position of features in a current frame, given the position, velocity and acceleration from previous frames. Lowe however, relies on the fact that in general the Newton search procedure converges very quickly.

Uenohara has described a real time system for tracking and image overlay [Uenohara and Kanade, 1995]. An initial manual alignment establishes the correspondence between a model and video image. Feature points on the model are associated with a corresponding small window in the video image. For each new frame, the 2D location of the feature point is identified using correlation based template matching, the 2D-3D correspondences are maintained using projective invariants and the pose updated using Newton's optimisation. This system enabled the real time tracking of a PC, and the overlay of an outline of a component board, and also the tracking of a human leg, with an overlaid bone outline. The system relied on dedicated vision hardware. The use of correlation matching is sensitive to window size, and window content [Burt *et al.*, 1982], which was alleviated by using the Karhunen-Loeve expansion, (similar to the eigenspace method of section 3.2.2).

3.4 A Framework For Image Registration

Registration is the determination of a mapping between coordinates in one space and coordinates in another such that points which correspond to the same physical location are mapped onto each other. Brown discussed different registration algorithms by considering four components, the feature space, search space, search strategy and similarity metric or similarity measure [Brown, 1992]. These four components are now described in detail.

3.4.1 Feature Space

The feature space represents the information which is used to align two images. This information may include points, lines or edges, surfaces or image intensity information. Unlike image intensities, points, edges, and surfaces must be extracted and explicitly described. Thus point, edge or surface based registration algorithms are called feature based and algorithms which simply read the raw image data are called intensity based.

In designing a registration algorithm, a choice of feature or features is necessary. The accuracy of the registration is determined by how accurately the features used to match actually represent corresponding points in the two images and the underlying object that was imaged. Two issues are important, the accuracy of the feature extraction and the accuracy of assigning point correspondence. The feature extraction process must be immune to image noise and artifacts. Determining point correspondence becomes exponentially more complex as the number of points increases. If the extraction process results in incorrect point matches, the registration algorithm must be designed to be immune to noisy point correspondences.

3.4.2 Search Space

The search space is the space of all transformations that can be used to align the images. This can include translations and rotations (rigid body transformations), scaling and shearing or warping. As the complexity of the transformation increases, so does the number of degrees of freedom to describe it and the number of possible transformations increases rapidly. Consequently the search space should be restricted to those transformations necessary to provide the desired accuracy and match the expected deformation.

3.4.3 Similarity Measure

In the context of image registration, a similarity measure computes the level of similarity of the two images for a given transformation. If points from two images are being aligned, the similarity measure could be the mean distance between corresponding points and therefore the lower the distance measured, the better alignment. The similarity measure can be maximised (or minimised) at alignment. The important characteristics of a similarity measure are that the maximum (or minimum) is found exactly at the correct alignment, as the transformation approaches alignment, the function monotonically increases (or decreases) towards the solution and that the measure does not take prohibitively long to calculate. The similarity measure determines the smoothness of the parameter space and a good similarity measure will have preferably only one global maximum and few, preferably zero, local minima (see section 2.4 for 'parameter space').

3.4.4 Search Strategy

The search space will contain many possible transformations, and the most accurately aligning transformation must be determined. The search strategy is the process by which the best alignment is found. Some registration problems enable a direct analytic solution to be calculated. For example given a set of n points in a 3D image, and n corresponding points in another 3D image, where $n \geq 3$, then several analytic solutions have been formulated to calculate the optimum rigid body transformation to align the two sets of points [Eggert *et al.*, 1997]. If an analytic solution is unavailable, numerical optimisation techniques must be used. A brute force algorithm tries every possible transformation in the search space, typically in a systematic fashion and is therefore classified as direct.

The properties of the search space determines the search strategy required. If the search space is perfectly smooth, then even a simple search strategy will find the solution. If the search space is jagged a more complicated search will be necessary. The range and level of detail of the search strategy should be appropriate to the object of interest, the image type and its resolution.

3.5 Medical Image 2D-3D Registration

A large body of work has been performed in medical image registration. However, this usually involves either 3D-3D volume registration of MR, CT, PET, single photon emission tomography (SPECT) to name but a few, or image to physical space registration for image guided surgery. Comprehensive reviews of image registration have been provided by Brown [Brown, 1992], medical image registration [van den Elsen *et al.*, 1993; Maintz, 1996; Little and Hawkes, 1997] and registration for image guided surgery [Lavallee, 1996]. The purpose of this review is to investigate 2D-3D medical image registration, with an emphasis on algorithms that match optical images to 3D images such as MR or CT scans.

3.5.1 Point Based Algorithms

With computer vision pose estimation algorithms, the ability to track fiducials enables a quick and efficient solution for the extrinsic parameters. In medical imaging, it is difficult to design markers that are easily localised in both the 2D and 3D images whilst also being accurate and non-invasive. However, one solution, proposed by Edwards *et al.* [Edwards *et al.*, 1999b] is to use markers that can be tracked using dedicated hardware rather than deducing the position from the 2D image. The 3D image and the video cameras are registered to the tracking device. Therefore the transformation from 3D image, to tracker, to video camera, to 2D image coordinates is achieved via an intermediate tracking device rather than by image processing. This is described in further detail below.

3.5.1.1 Edwards *et al.*

Edwards *et al.* use a tracking system to register MR/CT data to video views for the purpose of producing graphical overlays in the stereo operating microscope [Edwards *et al.*, 1999c; Edwards *et al.*, 1999b]. A patient has a custom fit, lockable acrylic dental stent (LADS) fitted to their upper teeth. The LADS device is a teeth clamp, with a set of MR/CT visible markers, localisation caps, or infra-red LED's attached. With the MR/CT visible markers, the patient is scanned. The markers are accurately located in the 3D image [Wang *et al.*, 1997]. The imaging markers are removed and replaced with localisation divots, which enable the position of these divots to be accurately recorded using a pointer tracked with an optical tracking device. By localising the position of the LADS markers in the operating room and in the pre-operative images, the world and MR/CT coordinate systems can be registered using an SVD algorithm [Arun *et al.*,

1987]. In addition, the video coordinate system can be registered to the world coordinate system using Tsai's camera calibration [Tsai, 1987] or an SVD technique [Trucco and Verri, 1998]. The microscope which contains the video camera also has LED's attached so that if the microscope moves, the registration transformation can be updated. The video camera is calibrated repeatedly for various zoom and focus settings. Once registration is achieved, renderings of surface models are overlaid into each eyepiece using stereo injection units. This provides a stereo augmented reality.

Edwards *et al.* report registration accuracy of $0.3-0.5$ mm on phantoms, and $0.5-4$mm on target structures during 3 operations [Edwards *et al.*, 1999b]. This system represents the first of its kind to provide accurate overlays in the operating microscope using stereo graphics for enhanced depth perception, and with an accuracy of around 1mm.

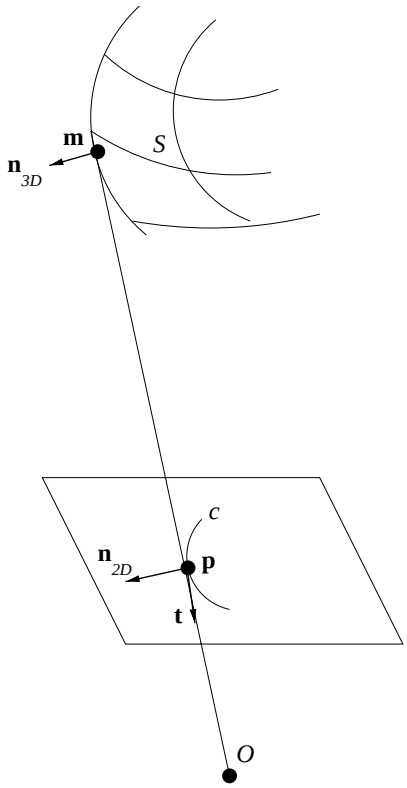
3.5.2 Contour Based Algorithms

As mentioned in the previous section, corresponding points or landmarks are difficult to extract from both a 2D and 3D medical image. However, the extraction of an external surface or contour is in some cases feasible. The silhouette of an object can be used as a strong cue for object recognition or shape reconstruction. In addition, if a 3D model of an object exists, then the external contour of the object in a 2D image can be matched to the surface of the 3D model. For this to be possible, information must either be projected from 2D to 3D or vice versa. Two such algorithms are now described.

3.5.2.1 Betting And Feldmar *et al.*

The method proposed by Betting and Feldmar in [Feldmar *et al.*, 1997] is a method for registering 3D images to either video or X-ray images. The method matches a projection of the 3D object to a contour in the 2D image. The algorithm is used to calculate the extrinsic parameters relating a CT scan of a mannequin head to a video image, and also the extrinsic and intrinsic parameters relating a CT scan of a skull to a radiograph of that skull.

Betting and Feldmar define the fundamental property of an occluding contour as: If a point $\mathbf{m}$ on a surface S is such that projection point $\mathbf{p}$ lies on the occluding contour c then the normal vector $\mathbf{n}_{3D}$ to S at point $\mathbf{m}$ is equal to the normal vector $\mathbf{n}_{2D}$ of the plane P defined by $(\mathbf{p}, 0, \mathbf{t})$ where $\mathbf{t}$ is the tangent vector to the occluding point contour at point $\mathbf{p}$. This is illustrated in figure 3.6. Thus at alignment, points on the 3D silhouette should



(a)

Figure 3.6: The fundamental property of the occluding contour. (Reproduced from [Feldmar *et al.*, 1997], with slightly different notation. See text, section 3.5.2.1.

project onto the 2D occluding contour, and have the same projected normal. With the assumption that the intrinsic camera parameters are known, to calculate an initial match, only three pairs of corresponding 3D and 2D points are required. Three 2D points are picked so that two lie on a tangent line, and the angle between the normals of the first and third is as close as possible to π. For every triplet of 3D points which is similarly configured, the projective transformation is calculated until a sufficient match is found. Given this initial transformation a modified iterative closest point (ICP) algorithm [Besl and McKay, 1992] refines the registration. Using the current estimate of the extrinsic parameters, a function $Match$ assigns correspondence between the 2D contour points, and the 3D surface points. A distance measure measures the Euclidean distance between 3D points projected onto the 2D image plane, and also the difference between the surface normals. This is minimised by alternately (a) keeping $Match$ fixed and updating the extrinsic parameters, and then (b) keeping the extrinsic parameters fixed and updating match.

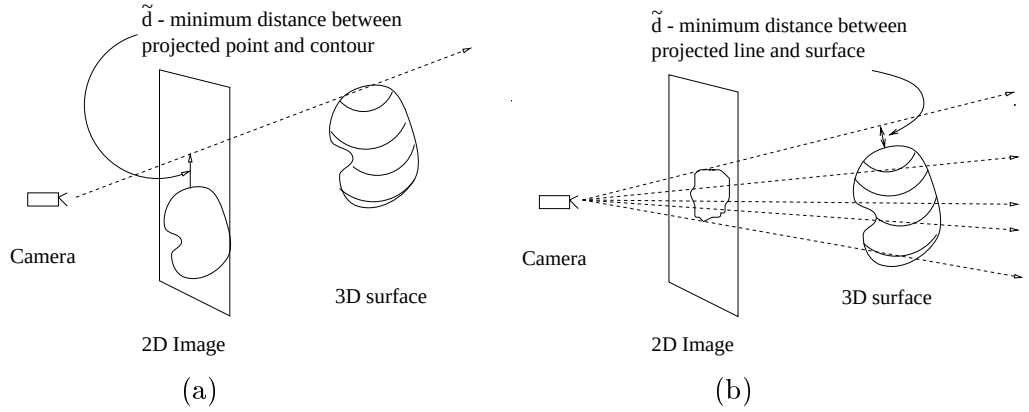


Figure 3.7: Illustration Of (a) Betting and Feldmar's method and (b) Lavallee and Szeliski's method

This algorithm was used to register a CT scan of a mannequin head to a video image by adjusting the extrinsic parameters. The algorithm took 10 seconds on a Dec alpha workstation. The initialisation process gives an average error of 2.1 pixels for the spatial distance, and 2 degrees for the difference in 2D and 3D normal angles, and after minimisation, this reduces to 0.76 pixels, and 0.17 degrees. The algorithm was also used to register a CT scan of a skull and a radiograph by adjusting the extrinsic *and intrinsic* parameters. The initialisation process gives an average error of 1.5 pixels and 1.9 degrees which is reduced to 0.79 pixels and 0.7 degrees.

3.5.2.2 Lavallee And Szeliski

The work by Lavallee and Szeliski [Lavallee and Szeliski, 1995], is similar to that of Betting and Feldmar, in that it is also a silhouette based technique. The difference is that Betting and Feldmar's technique projects the shape of the 3D object onto the 2D image, and minimises a 2D based distance function and the distance between the projected surface and contour normals, whereas the algorithm presented by Lavallee and Szeliski projects lines from the 2D points, and minimises the distance between these lines, and the 3D object's surface, which is a 3D distance function (see figure 3.7).

The video camera is calibrated using the N-Planes Bicubic Spline (NPBS) method [Champleboux *et al.*, 1992]. The 2D occluding contour and 3D surface are segmented. From the CT volume, they create a modification of an octree which they call an octree spline. This enables a fast computation of the distance of a point to a surface. The distance of a line to a surface is defined as the minimum distance between the points on that line, and the surface. The similarity measure is defined by the sum of the squared distances

between all the projected lines, and the surface. Minimisation is performed using the Levenberg-Marquardt algorithm, which is a non-linear iterative method. The minimisation ends when the energy function is below a fixed threshold, or when the normal of the gradient of the energy function is below a threshold or when a maximum number of iterations is reached.

The algorithm was tested on real and synthetic data. The most notable experiment consisted of an isolated vertebra with two tubular 3mm holes. This was CT scanned and the position of the two holes were calculated. The 3D surface was then segmented from the CT scan, giving 200,000 points, which was used to build a six-level octree spline. The vertebra was placed in the field of view of two cameras, and then an edge extraction algorithm was used to extract between 10 and 200 contour points on each image. The algorithm was then applied. They used a laser beam attached to a robot arm, calibrated with the cameras, such that once registered, the laser beam should be aligned with the drilled holes. The authors report that the alignment was visually perfect and performed in 1 - 4 seconds. They also highlight that least squares minimisation could be prone to local minima, but they have not found this to be the case.

3.5.3 Surface Based Algorithms

Surface based algorithms actually perform all the registration in 3D. To do this, either a surface representation must be deduced from the video image or some other device such as a laser range finder. If a laser range finder is used it must be calibrated with respect to the video images. i.e. the transformation between 3D laser coordinates and video image coordinates must be known. Once a surface representing that present in the video image has been reconstructed, it must be registered to the surface extracted from the 3D image. Three such algorithms are now described in detail.

3.5.3.1 Grimson *et al.*

The work done by Grimson *et al.* [Grimson *et al.*, 1995; Grimson *et al.*, 1996] uses a laser range finder in conjunction with a single video image to deduce the surface visible in the video image. The laser range finder can reconstruct a surface accurate to 0.08 mm [Grimson *et al.*, 1996]. This is registered to a surface derived from MR/CT. The reconstructed surface is first manually edited and the two surfaces manually aligned. If $\mathbf{l}_i$ is a vector representing a laser point and $\mathbf{m}_j$ is a vector representing a model point,

with $\mathbf{T}$, a matrix representing the transformation, then the evaluated function is:

$$E_1(\mathbf{T}) = -\sum_i \sum_j \exp^{-(|\mathbf{T}\,\mathbf{l}_i - \mathbf{m}_j|^2/2\sigma^2)} \tag{3.3}$$

which due to its inverse exponential nature gives a smooth cost function. The search strategy was the Davidon-Fletcher-Powell quasi Newton method (DFP) described in [Press *et al.*, 1992]. This is a gradient based search requiring calculations of derivatives. The Gaussian function enables a multiresolution approach, by changing the variance. The resultant pose is refined using a least squares measure of the form:

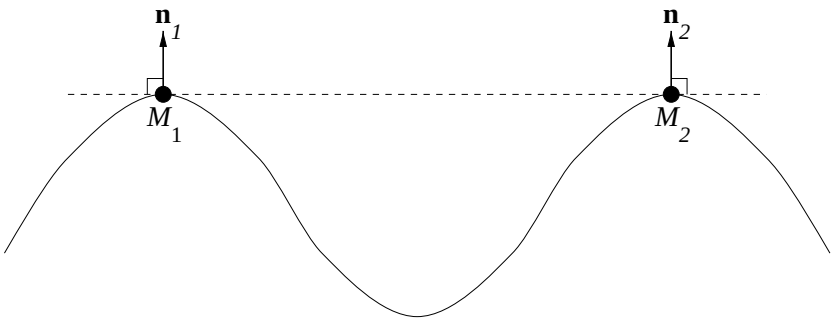
$$E_2(\mathbf{T}) = \left[\frac{1}{n}\sum_i min\left\{d^2_{max}, min_j|\mathbf{T}\,\mathbf{l}_i - \mathbf{m}_j|^2\right\}\right]^{\frac{1}{2}} \tag{3.4}$$

where n is the number of points and d^2_{max} is a maximum distance threshold. This is more accurate but prone to local minima. The final pose is perturbed randomly, and re-registered to further avoid local minima.

Using a 0.9375 $\times$ 0.9375 $\times$ 1.4mm resolution MR image, results in an RMS error of the order of 1.5mm. The algorithm takes 2-4 minutes for the laser scanning and the alignment, and the capture range is of the order of 5mm or degrees, with 100% success, reducing to 70% success at 10mm or degrees perturbation. This measure of success was obtained by randomly perturbing the start point from a fixed point, and seeing how many times the algorithm reconverged to the start point. The threshold for successful was an RMS value of 2.5mm. The experiment ran 10 tests, at each of 1 - 10 mm or degrees misregistrations. More recent clinical work tests this system in 70 patient cases [Grimson *et al.*, 1998].

3.5.3.2 Betting And Feldmar *et al.*

Betting and Feldmar also propose a surface based video - MR registration algorithm [Feldmar *et al.*, 1997]. Two video images are taken, and a surface reconstruction leads to a dense set of points and normals. A second surface is extracted from an MR scan. The algorithm starts by computing pairs of bitangent points (see figure 3.8). Two points are bitangent if the plane defined by each point and its normal are the same. The initial estimate is performed as follows. Two points on one surface are taken, and the distance between them calculated. Then all pairs of similarly separated points in the second surface are tested. The two transformations to align the four points are calculated, and this process repeated until a suitable transformation found. Obviously, in practice,



(a)

Figure 3.8: Two points are bitangent if the plane defined by each point and its surface normal is the same.

the algorithm takes into account that pairs of points will not be exactly superimposed due to discretisation noise, and also points in one surface may not correspond to points in another. The iterative match is a modified ICP algorithm where, instead of a 3D distance function, a 6D distance function computes the Euclidean distance between each corresponding point and their surface normals. The algorithm finds 5000 bitangent point pairs on the MRI surface, and 598 pairs on the stereo surface, in about 30 seconds. The initial estimate of the registration takes 30 seconds. At this point 80% of the points on the stereo surface have a closest point within 8mm, compared with the voxel dimensions of 4mm $\times$ 4mm $\times$ 8mm. The modified ICP is then applied using 15000 points on the MRI and 10000 on the stereo surface. This takes 20 seconds, whereupon 85% of the stereo points have a closest point less than 3mm away. The registration is visually accurate, allowing pre-operative data to be overlayed on the video.

3.5.3.3 Colchester *et al.*

Colchester *et al.* [Colchester *et al.*, 1994; Henri *et al.*, 1995; Colchester *et al.*, 1996] also use a video based surface reconstruction which is matched to a pre-operative surface extracted from an MRI/CT scan. The video surface is reconstructed by projecting a series of stripes onto the object. Two video cameras capture an image of the patient, and detect the stripes. Corresponding points in each view are matched and then surface points reconstructed using triangulation. The surfaces are registered using a cost function of the form

$$C = \sum_{i=1}^{N} log(1 + \frac{1}{2} d_i^2) \tag{3.5}$$

where i is a point number, N the total number of points and d_i is the distance between a point on the reconstructed surface and the corresponding closest point on the MR surface. The distance is found with a distance map, and if a distance is > 10mm the point is ignored. This technique coupled with the log function deals well with outliers, and incorrect matches. A simple optimisation procedure, incorporating multiple start points and multi-resolution step sizes is employed to minimise the cost function and hence register the surfaces. The surface reconstruction takes 30 seconds, on a SUN SPARC IPX, and has an accuracy of 0.5mm in 3D. The MR data set was $256 \times 256 \times 80$ slices with $0.94 \times 0.94 \times 2.0$mm voxel dimensions. The algorithm registers reliably with real video data, coping with misregistrations of $\pm$ 10mm and $\pm\ 20^\circ$, resulting in a mean surface separation of 0.4 ($\pm$ 0.5)mm, and a maximum surface separation of 2.2mm, which was visually inspected, and 'no deviations were apparent between the two surfaces'. If the surfaces were fully overlapped, then the *log* based function performed worse than a least squares, but better if the two surfaces were different, and not fully overlapping at alignment. The registration process took 153 seconds on a HP 9000/715 (50 Mhz) workstation. It was not used to track the patient or cameras.

3.5.4 Intensity Based Algorithms

In 3D-3D medical image registration, intensity based methods have proven very popular and successful [West *et al.*, 1997]. The similarity measure is based on the underlying image intensities alone. Inspired by the work of Woods, registering PET-PET images [Woods *et al.*, 1992] and then PET-MR images [Woods *et al.*, 1993], interest rapidly grew. Van den Elsen *et al.* used a correlation technique to match MR-CT [van den Elsen *et al.*, 1994], and Hill *et al.* used a measure of dispersion of the corresponding image intensities in a grey level histogram [Hill *et al.*, 1994]. The work of Hill led directly to Viola and Wells [Viola and Wells, 1995], and Collignon and Maes [Collignon *et al.*, 1995] independently proposing the use of mutual information as a similarity measure. Since then, the use of mutual information (MI) and subsequently normalised mutual information (NMI) [Studholme *et al.*, 1999] has grown significantly, being applied to many different image modalities and applications, with performance superior to 3D-3D feature based techniques [West *et al.*, 1997; West *et al.*, 1999]. Intensity based methods have been used for 2D-3D medical image registration. Before looking at specific work, some intensity based similarity measures will be described in a general context.

3.5.4.1 Intensity Based Similarity Measures

Penney reviews similarity measures for 2D-3D radiograph to CT registration [Penney *et al.*, 1998; Penney, 1999]. Of particular interest for later in this thesis are the similarity measures normalised cross correlation (NCC), gradient correlation (GC), joint entropy (JE), mutual information (MI), and normalised mutual information (NMI). These are outlined below.

Normalised Cross Correlation

Let two images be denoted by V, R, and let $v(x, y)$ and $r(x, y)$ denote the intensity value at location (x, y) for each image. The normalised cross correlation (NCC) of image V and R is defined as

$$\mathrm{NCC}(V, R) = \frac{\sum_{x,y}(v(x, y) - \overline{v})(r(x, y) - \overline{r})}{\sqrt{\sum_{x,y}(v(x, y) - \overline{v})^2}\sqrt{\sum_{x,y}(r(x, y) - \overline{r})^2}} \tag{3.6}$$

where $\overline{v}$ and $\overline{r}$ denote the mean intensity value in images V and R respectively. Normalised cross correlation assumes a linear relationship between intensities in one image, and the corresponding intensities in the other. It is a measure of how the corresponding intensities fit to a straight line. The intensity value itself is used so a few large differences in intensity may have a significant effect on the similarity [Penney *et al.*, 1998].

Gradient Correlation

Each image R and V is convolved with vertical and horizontal Sobel edge filters [Gonzalez and Woods, 1992] to approximate derivatives. This yields images V^h, V^v, R^h and R^v where superscripts v and h refer to images after convolving with vertical and horizontal Sobel edge filters respectively. The gradient correlation (GC) is defined as

$$\mathrm{GC} = \frac{\mathrm{NCC}(V^v, R^v) + \mathrm{NCC}(V^h, R^h)}{2} \tag{3.7}$$

i.e. the mean average of the NCC of the vertical and horizontally convolved images. Gradient measures concentrate on edge information, filtering out low frequency differences in the images. However, the correlation based calculations will still be affected by large differences in intensity between corresponding pixels [Penney *et al.*, 1998].

Joint Entropy

Joint entropy is a measure of the amount of shared information in two sets of symbols [Reza, 1961]. Let v be an intensity value in image V and likewise r be an intensity value in image r. Let $p(v)$ denote the probability of intensity v in image V, $p(r)$ be the probability of intensity r in image R and $p(v,r)$ the joint probability of intensity v and r occurring at corresponding pixel locations in images V and R respectively. Let $\mathcal{V}$ be a random variable denoting the distribution of intensity values in image V and likewise $\mathcal{R}$ be a random variable denoting the distribution of intensity values in image R. The marginal entropy H of each random variable $\mathcal{V}$ and $\mathcal{R}$ denoted using $H(\mathcal{V})$ and $H(\mathcal{R})$ is defined as

$$H(\mathcal{V}) = -\sum_v p(v)\log p(v) \qquad H(\mathcal{R}) = -\sum_r p(r)\log p(r) \tag{3.8}$$

and the joint entropy $H(\mathcal{V},\mathcal{R})$ is defined as

$$H(\mathcal{V},\mathcal{R}) = -\sum_v \sum_r p(v,r)\log p(v,r) \tag{3.9}$$

An image with constant intensity contains minimum entropy or information. The joint entropy of two images measures the combined entropy of the two images. If the two images are identical, the joint entropy equals the marginal entropy. If the images differ, the combined image will contain more information. As entropy based measures are calculated from probability distributions, they make no assumptions about the absolute value of an intensity. Joint entropy assumes that as alignment is reached the probability of co-occurrence of pixel intensity pairs should be maximised. Thus, these measures should be robust to large differences in a small number of pixels i.e. outliers.

Mutual Information

From the definitions of entropy above, mutual information is simply defined as

$$I(\mathcal{V};\mathcal{R}) = H(\mathcal{V}) + H(\mathcal{R}) - H(\mathcal{V},\mathcal{R}) \tag{3.10}$$

Mutual information is also based on probability distributions. Whilst mutual information has been found very successful for volume registration [West *et al.*, 1997], Penney found it to be inadequate for 2D-3D registration of radiographs to DRR images formed from a CT scan [Penney *et al.*, 1998].

Normalised Mutual Information

Studholme proposed normalised mutual information which is defined as

$$Y(\mathcal{V};\mathcal{R}) = \frac{H(\mathcal{V}) + H(\mathcal{R})}{H(\mathcal{V},\mathcal{R})} \tag{3.11}$$

Joint entropy and mutual information can only be evaluated for corresponding pairs of pixels in two images, and hence are likely to be affected by the exact number of pixel pairs used, which is determined by the volume of overlap of the two images when evaluating the measures. Normalised mutual information is more invariant to changes in the volume of overlap than mutual information for 3D to 3D image registration [Studholme *et al.*, 1999].

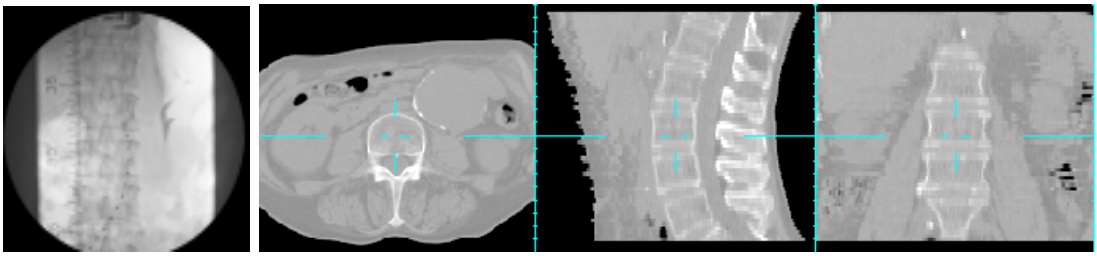
Having described those intensity based similarity measures previously used in the medical imaging literature, attention will now be drawn to specific algorithms. To date, for 2D-3D medical image registration, intensity based methods have mainly been used on images of X-ray attenuation such as radiographs, fluoroscopy and portal images registered to CT images. Intensity based registration for optical images to a 3D model has been proposed by Viola and Wells [Viola and Wells, 1995; Viola and Wells III, 1997], but so far has not been widely used. Although attenuation images such as radiographs are very different in content to video images, several radiograph - CT image registration algorithms are described below. For a more complete review see [Penney, 1999].

3.5.4.2 Lemieux *et al*

Registration of radiographs and CT images, using image intensities alone, was pioneered by Lemieux [Lemieux *et al.*, 1994]. Lemieux took an anterioposterior and lateral radiograph of a plastic skull phantom, and registered them to a CT scan of the same skull phantom. The transformation relating the two radiographs to world coordinates was known, and the projection parameters of the X-ray source were calibrated. The transformation calculated was the CT to world coordinate transformation involving three translations and three rotations. The registration algorithm was an iterative procedure, based around the production of Digitally Rendered Radiographs (DRRs). For a given transformation, a virtual X-ray source was defined relative to the CT volume, with the same projection parameters as the real x-ray source. Virtual X-rays were then projected through the CT volume and summed to produce a simulated radiograph, i.e. a DRR. An interface displaying the two real radiographs, and for a given pose, the two DRR's,

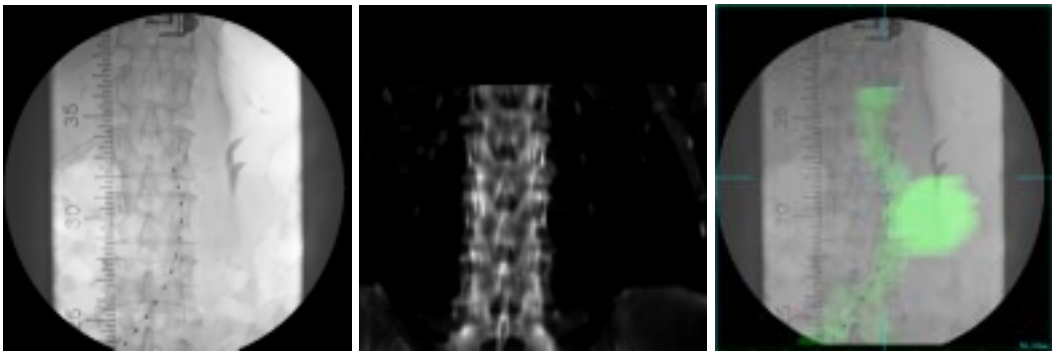
allows the user to manually align the CT, via the radiographs, to within one centimetre from registration within one minute. An eight stage orientation initialisation process then uses Brent's line minimisation [Press *et al.*, 1992] to adjust two of the three rotations with respect to two views, and two resolutions. The minimisation is performed by selecting a value for the rotation parameter, producing the DRR, and comparing it to the corresponding radiograph using a correlation based measure. Then three passes of Powell's minimisation [Press *et al.*, 1992], one at 1/4 resolution and two at 1/2 resolution were used to refine the cost function using a gradient based measure. The gradient based measure was found to be more accurate near the optima than the correlation function. The correlation measure was found to be a smoother function when further away from the optima when compared with the gradient based measure. Lemieux performed 3 sets of 100 registrations with different offsets from the stereotactically correct gold standard solution, with mean final registration errors of 0.52 -2.76 mm with success rates of 92 - 99% for the three sets.

Subsequently other authors have investigated the use of registration using DRRs. Whereas Lemieux produced DRRs by simply summing the intensity values along a line of projection in the CT, Brown studies the actual relationship between real radiograph intensities, and the CT values being composited, with the aim of improving registration accuracy [Brown and Boult, 1996]. Brown registers a single view radiograph to a CT volume of a femur where, on simulations, results are only good for small misregistrations < 4 mm or degrees from the correct solution, and for real images, the results were mixed even for misregistrations that were only 2mm or degrees from the correct solution. Murphy demonstrates a system for potential use in a radiotherapy treatment room for head images [Murphy, 1997]. Murphy also uses DRRs, but computes small regions of interest around the external skull contour in the radiograph. When producing the DRRs, only rays that correspond to the regions of interest are actually computed, thereby saving time. With small initial misregistrations, $\pm 1 - 3$mm or degrees, registration was performed in two seconds, with an accuracy of 0.7mm.



(a) (b)

Figure 3.9: Corresponding (a) fluoroscopy and (b) three orthogonal views of a CT image. See text section 3.5.4.3.



(a) (b) (c)

Figure 3.10: Registration of fluoroscopy to a CT image using an intensity based algorithm [Weese *et al.*, 1997b; Penney *et al.*, 1998]. See text section 3.5.4.3.

3.5.4.3 Weese, Penney *et al*

Weese and Penney [Weese *et al.*, 1997a; Weese *et al.*, 1997b; Penney *et al.*, 1998] have developed a single view DRR based algorithm for registering spine fluoroscopy and CT images. Image 3.9(a) shows a fluoroscopy image of a patient's spine, taken from an aortic stenting procedure. Image 3.9(b) shows three orthogonal slices through the patient's CT scan. It is generally difficult to relate the 2D image (a) with the 3D image (b) without registration. Figure 3.10 illustrates the intensity based registration of fluoroscopy to CT images. The image in figure 3.10(a) shows an example fluoroscopy image, with (b) the DRR. The DRR is compared with the fluoroscopy image using a similarity measure. The DRR is pose dependent, and thus the pose is altered until the best matching DRR is found. Once registered, the aorta from the CT image can be rendered and overlaid on the fluoroscopy image (c), or points in the CT image can be identified and the corresponding 2D point calculated. This provides a mechanism to link the two different modalities.

Penney compares the similarity measures normalised cross correlation, entropy (of the subtracted fluoroscopy and DRR image), mutual information, gradient correlation, as described in section 3.5.4.1, and also pattern intensity (PI) and gradient difference (GD). Let S be the scaled subtraction image of the fluoroscopy and DRR, and $s(x, y)$ represent the intensity value at pixel location (x, y) in image S. Pattern intensity can be written as

$$\mathrm{PI}_{r,\sigma} = \sum_{x,y} \sum_{d^2 \leq r^2} \frac{\sigma^2}{\sigma^2 + (s(x,y) - s(i,j))^2} \tag{3.12}$$

$$d^2 = (x - i)^2 + (y - j)^2 \tag{3.13}$$

where σ is an arbitrary constant used to adjust the measure's sensitivity to noise. If the fluoroscopy image and DRR are differentiated with vertical and horizontal Sobel edge filters, then let $v^v(x, y)$ and $v^h(x, y)$ denote the intensity value at location (x, y) in the vertical and horizontally differentiated fluoroscopy image respectively, and $r^v(x, y)$ and $r^h(x, y)$ denote the intensity value at location (x, y) in the vertical and horizontally differentiated DRR image respectively. Gradient difference is then calculated as

$$\mathrm{GD} = \sum_{x,y} \frac{A_v}{A_v + (v^v(x,y) - A_s(r^v(x,y)))^2} + \sum_{x,y} \frac{A_h}{A_h + (v^h(x,y) - A_s(r^h(x,y)))^2} \tag{3.14}$$

where A_v and A_h are constants calculated from the variance of the fluoroscopy vertical and horizontal gradient images and A_s is a scale factor. For registration of fluoroscopy to CT images of a skull spine phantom and also for simulated clinical images with obstructions such as soft tissue structures and an interventional stent the similarity measures PI and GD performed accurately and with a 100% success rate. The RMS error values for the recovered extrinsic parameters was < 1 for all degrees of freedom except that along the projection axis of the X-rays.

Weese uses the pattern intensity measure and studies the use of shear warp factorisation as a method of fast volume rendering to produce the DRRs, and lookup tables for fast computation of PI to reduce the registration time to < 4 seconds [Weese *et al.*, 1999].

3.5.4.4 Viola And Wells *et al*

The first image intensity based optical to medical image registration algorithm is that of Viola. He used mutual information to align a 3D surface model of a skull phantom with a single video image. Mutual information has been described in section 3.5.4.1 In Viola's example the mutual information between video image intensities and surface normals is

calculated. Equation 3.10 can easily be expanded to incorporate vector quantities like surface normals.

A surface model of a skull phantom is extracted from a CT scan. The surface model contains between 7000 and 65000 points. For each iteration, the model points are projected, using z-buffering (every 300 iterations), to find which image points they correspond to. The derivative of mutual information is calculated. This is done using Parzen Window density estimation [Duda and Hart, 1973], which relies on taking a small number of available points, and estimating the probability densities, and the derivative of mutual information. Using the Parzen Window as a probability density estimator assumes the data is continuous, whereas other authors [Collignon *et al.*, 95; Studholme *et al.*, 1999] use binning of intensity values into a discrete histogram. A stochastic gradient based search is performed, with the step size, or update parameter decreasing as convergence is achieved. In Viola's thesis [Viola, 1995], and the paper [Viola and Wells, 1995] the results show that the algorithm is capable of recovering from misregistrations of $\pm$10mm or degrees, with 50 successful tests out of 50, with a final standard deviation of 0.61mm for x translation, 0.53mm for y translation and 5.49mm for z translation, and 3.22° for the rotation. With $\pm$20mm or degrees, this degrades to 1.11mm, 0.41mm, 9.81mm and 3.31° for the same parameters. This registration took 35 seconds on a Sun Sparc Station 5. The authors claim this could be dramatically speeded up with the use of a digital signal processor, which would perform very fast random memory access. As in all 2D-3D registration algorithms, the z translation, which is a change in depth relative to the camera, is the least constrained.

Hata described a performance enhancement of Viola's algorithm based around an OpenGL implementation, utilising graphics hardware [Hata *et al.*, 1996]. The original Parzen Window [Duda and Hart, 1973] method of estimating probability distributions was replaced with a discrete histogram approach. Leventon extended this method to include stereo views [Leventon *et al.*, 1997], testing the algorithm on images of a model car. The idea of using multiple views is investigated in chapter 5.

3.6 Comparison Of Algorithms

In this section, the algorithms are compared. The aim is to determine what algorithms from the literature are to be used and where the work of this thesis should seek to contribute.

3.6.1 Camera Calibration

This chapter has reviewed several camera calibration techniques which determine the intrinsic and extrinsic parameters, only the intrinsic parameters, or a subset of the intrinsic parameters. Each paper claims good results. Furthermore, all methods generally rely on accurately extracted points and/or lines.

For a full intrinsic and extrinsic calibration, at least 6 pairs of corresponding 3D and 2D points are required. Therefore if 6 pairs of points could be accurately extracted, then registration would be feasible. Unfortunately, its is difficult to accurately identify human anatomical landmarks automatically. In addition, intrinsic and extrinsic parameters are known to be closely coupled, i.e. inaccurately deduced intrinsic parameters lead to inaccurate extrinsic parameters. Furthermore, most algorithms recommend using many points for calibration e.g. 60 [Tsai, 1987]. This is impractical if not impossible. Therefore an initial strategy would be to use a calibration procedure to determine the intrinsic parameters. This can be done before any registration task, using existing software and an accurately machined calibration object.

Tsai defines the 'radius of ambiguity zone' as an error measure. For a given 2D calibration point, a line is projected into 3D space, and the distance to the corresponding 3D point is measured in the plane of the test object. Tsai calibrates a Fairchild CCD 3000 camera with Fuji 25mm lens. Tsai reports an average error of 0.0178 mm and a maximum of 0.0331mm using a single set of coplanar points. He then uses a second camera to provide stereo reconstruction through triangulation, calibrates both cameras using multiple sets of coplanar points, and measures the error as 0.0198 mm. The computational time is 9 seconds for the latter case.

Weng uses different error measures, but reports an accuracy of 0.437 mm for reconstructed 3D points using two Cosmicar 25 mm tele-lenses. This is higher than Tsai, but cannot be directly compared as Tsai's and Weng's experimental setup are different. Computing the calibration matrix and then extracting parameters has been performed

by Strat [Strat, 1984], Ganapathy [Ganapathy, 1984] and Faugeras [Faugeras, 1993], but they don't evaluate the accuracy of the evaluation in terms of metrics like 'radius of ambiguity zone' or the accuracy with which 3D points can be reconstructed. Faugeras [Faugeras, 1993] and Robert [Robert, 1996], both demonstrate the variation in recovered camera parameters when noise is added, but this depends on experimental setup and does not indicate an absolute measure. King *et al.* [King *et al.*, 1999] use Tsai's camera calibration to calibrate a fixed zoom and focus operating microscope with an accuracy of 0.26 mm at the focal plane and 0.3-0.4 mm for a variable zoom and focus calibration.

In summary, a full point based registration will be extremely difficult to do. However, the intrinsic parameters can be retrieved through calibration using existing methods. Tsai's method is widely cited, often used as a benchmark and freely available at `http://www.cs.cmu.edu/~rgw/`. Therefore, in general, Tsai's method will be used throughout this thesis.

3.6.2 Pose Estimation

If the intrinsic parameters are recovered through calibration, then the registration problem reduces to one of pose estimation, i.e. estimating the extrinsic parameters. However, the problem of pose estimation in computer vision is markedly different from the medical registration problem considered in this thesis. Many of the published pose estimation algorithms rely on being able to extract points [Fishler and Bolles, 1981; Wolfe *et al.*, 1991; Haralick *et al.*, 1994; DeMenthon and Davis, 1992b], points and lines [Liu *et al.*, 1990; Phong *et al.*, 1995] or angles [Wu *et al.*, 1994]. For this type of problem, pose estimation is widely studied. However, for the same reason as above, such easily identifiable points, lines or features are unlikely to be present in a medical scene.

View based pose estimation is based on comparing a test image against a database of pre-stored images. Although, this does not require an explicit 3D model and hence will not require feature extraction, it does require that the subject of interest be available before analysis takes place. These methods have practical limitations in terms of memory/disk usage, pre-processing time and feasible accuracy. Sufficient images must be captured to describe the likely poses and illuminations. This could involve many hundreds of images and many gigabytes of disk space. Next, principle component (eigenvector) analysis must be performed, which does fortunately reduce required disk space. When a test image is acquired it is compared with those in the database. Results show that pose estimation

may be accurate to 0.5 - 1.0 degrees [Murase and Nayar, 1995b], but in this paper, these methods only estimated one pose parameter, a rotation. In principle these methods could be extended to recover all six extrinsic parameters, but at increased inconvenience in data collection and storage, and increased computational cost.

These methods were classified as unsuitable due to the difficulty of collecting enough images to describe all possible poses and illuminations, the fact that the patient may not be available before an operation, the fact that the surface that needs registering may not be visible before an operation, and the surface may be occluded, or its appearance may change during an operation.

3.6.3 Tracking

In section 3.3, tracking related algorithms were reviewed. It was stated that algorithms for computing 'structure from motion' were not applicable to this registration task. One method used a whole image sequence to reconstruct a set of points that matched the information in a series of video images. Another method computed motion estimates from optical flow. Optical flow produces a dense approximation of the true motion field but again requires two images to compute the change in intensity pattern over time. Thus these methods are not relevant for registering a model to a single frame. The most relevant tracking algorithms are those which match a model with an image, and in general these methods calculate the extrinsic camera parameters using iterative procedures such as Newton's optimisation to minimise a cost function and are essentially similar to the pose estimation algorithms, except the emphasis is on speed. For the same reasons that most computer vision algorithms are not relevant to this medical application, neither are the tracking algorithms.

3.6.4 Medical 2D-3D Registration Algorithms

In this review, sections 3.5 to 3.5.4.4 investigated the medical 2D-3D literature, sub-dividing the algorithms into point based, contour based, surface based and intensity based. The algorithms could also have been classified as video - MR/CT based, or radiograph/fluoroscopy - CT based.

Penney provides a thorough comparison of the radiograph/fluoroscopy - CT registration algorithms [Penney, 1999]. Two points are important. First, radiograph/fluoroscopy images are completely different types of images to video images. The former display the

amount of X-ray radiation passed through an object, the latter, the visible radiation reflected off of an object. Secondly, Penney points out the advantages of intensity based algorithms over feature based algorithms. Penney selects an intensity based algorithm on the basis that the intensity based algorithms are more accurate, they avoid a segmentation process which may be error prone, and the fact that in 3D-3D volume registration intensity based methods have outperformed feature based methods [West *et al.*, 1997; West *et al.*, 1999]. The intensity based, radiograph - CT registration algorithms reviewed all illustrate the method that intensity based matching can be performed by simulating a 2D image from the 3D CT, and comparing the simulated image with the real radiograph. In each case, similarity measures such as correlation or gradient measures [Lemieux *et al.*, 1994], pattern intensity [Weese *et al.*, 1999], gradient difference [Penney *et al.*, 1998] and so on are optimised by a multidimensional search strategy. Penney illustrates that the choice of similarity measure must consider the available intensity information that can be matched, and also be robust to spurious information such as interventional stents in the radiograph that will not match any part of a pre-operative CT.

3.6.4.1 Video - MR/CT Registration Algorithms

As with the camera calibration, and pose estimation algorithms, it is difficult to compare the video - MR/CT algorithms. Table 3.1 provides a summary of the main points of each of the reviewed algorithms, and table 3.2 shows the testing procedure or how many times the registration algorithm has been applied.

The degree of automation often determines clinical applicability. This is not considered here in detail, as many of the details are missing from the papers. It is sufficient to point out that Edwards, Colchester and Grimson's method are used clinically, the others are not. Grimson's method has been used on 70 patients [Grimson *et al.*, 1998], Edwards' most recent system on 3 patients [Edwards *et al.*, 1999b], and Colchester's system on 6 neurosurgical operations [Colchester *et al.*, 1996]. Edwards' is the only system whose registration accuracy has been compared with bone implanted markers, and the accuracy ranges from 0.5 - 4mm. Furthermore, Edwards' system tracks the patient moving relative to the video cameras, and updates the registration at 1-2 times per second. Grimson and Colchester both use surface matching and cite the mean distance between the two surfaces as a measure of registration performance. This cannot be considered an accurate error metric, but a low distance of 1.6 and < 1 millimetres for Grimson and Colchester's

Algorithm	Feature Space	Similarity Metric	Search Space	Search Strategy
Edwards	Tracked Markers	3D distance^2	ext.	Direct
Betting	2D Contour 3D Surface	5D distance^2	ext.	Modified ICP
Betting	3D Surfaces	6D distance^2	ext. ext, int.	Modified ICP
Colchester	3D Surfaces	3D log distance	ext.	Decreasing Step Size
Grimson	3D Surfaces	3D Gaussian distance	ext.	Davidon Fletcher Powell Quasi Newton
Lavallee	2D Contour 3D Surface	3D Distance^2	ext.	Levenberg Marquardt
Viola	2D Intensities 3D Surface	Mutual Information	ext.	Stochastic Gradient Descent

Table 3.1: A summary of video-3D algorithms.

Algorithm	Images	Tests	Accuracy	Time	Hardware
Edwards	Clinical	Many	0.5-4mm	Real time	Sun,Intergraph
Betting	Video/CT Phantom	1	0.76 pix 0.17 degrees	10s	DEC Alpha
	X-ray/CT Skull	1	0.79 pix 0.7 degrees		DEC Alpha
Betting	2 Video MRI, face	1	1.6 mm	50s	DEC Alpha
Colchester	Clinical	Many	$<$ 1.0mm	180s	Sun Sparc IPX
Grimson	Clinical	Many	1.6mm	120-240s	
Viola	Video/CT Phantom	200	1.34,0.99,11.01mm 3.09 degrees	35s	Sun Sparc 5

Table 3.2: A summary of video-3D algorithms testing and performance.

methods respectively indicates that the registration was probably successful. Grimson's and Colchester's method both require 120 - 180 seconds to re-register if the patient moves.

Both Colchester *et al.* and Grimson *et al.* use surface matching and have used their algorithms in clinical situations. Both methods rely on the accuracy of surface reconstruction. Colchester's method projects lines onto a surface. Two video cameras capture images, an edge detection algorithm used, and corresponding points are matched between views. Edge detection and corresponding point matching are known to be difficult problems, that to date are still ongoing research areas. The problem is increased if the surfaces are wet, shiny, and overly textured. For each incorrectly identified edge pixel, the search for correspondences across views increases. Thus it would be better to have an algorithm that does not rely on an edge detection process. Grimson uses a laser scanner, which gives a very accurate surface reconstruction. However, laser scanners can be inconvenient to use within a medical environment. It would be better to have a system that does not need a laser.

Betting and Feldmar's paper [Feldmar *et al.*, 1997] simply repeats the results in [Feldmar *et al.*, 1994; Betting and Feldmar, 1995; Betting *et al.*, 1995]. Yet for each of the two algorithms reviewed, only one registration result exists. Furthermore no gold standard is used and the methods only use phantoms. The main problem with using a method such as these is that they require segmentation in both the 2D and 3D images, and also require that an external contour is indeed present in the video images. This will make it difficult to apply to a wide variety of cases.

Viola's method removes the need for segmentation of the 2D image. This is a significant step in the right direction. However, it has only been tested on skull phantoms. At the very least, further tests need to be done. In addition, it assumes that the surface being viewed is of one material type, and textureless. i.e. It is one smooth colour, reflecting light in a consistent manner over the whole surface, subject to lighting conditions. This is in practice not the case. Most surfaces exhibit some level of texture. In addition, in an operating room environment, surfaces become wet, and can be covered in blood. Viola's method may not work well in practical applications.

3.7 Conclusions

In this chapter, papers relating to registration have been studied. From the camera calibration papers, Tsai's method [Tsai, 1987] is freely available, and widely used. From the medical video - MR/CT registration algorithms, the scenarios and images which have been used suggest that in the first case it is reasonable to develop a method that only recovers the six extrinsic camera parameters. Thus Tsai's method will be used in the remainder of the thesis to recover the intrinsic and/or extrinsic parameters.

The computer vision literature on pose estimation is vast. However, the algorithms for pose estimation of cars, estimating pose from aerial views from an airplane and so on, are often not applicable for registering 3D medical images to video images.

In the medical field, two categories of algorithms were reviewed. For, radiograph/fluoroscopy - CT images, the intensity based methods were more accurate and robust compared with the feature based approaches.

There exists manual methods to perform medical video - MR/CT registration [Gleason *et al.*, 1994; Nakajima *et al.*, 1997]. The method is to display on a computer monitor a video image of the patient, and a rendering of surfaces on the pre-operative data. The patient and/or camera are moved until the fused images appear aligned. The goal of this project is to produce an automatic procedure to achieve registration and validate its accuracy. Thus manual methods are not considered further. In addition, for methods using bone pins or a less invasive LADS bite block [Edwards *et al.*, 1999b], systems already exist for image guided surgery that perform this task well. So, in order to extend 2D-3D registration to as many applications as possible, a non-invasive method will be used, such that for instance the radiotherapy applications mentioned in the introduction can be realised.

For the video-MR/CT algorithms, point and surface based algorithms are currently used. These are limited by the accuracy of the extraction process. Viola's work on producing a method which does not rely on a 2D segmentation or feature extraction and is a significant advancement. Thus, having looked at the literature, the specifications for a 2D-3D registration algorithm are as follows:

The algorithm will calculate the extrinsic parameters only. This could be extended to include intrinsic parameters in the future. The algorithm will be intensity based. No 2D segmentation will be performed. The algorithm will use either a 3D surface segmentation or some volume rendering approach to define significant surface points. All algorithms described in this chapter, and the transformations described in the previous chapter, have all assumed that the object of interest is a rigid body. In this thesis, rigid body registration is used because it is mathematically simple. Therefore, the proposed algorithm will only be applicable to register images of reasonably rigid objects. i.e. the head as opposed to the abdomen. Clearly however, even the skin can deform by several millimetres, thus it must be understood that the algorithm uses rigid body registration as a reasonable, simple starting point. A registration algorithm that incorporates deformation of the 3D image would be more realistic and more accurate. However, such an algorithm will be left for future work.

Considering algorithm speed, i.e. the time taken to register, it would seem reasonable that if an algorithm took 3-5 minutes, then it would be sufficient for the proposed applications. Clearly however, a quicker algorithm would be more convenient to use.

In the first half of the thesis, the algorithm will be tested on video and CT images of a plastic skull phantom. In the second half of the thesis, the algorithm will be tested on volunteers. This means that video images will be of the skin surface. In this case, segmentation of a skin surface from an MR or CT scan is not difficult. Segmentation is an ongoing area of research, but will not be covered in this thesis, as any segmentation is likely to be a) simple and b) easily accomplished using software packages such as ANALYZE (Biomedical Imaging Resource, Mayo Foundation, Rochester, MN, USA.) An initial algorithm will be developed that works with one smooth textureless surface. Ultimately, the algorithm will be required to work under varying lighting conditions, with varying surface texture, possibly even changing surface texture, and then with multiple surfaces e.g. skin and bone.

Part II

Methods, Experiments And Results

Chapter 4

Single View Registration

4.1 Introduction

This chapter describes an algorithm to register a 3D medical image to a single 2D video image. The algorithm is an image intensity based method inspired by the work of Viola [Viola, 1995; Viola and Wells III, 1997]. Whilst the proposed algorithm is a significantly different implementation from that of Viola's algorithm, this chapter describes work to test existing ideas and concepts. Registration is achieved by producing rendered images of a surface model extracted from a 3D image, and measuring the similarity of the rendered and video images using mutual information. The mutual information is maximised with respect to the pose parameters until the optimum pose is found. It is assumed that the intrinsic parameters of the camera are known. This chapter describes the algorithm in detail, and demonstrates its performance. The algorithm was tested using a CT scan and various video images of a skull phantom. The experiments in this chapter assess the algorithm with respect to (1) the rendering light source position, (2) registration robustness, (3) accuracy, (4) range of capture, (5) performance with changing field of view, (6) performance with changing focal length of the video camera and (7) performance with other similarity measures.

4.2 Aim

The main aim of this chapter is to investigate experimentally whether the mutual information of a single rendered and video image pair, optimised using a gradient ascent search strategy is sufficient to register a video image with a 3D image. The measure of similarity, mutual information, and the search strategy are the important issues and are now discussed in detail.

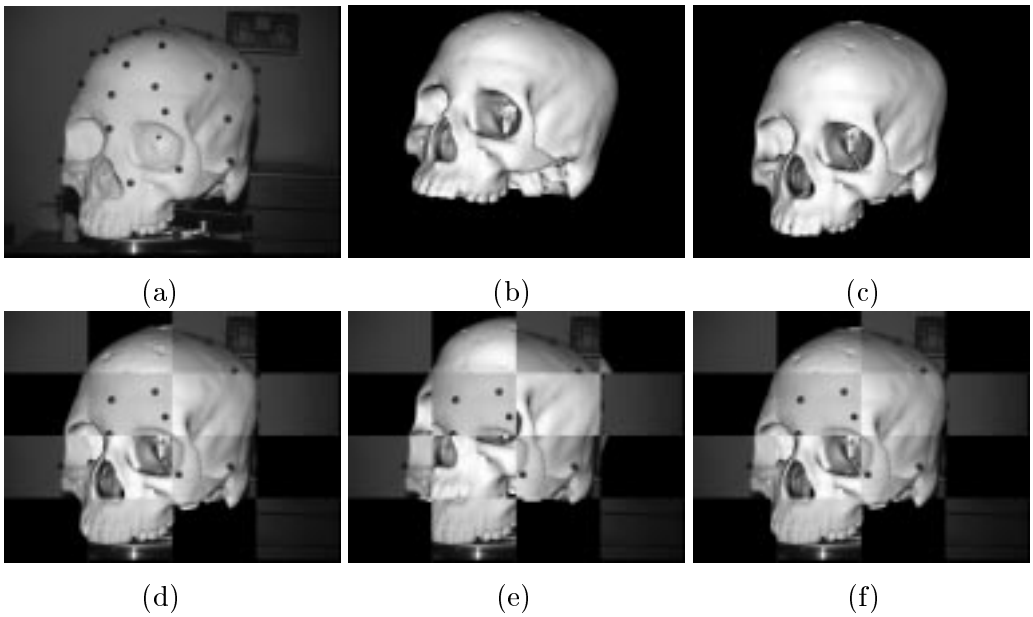


Figure 4.1: These images demonstrate the algorithm. (a) A sample video image. (b) A rendering of the surface model at a misregistered pose. (c) A rendering of the surface model at a registered pose. (d) A rendering performed at the 'gold standard' pose, mixed with the video image. (e) A rendering of the surface model at a misregistered pose and mixed with the video image. (f) A rendering of the surface model at a registered pose and mixed with the video image.

4.3 Methods

Figure 4.1 illustrates the algorithm. The aim is to recover the video camera's extrinsic parameters (defined in section 2.2.8). The extrinsic parameters are the six degrees of freedom, three rotations and three translations that relate the 3D image and camera co-ordinate systems. The algorithm is illustrated in figure 4.1: A video image (a) is taken. The algorithm starts with incorrect extrinsic parameters and produces a rendering (b) using these incorrect extrinsic parameters. The similarity measure shows that the similarity between (a) and (b) is low. A gradient ascent search strategy searches for a better set of extrinsic parameters that makes the rendering more similar to the video image. Figure (c) shows the surface model rendered at the registered pose i.e. with the recovered set of extrinsic parameters. Figures 4.1(d), (e) and (f) each show a 'checkerboard' mix of rendered image and video image, where the rendered image is produced at the 'gold standard' (a known ground truth), misregistered and registered pose respectively. The checkerboard display is for visualisation purposes only. From figure 4.1(f) we see that the rendered image, and hence the 3D image are aligned or registered with the video image.

4.3.1 Choice Of Similarity Measure

As discussed in section 2.5, video image intensities depend on an object's ambient, diffuse and specular reflectance characteristics, and also on the relative position of light sources and the viewing video camera. Predicting or modelling the exact image intensity is difficult as many possibly unknown factors are involved.

Given a rendered image, and a video image, a number of different similarity measures could be used. Sums of squared differences (SSD) can be used to compare video and rendered image intensities if, at alignment, a video image intensity should exactly match a rendered image intensity. Normalised cross correlation (NCC) could be used as an intensity based similarity measure if, at alignment, the video image intensities should be related to the rendered image intensities by some linear function.

In practice, neither of these conditions are likely to be true. The surface is rendered using a Lambertian reflection model. This implicitly assumes that as the rendered surface is assigned the same colour and reflectance throughout, e.g. a white surface, but shaded according to a Lambertian reflectance model, that the surface being matched in the video image should also have one colour and reflectance throughout. The video image will be corrupted by noise, and the surface being observed by the video camera is likely to be textured, i.e. a human face. Thus SSD and NCC are likely to fail as similarity measures as they impose restrictions on how functionally similar rendered image and video image intensities should be.

Mutual information has proven to be a flexible, accurate and robust measure for 3D volume registration [West *et al.*, 1997; West *et al.*, 1999]. Mutual information is a function of the probability of corresponding pixel (or voxel) intensity values. Thus, no functional relationship between two sets of image intensities is assumed, and hence mutual information seems a suitable starting point for computing the image similarity. The experiments in this chapter test whether mutual information is indeed a suitable similarity measure for this algorithm.

Viola used the mutual information between the video intensity and the corresponding model surface normal as a similarity measure [Viola and Wells III, 1997]. The video image intensity is described with one random variable, and the unit surface normals require two random variables. Hence, mutual information is calculated using a 3D joint probability distribution. By optimising mutual information of these three variables, Viola's algorithm

obtains a pose that produces the most consistent match between video image intensities and the corresponding model surface normals. This aims to avoid specifically modelling the lighting function, it merely assumes that some functional relationship exists. Viola adopts this method to solve a general purpose pose estimation problem.

Producing a rendering and measuring the similarity with a video image using mutual information assumes that we can approximate 'sufficiently closely' the actual lighting function evident in the video image. This is not an unreasonable step to take. In many medical applications using video images, e.g. operating microscopes or endoscopes, the light source is known to be fixed relative to the video camera. Furthermore the light source is known to be co-axial or near co-axial with the camera. This knowledge can be exploited when producing the renderings. In other words, it is feasible and practical to impose lighting restrictions as doing so should make the rendering more similar to the video image, and allow for more reliable matching.

Viola's method was originally implemented using Parzen Windowing [Duda and Hart, 1973] to estimate the necessary 3D probability distributions. Hata, at the same group as Viola, implemented Viola's method, but using histograms to form discrete approximations of the probability distributions [Hata *et al.*, 1996]. A similar method to Viola's was implemented by the author using histograms, and a simple gradient ascent search strategy, but it failed to work. No communication with Viola took place, and it was concluded that estimating the 3D probability distributions was unreliable when sampling statistics are small in comparison with the variability of the data. In other words if a 3D histogram was formed, e.g. $64 \times 64 \times 64$ bins, then the data available was too sparsely distributed within this histogram to enable accurate estimates of the underlying probability distributions. Further, possibly collaborative work, could provide further insight.

To summarise, the rendering method described in the previous section was chosen in preference to Viola's surface normal method, as an initial implementation of Viola/Hata's algorithm failed to work, and the rendering method showed some promise. Furthermore, the rendering method is simple to implement and the estimation of 2D joint probability distributions of rendered image intensities and video image intensities is reliable. Given that in practice, the scene illumination can be controlled when capturing an optical image and therefore duplicated (albeit rather simplistically) when producing a rendering, then the rendering method seems a reasonable approach.

Using histograms to calculate the probability distributions [Studholme *et al.*, 1999; Maes *et al.*, 1997] is more popular than Parzen Windowing. This is likely to be due to their simple implementation and their suitability to the discrete image intensity information. Studholme [Studholme *et al.*, 1999] proposed normalised mutual information as an overlap invariant similarity measure. In the following experiments, the overlap between video and rendered images does not vary significantly, and so mutual information is used.

4.3.2 Evaluating Mutual Information

Using standard computer graphics techniques [Foley *et al.*, 1990], a rendering of a 3D surface model extracted from the 3D image can be produced. To produce a rendering, a virtual camera and virtual light source are created, each with a given pose. The surface is assumed to have purely Lambertian reflection. The light source used to produce the video images, is known to be approximately co-incident and co-axial with the video camera. Thus, the virtual light source is set to be exactly aligned with the virtual rendering camera to produce similar surface shading. Section 3.5.4.1 described the similarity measure mutual information, the formulation of which is repeated here.

A histogram is formed from the intensities in each image. Each histogram is divided by the total number of counts in that histogram, which results in a probability distribution. Let v be an intensity value in video image V and likewise r be an intensity value in rendered image r. Let $p(v)$ denote the probability of intensity v in image V, $p(r)$ be the probability of intensity r in image R and $p(v,r)$ the joint probability of intensity v and r occurring at corresponding pixel locations in images V and R respectively. Let $\mathcal{V}$ denote a random variable describing the distribution of intensity values in image V and likewise $\mathcal{R}$ denote a random variable describing the distribution of intensity values in image R. The entropy H of each random variable $\mathcal{V}$ and $\mathcal{R}$ denoted using $H(\mathcal{V})$ and $H(\mathcal{R})$ is

$$H(\mathcal{V}) = -\sum_v p(v)\log p(v) \qquad H(\mathcal{R}) = -\sum_r p(r)\log p(r) \tag{4.1}$$

and the joint entropy $H(\mathcal{V},\mathcal{R})$ is

$$H(\mathcal{V},\mathcal{R}) = -\sum_v \sum_r p(v,r)\log p(v,r) \tag{4.2}$$

and the mutual information is then

$$I(\mathcal{V};\mathcal{R}) = H(\mathcal{V}) + H(\mathcal{R}) - H(\mathcal{V},\mathcal{R}) \tag{4.3}$$

$H(\mathcal{V})$, $H(\mathcal{R})$, $H(\mathcal{V},\mathcal{R})$ and $I(\mathcal{V},\mathcal{R})$ are evaluated using pixel locations where the video and rendered image intensity value is not zero (background). From equation (4.3), the similarity of a video image V and rendered image R can be computed. The similarity or value of mutual information will change as the pose of the rendered surface model is changed. The pose of the rendered surface model is controlled by the current estimate of the extrinsic camera parameters (defined in section 2.2.8) which are the parameters to be determined for this registration task. The mutual information is maximised with respect to the extrinsic parameters.

4.3.3 Choice Of Search Strategy

The choice of optimisation strategy or search strategy depends on how well behaved the cost function is with respect to the parameters being optimised [Press *et al.*, 1992].

In 3D-3D medical image registration mutual information provides a smooth, well behaved parameter space (see section 2.4), leading to reliable registration. Recently interpolation artifacts have been discussed [Pluim *et al.*, 1999], the exact cause of which is as yet unknown. However, in general mutual information is smooth and well behaved.

If the cost function is smooth and well behaved, even a simple search strategy can find the maximum (or minimum). The gradient ascent search is chosen for its simplicity, in the knowledge that it is not necessarily a good search strategy as it can take many iterations to proceed along a valley floor in the parameter space [Press *et al.*, 1992]. If this method performs badly then either a different similarity measures can be tested or a more complicated search procedure such as simulated annealing [Press *et al.*, 1992].

4.3.4 Search Strategy

To optimise the mutual information between video and rendered images, a gradient ascent method is used. Let $\mathbf{t} = (t_x, t_y, t_z, r_x, r_y, r_z)$ which are the extrinsic camera parameters, three translations and three rotations being optimised. Let τ be the current iteration, $\mathrm{MI}(\mathbf{t})$ be the value of MI with transformation parameters $\mathbf{t}$, $\nabla\mathrm{MI}$ be a 1×6 unit length vector of partial derivatives of mutual information with respect to each parameter in $\mathbf{t}$, S be a step size, then

$$\mathbf{t}_{\tau+1} = \mathbf{t}_\tau + S\,\nabla\mathrm{MI} \tag{4.4}$$

The partial derivatives are calculated numerically by central differences using an increment to each parameter of size S.

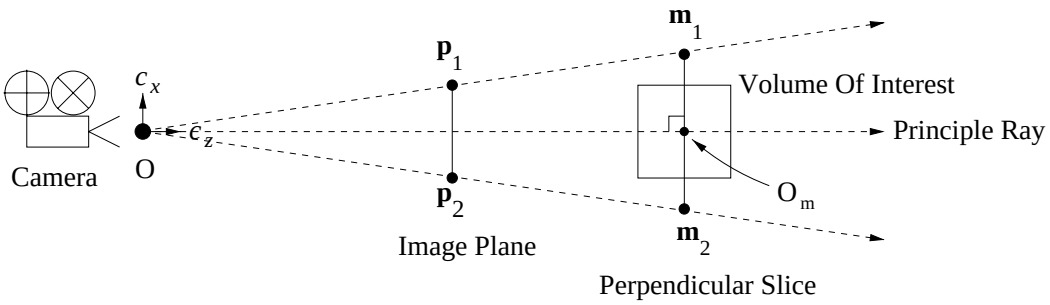


Figure 4.2: The pixel to millimetre ratio. See text, section 4.3.5.

4.3.5 Matching 2D And 3D Resolution

In 3D-3D medical image registration, great care must be taken to make sure that the two images being registered are compared at similar resolutions. The resolution of two different images can be made comparable by resampling the higher resolution image to match the lower resolution one [Studholme, 1997]. Matching image resolutions appears to be an important consideration in 2D-3D registration. For example, the experiments in this chapter use a CT scan of a skull phantom. The CT scan has voxel sizes of 0.488 $\times$ 0.488 $\times$ 1.0 mm. A typical video image used in this chapter can have 768 $\times$ 576 pixels, spanning a field of view (FOV) of approximately 230 $\times$ 173 mm. Thus each pixel represents approximately 0.33 $\times$ 0.33 mm and hence the video image will contain information at a spatial resolution not present in the CT scan. Both theoretically and experimentally this extra information might be a hindrance to the registration algorithm. Therefore, video image resolution was reduced by convolution with a Gaussian kernel. The standard deviation of the Gaussian kernel was computed by first calculating a pixel to millimetre ratio m and then a standard deviation σ as shown below. Figure 4.2 shows the model coordinate system origin $\mathrm{O_m}$ and the camera centre of projection O, which is the origin of the camera coordinate system. Two image pixels $\mathbf{p}_1$ and $\mathbf{p}_2$ lying on the same vertical column in the image are chosen. A plane through the origin and perpendicular to the principle ray is defined, and the rays from the centre of projection, through $\mathbf{p}_1$ and $\mathbf{p}_2$ intersect the plane at $\mathbf{m}_1$, and $\mathbf{m}_2$. Let s be the mean of the x, y, and z voxel dimensions. The vertical pixel to millimetre ratio m_v is approximated by $s(\parallel \mathbf{p}_1 - \mathbf{p}_2 \parallel)/ \parallel \mathbf{m}_1, \mathbf{m}_2 \parallel$. Similarly two points lying in the same horizontal row of the image plane can be chosen and a horizontal pixel to millimetre ratio calculated m_h. The mean average pixel to millimetre ratio m is then $(m_v + m_h)/2$.

This value is only an approximation as the true pixel to millimetre ratio will vary with distance from the camera and from the principle ray. If for instance $m = 2$ then the video image is assumed to be *approximately* twice the resolution of the 3D data.

From m the standard deviation σ of the Gaussian convolution kernel can be calculated as follows. A 1D image in the spatial domain can be considered as a continuous signal sampled with a set of delta functions. Figure 4.3(a) shows a set of 1D delta functions separated by Δx. Figure 4.3(b) shows the Fourier transform of (a), which is also a set of delta functions separated by $1/\Delta x$. This illustrates that units of Δx in the spatial domain correspond to units of $1/\Delta x$ in the frequency domain.

If $m \leq 1$, no blurring is applied. If $m > 1$, blurring is applied to the video image to reduce the effective resolution. Consider the case in figure 4.3(d): In this case, let $m = 2$, i.e. the value for σ should halve the resolution. The frequency domain is multiplied with a Gaussian function $G(u, \sigma_u)$ whose full width at half maximum (FWHM) covers $\pm 1/(m\Delta x)$ of the frequencies. The Gaussian function $G(u, \sigma_u)$ is described using

$$G(u, \sigma_u) = e^{-\left[\frac{u^2}{2\sigma_u^2}\right]} \tag{4.5}$$

For example if $\Delta x = 1$, $m = 2$,

$$\frac{1}{2} = e^{-\left[\frac{(\frac{1}{2})^2}{2\sigma_u^2}\right]} \tag{4.6}$$

which gives $\sigma_u = 2.35$. Multiplication with a Gaussian function of standard deviation σ_u in the Fourier domain is equivalent to convolution in the spatial domain with a Gaussian function of standard deviation $\sigma = 1/\sigma_u$.

To summarise, the standard deviation of the convolution kernel σ is calculated from m using

$$\sigma_u = \sqrt{\frac{-(\frac{1}{m})^2}{\ln(\frac{1}{2})}} \tag{4.7}$$

$$\sigma = 1/\sigma_u \tag{4.8}$$

To implement the Gaussian blurring it is necessary to decide on a width for the convolution kernel. An adequate choice is that the width $w = 5\sigma$ which subtends 98.76% of the Gaussian function [Trucco and Verri, 1998].

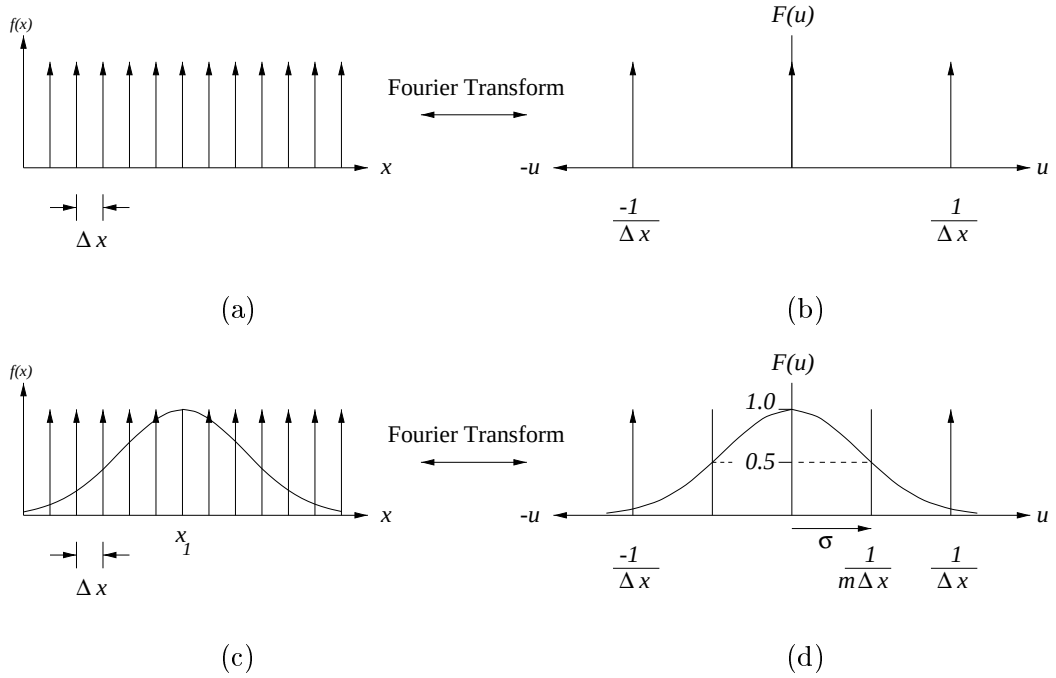


Figure 4.3: Blurring kernels (see text).

4.3.6 Multi-Resolution Approach

In addition to the smoothing to match the 2D and 3D resolutions, a multi-resolution search strategy was implemented. For a general optimisation problem, the purpose of a multi-resolution approach is twofold, to avoid local minima, and depending on implementation, to increase processing speed.

Avoiding local minima is usually achieved by starting an algorithm at a coarse level of detail, where for instance images are blurred to remove information, and large step sizes are taken through parameter space [Studholme *et al.*, 1995]. Thus, small details in the image will not affect the similarity measure, and the search space is much smoother. The algorithm then proceeds by finding the best solution at a given resolution and then increasing the resolution to finer and finer detail, whilst reducing the step sizes.

In general, changing the resolution of the surface model during registration is too costly as this would involve computing and storing one surface model for each resolution. In addition the rendered images will usually contain less information than the video image, and smoothing the rendered image would have to be repeated at each iteration. Therefore a multiresolution approach was implemented which smoothes the video image only. The algorithm first matches the surface model to a low resolution image, and then to

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

Figure 4.4: Registration is performed at a low resolution and repeated at progressively higher resolution. Here, $m = 3$, (a) shows the original image, (b) image blurred with Gaussian $\sigma = 10.0$, (c) $\sigma = 5.0$, (d) $\sigma = 2.5$

progressively higher resolution images. This is illustrated in figure 4.4, and can be seen in the algorithm outline in figure 4.6.

The blurring to match 2D and 3D resolutions is denoted by m and the blurring to implement a multi-resolution search strategy is denoted by resolution level L. The total blurring factor $b = m\,L$, and the value b used instead of the value m in equations (4.7) and (4.8). Thus for pixel to millimetre ratio m and resolution levels of $L = 4, 2, 1$:

$$b = m\,L \tag{4.9}$$

$$\sigma_u = \sqrt{\frac{-(\frac{1}{b})^2}{\ln(\frac{1}{2})}} \tag{4.10}$$

$$\sigma = 1/\sigma_u \tag{4.11}$$

The multi-resolution blurring is illustrated in figure 4.4. The original image is shown in image (a). $m \approx 3$. Three resolutions were chosen, $L = 4, 2, 1$ which, using equations (4.9),(4.10) and (4.11) gives $\sigma \approx 10.0, 5.0, 2.5$. The original image convolved with a Gaussian kernel of standard deviation of $\sigma \approx 10.0, 5.0, 2.5$ are shown in (b),(c) and (d) respectively. The exact amount of smoothing was found not to be a critical factor in the experiments that follow.

4.3.7 Lighting Models

Section 4.3.1 justified the use of a rendering based matching method, where a rendered image of the 3D model is matched to a video image, using mutual information. The rendered image is produced by defining a virtual light source and camera with a pose relative to the 3D model.

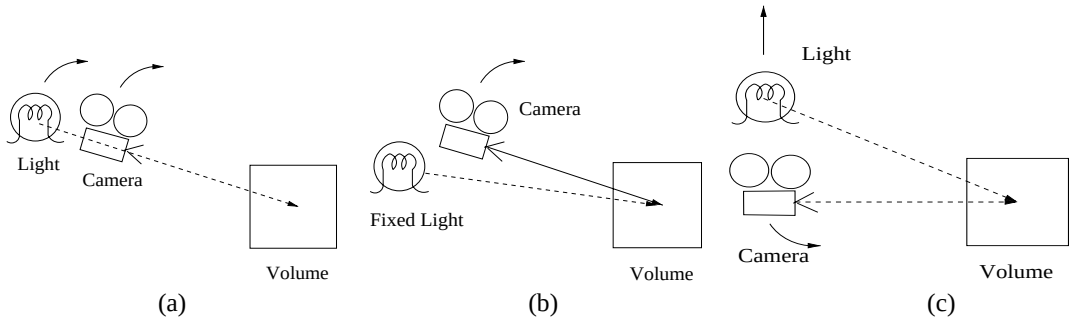


Figure 4.5: The position of the rendering light source can be (a) aligned with the camera (b) fixed, (c) optimised independently.

During the registration process, the camera is moved relative to the 3D model, until the maximum of mutual information is reached, which should be at the registered pose. Mutual information should be able to match images without assuming a specific relationship between rendered and video image intensities. So, does it matter whether the virtual light source, and virtual camera are kept aligned, or is mutual information sufficiently flexible to provide accurate alignment regardless of virtual light position? This is tested experimentally in section 4.4.2 using the following scenarios illustrated in figure 4.5.

Figure 4.5(a) shows a rendering setup where the rendering light source and rendering camera are aligned with each other. The light is positioned to the left of the camera for clarity. The VTK implementation of a point light source [Schroeder *et al.*, 1997] defines a light with a position and focal point. The vector from the light's position to the light's focal point defines the direction of all the virtual rendering light rays i.e. the light simulates a point light source at infinity emitting parallel rays of light onto the rendered scene. The two characteristics of this setup are that rays of light are all parallel to the camera's optical axis, and when the camera moves relative to the volume of interest, the light is moved with it. This method is called a 'moving' light source.

Figure 4.5(b) shows a setup where the light is given some initial position and direction, to specify the direction of the light rays. During optimisation, the camera's extrinsic parameters are adjusted to find the best pose, but the light remains in a fixed pose relative to the volume of interest. This method is called a 'fixed' light source.

Figure 4.5(c) shows a set up where the pose of the rendering light and camera are optimised independently. This means that the registration algorithm tries to find the pose of the camera that creates the best alignment and also the best lighting position to make the rendering look most similar to the video image. This setup is called an 'optimising' light source.

4.3.8 Summary Of The Algorithm

The algorithm is summarised in figure 4.6 on the following page in pseudo-code. The algorithm employs a multi-resolution gradient ascent search strategy to maximise the mutual information of the video image and a rendered image computed at each iteration, with respect to the extrinsic camera parameters $\mathbf{t} = t_x, t_y, t_z, r_x, r_y, r_z$ which determine the pose of the surface model and hence the 3D image with respect to the video image.

The controlling parameter of the search strategy is the step size S, which is applied in equation 4.4. The value of S must be chosen to reflect the likely size of the search space. For instance if the initial estimate is approximately aligned to within four millimetres and degrees for each of the extrinsic parameters, then S can be set to four. If the initial estimate is closer to the expected solution, S should be smaller, and if the estimate is greatly misaligned, S should be set to a larger value. The experiments in the next section investigate the performance of the algorithm as the misregistration increases.

Another parameter is the histogram size used for the calculation of mutual information. Studholme states that for 3D-3D registration the choice of histogram size from $32 \times 32 \ldots 256 \times 256$ bins has little affect on the accuracy of precision of the final registration estimate [Studholme, 1997]. However, in 3D-3D registration, the lower bound of the number of bins in the histogram is determined by the number of material types delineated by the imaging modality. The upper bound is determined by the resolution of the analogue to digital converter (ADC) used to digitise the image. In the method proposed in this chapter, the image intensities vary with the surface normal of the object and surface model. This means that the video and rendered image intensities represent a continuous quantity, not a discrete number of material types. Thus the video and rendered image intensities are discrete representations of a continuous quantity i.e. there is no lower limit. The upper limit will be 256 bins as the images are all stored with 8 bits. With fewer bins, the histogram will be more densely populated and the mutual information will change more gradually as the extrinsic parameters are changed. Thus,

```
procedure Compute_Similarity()
    set pose according to extrinsic parameters t_x, t_y, t_z, r_x, r_y, r_z
    produce rendering
    compute mutual information of video and rendering
    return mutual information
end procedure
procedure Register()
    for each L in 4, 2, 1
        Set histogram size to 256/L by 256/L
        Calculate m and b using equation (4.9)
        Calculate σ_u using equation (4.10) and σ using equation (4.11)
        Convolve video image with a Gaussian filter, standard deviation σ, width w
        From current estimate of t = t_x, t_y, t_z, r_x, r_y, r_z
        Set Current_Similarity = Compute_Similarity()
        Set step size S to some initial value
        while S > 0.05
            for each j in t_x, t_y, t_z, r_x, r_y, r_z
                increment parameter j by i × S
                plus_j = Compute_ Similarity()
                decrement parameter j by 2 × i × S
                minus_j = Compute_ Similarity()
                increment parameter j by i × S
                gradient_j = plus_j − minus_j
            end for each j
            calculate unit gradient vector from each gradient_j for each j ∈ t_x, t_y, t_z, r_x, r_y, r_z
            normalise gradient vector to length S
            add gradient vector to t
            if Compute_Similarity() < Current_Similarity
                subtract gradient vector from t
                divide S by 2.0
            end if
        end while
    end for each
return the registration result = t
```

Figure 4.6: Mono View Registration Algorithm.

fewer bins are suitable for a low resolution estimate of mutual information. The method used in this chapter was chosen to be: At each resolution $R = 4, 2, 1$, the number of bins was $64 \times 64, 128 \times 128, 256 \times 256$ respectively. As resolution increases, so does the histogram size, to make the algorithm more sensitive to changes in the registration parameters. The exact number of histogram bins was found not to be a critical factor in the experiments that follow.

4.3.9 Protocol For The Evaluation Of The Test Procedure

The test procedure is summarised below.

- **A gold standard registration is defined**. The experiments use a specially prepared skull phantom with rigidly attached fiducial markers. Fiducial markers enable an independent gold standard registration to be calculated. This fiducial based registration is taken as the ground truth, i.e. the correct solution, against which registration performance is tested. The calculation of the gold standard registration determines the video camera's intrinsic and extrinsic (t_x, t_y, t_z, r_x, r_y and r_z defined in section 2.2.8) parameters. See section 4.3.10.

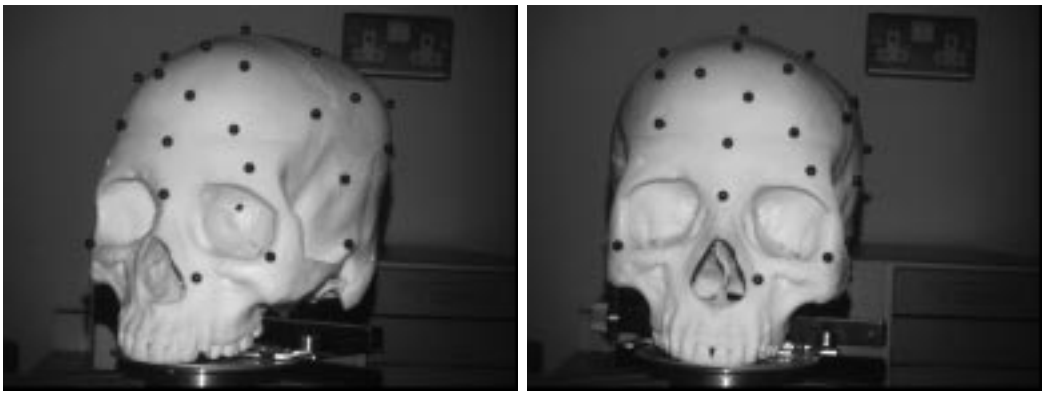
- **A misregistration is added.** From the known gold standard pose, an offset is added so that the surface model and 2D video image are mis-registered. This is done repeatedly and systematically to ensure an unbiased and rigorous testing procedure. See section 4.3.11.

- **The algorithm is used to register.** For each mis-registration, the algorithm then registers the surface model and the video image. See sections 4.3.8 to 4.3.7.

- **Registration is assessed.** Each registration is assessed as a success or failure. Each starting pose is calculated as an offset from a given pose, which can be either a gold standard, a manual estimate or a previous registration. Thus it is expected that all the registrations should cluster around a mean pose. A failed registration is a registration where any one of the extrinsic parameters $t_x \ldots r_z$ has moved further away from the expected gold standard position than when it started.

- **An error measure is calculated.** The performance is assessed using the 'projection error' and '3D error', which are calculated for all successful registrations. These error measures are used to assess accuracy and precision. See section 4.3.12.

These items are now discussed in further detail.



(a) (b)

Figure 4.7: Two video images of the skull phantom.

4.3.10 Gold Standard Registration

The skull phantom used in the experiments has 23 black ball bearings with 5mm diameter attached to it, which can be seen in figure 4.7. The ball bearings serve as fiducial markers and can be accurately localised in the 3D image using an initial manual estimate as a starting point for an intensity weighted centre of gravity operator [Wang *et al.*, 1997]. A centre of gravity operator is capable of finding the centroid of the fiducials to sub-voxel accuracy [Bose and Amir, 1990; Chiorboli and Vecchi, 1993]. The markers are manually located in the 2D image.

Once corresponding pairs of 2D and 3D coordinates have been localised, a gold standard transformation can be calculated using Tsai's algorithm [1][Tsai, 1987]. The minimum requirement is 6 2D and 3D point pairs, and knowledge of the video camera sensor array element size. These parameters can be obtained experimentally [2].

Tsai's algorithm is a non-linear camera calibration technique which, given a set of 2D and 3D point correspondences optimises a set of extrinsic and intrinsic camera parameters as described in section 3.1. The output of Tsai's algorithm is the intrinsic and extrinsic camera parameters which are taken as the correct, or gold standard solution.

Note that when registration tests were performed, the black spherical fiducials used to calculate the gold standard, are air-brushed out of both the video and CT images. Thus they are not used to perform the registration.

[1] from Reg Wilson: `http://www.cs.cmu.edu/~rgw/`

[2] http://www.ius.cs.cmu.edu/IUS/usrp2/rgw/www/faq.txt

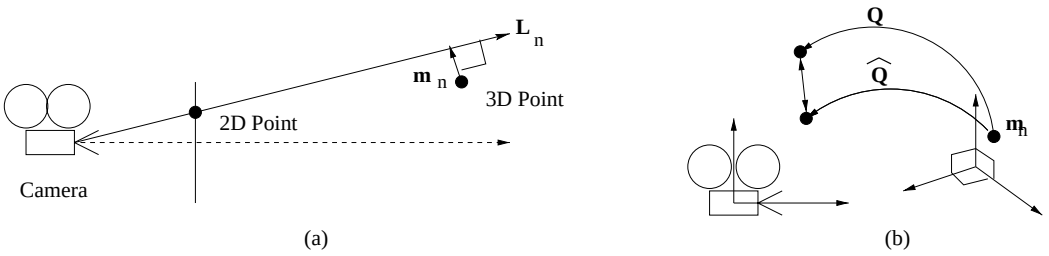


Figure 4.8: Two error measures for assessing registration error are (a) the projection error and (b) the 3D error. See text section 4.3.12.

4.3.11 Producing Misregistrations

An offset size δt is chosen. The offset $\pm\delta t$ is added to each of the gold standard extrinsic parameters. Adding this offset misregisters the 3D and 2D images, and the algorithm is used to recover, as closely as possible, the correct registration parameters. To ensure a thorough and systematic testing procedure, every combination of adding $\pm\delta t$ to all of the six parameters t_x, t_y, t_z, r_x, r_y and r_z is tested. This gives 64 tests for each value of δt, and for all 64 tests, the proposed algorithm is used to re-register the 2D and 3D images.

4.3.12 Error Measures

For each registration, two error measures are calculated. Projection error is illustrated in figure 4.8(a), and the 3D error is illustrated in figure 4.8(b). Projection error and 3D error are described below.

4.3.12.1 Projection Error

For a set of 2D and corresponding 3D points the projection error is calculated as

$$\text{projection error} = \frac{1}{N}\sum_{n=1}^{N} D(\mathbf{L}_n, \mathbf{m}_n) \tag{4.12}$$

where N is the number of points being used to evaluate the error, $\mathbf{m}$ denotes a 3D model point, $\mathbf{L}$ denotes a line projected through a 2D point, n denotes a specific point and line number and $D(\mathbf{L}_n, \mathbf{m}_n)$ is the closest Euclidean distance from the point $\mathbf{m}_n$ to the line $\mathbf{L}_n$. The projection error is the arithmetic mean of $D(\mathbf{L}_n, \mathbf{m}_n)$ evaluated over the set of point pairs. In the following experiments, the projection error is evaluated using all the points in the surface model. Each 3D point is projected onto the 2D image plane using the gold standard transformation. This gives perfectly matching 3D and 2D points. The

registration result will not give a perfect alignment. So if the registration transformation matrix is used to back project each of the 2D points, then each projected line will give an error $D(\mathbf{L}_n, \mathbf{m}_n)$, which is the measured quantity.

4.3.12.2 3D Error

For each point in a surface model $\mathbf{m}_1, \mathbf{m}_2, \ldots \mathbf{m}_N$ the 3D error between two rigid body transformations $\mathbf{Q}$ and $\hat{\mathbf{Q}}$, is given by

$$\text{3D error} = \frac{1}{N} \sum_{n=1}^{N} D(\mathbf{Q}\mathbf{m}_n, \hat{\mathbf{Q}}\mathbf{m}_n) \tag{4.13}$$

where $\mathbf{m}_n$ is a 3D model point where n denotes a point number, and $D(\mathbf{Q}\mathbf{m}_n, \hat{\mathbf{Q}}\mathbf{m}_n)$ is the Euclidean distance between the point $\mathbf{m}_n$ multiplied by a rigid body transformation matrix $\mathbf{Q}$, such as the gold standard rigid body matrix, and the same point $\mathbf{m}_n$ multiplied by another rigid body transformation matrix $\hat{\mathbf{Q}}$, such as the registration result.

4.3.13 Choice Of Error Measures

Consider a third error measure similar to projection error, which will be called 'pixel error'. A 3D point could be projected using equation (2.18) to obtain a 2D pixel coordinate and the Euclidean distance of this projected point from some gold standard pixel location could be calculated. The mean pixel error could be calculated using an appropriate set of points. However the error measured will be dependent on the pixel size. In terms of an image guided surgery application pixel size is unimportant. What is important is real distances in the patient space in millimetres. The projection error described in 4.3.12.1 has units of millimetres. Projection error will give a good indication of visually how well aligned the surface model is with the video image. This is useful for augmented reality applications where the objective is to overlay a rendering on a video image. However, if the surface model is a long way from the video camera, the projection error could be small, and the surface model could be mis-registered by a large distance along the optical axis of the camera. Thus 3D error gives a good indication of the accuracy in terms of fully recovering the correct transformation, which is useful if the objective is to use the 2D-3D registration to interact with or measure the real 3D space.

4.4 Experiments

The first half of this chapter described components of an algorithm to register a single video image to a 3D image. All images are of a plastic skull phantom. This section tests the performance of the algorithm as follows.

- **Testing the accuracy of the gold standard.** Any gold standard will have errors associated with it. A simulation was performed to study the accuracy of the gold standard registration as noise was added to the 2D or the 3D points. This enables the specification of the required fiducial localisation for a suitable gold standard. The fiducials were then extracted in both the 2D and 3D images, and a leave-one-out test used to experimentally test the accuracy of the gold standard. See section 4.4.1.

- **Testing which lighting model to use.** For small misregistrations of size $\delta t = \pm 4$ mm and degrees, the three lighting models moving, fixed and optimising were tested. See section 4.4.2.

- **Accuracy, Robustness and Range of Capture.** Using the moving lighting model, the algorithm was tested using misregistration sizes $\delta t = \pm 4, 8, 12, 16$ mm and degrees. See section 4.4.3.

- **Performance with Changing Field of View.** The two video images used in section 4.4.3 were masked to reduce the effective field of view. The registration tests were repeated for misregistration size $\delta t = \pm 8$ mm and degrees for images with different regions masked. See section 4.4.4.

- **Performance with Changing Focal Length.** Four images were taken using four difference focal lengths of the video camera. The registration tests were repeated for misregistration size $\delta t = \pm 8$ mm and degrees for each image. See section 4.4.5.

- **Comparison Of Similarity Measures.** For misregistrations of size $\delta t = \pm 8$ mm and degrees, the similarity measures mutual information (MI), normalised mutual information (NMI), normalised cross correlation (NCC) and gradient correlation (GC) were compared. These similarity measures are formulated and explained in section 3.5.4.1, and the experiments are described in section 4.4.6.

4.4.1 Validating The Accuracy Of The Gold Standard Registration

4.4.1.1 Methods

The gold standard extrinsic and intrinsic parameters calculated by the Tsai calibration will have errors associated with each parameter. Four tests were performed to assess the accuracy of the gold standard.

The image in figure 4.7(a) was taken, and the locations of the fiducial markers were extracted in the 3D image, using a centre of gravity operator, and in the 2D image by manually clicking on the fiducial location. Using corresponding 3D and 2D points, Tsai's algorithm was used to produce a set of gold standard intrinsic and extrinsic parameters. Using these parameters, the projection matrix (matrix $\mathbf{M}$ in equation 2.1) was formed and the 3D points projected to 2D pixel locations to form perfectly matching 3D and 2D points.

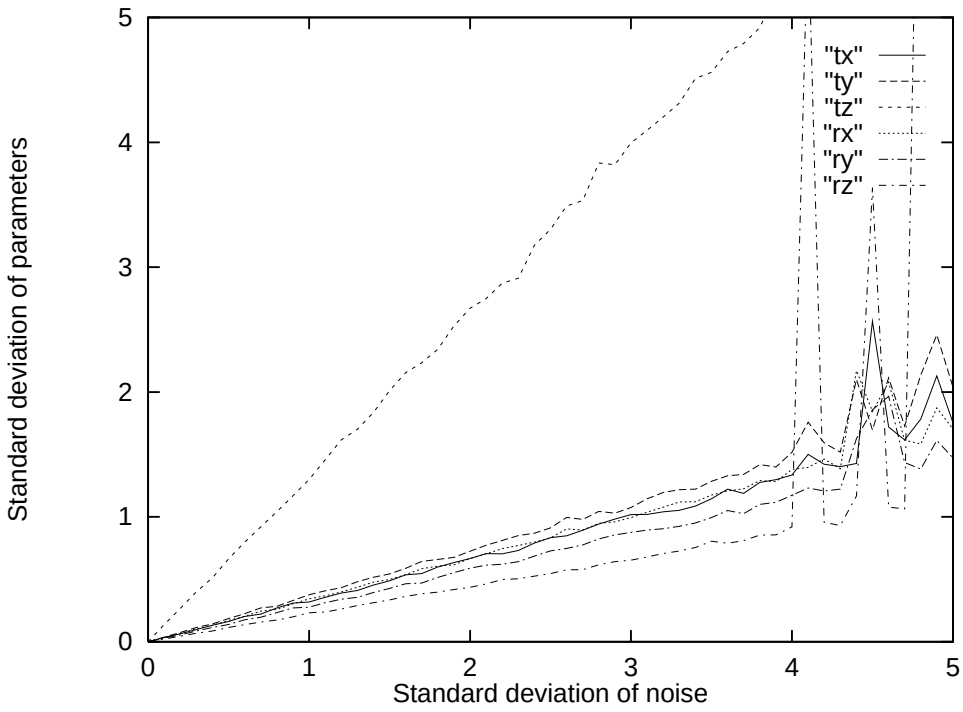
Random, zero mean, Gaussian noise with standard deviation $\sigma_n = 0.1, 0.2 \ldots 5.0$ was added to each of the 2D points, and a modified Levenberg-Marquardt [Press *et al.*, 1992] non-linear optimisation, was used to optimise the extrinsic parameters only[3]. This was repeated 1000 times, and the mean and standard deviation of the recovered extrinsic parameters calculated. In addition, the mean and standard deviation projection and 3D error was calculated for each noise level. This experiment was repeated, adding noise to just the 3D points. The purpose of this simulation was to determine what the effects of noise are on the accuracy of the recovered extrinsic parameters and error measures for a typical setup used in this chapter.

Subsequently, the image in figure 4.7(a) was taken, and the corresponding 2D and 3D point locations extracted as before. A leave-one-out test was used to determine the accuracy of the gold standard parameters. For a set of points, Tsai's algorithm used all but one of the corresponding 2D and 3D point pairs to calculate the gold standard parameters. The remaining point was used to calculate a projection error in millimetres. This was repeated for every combination of points.

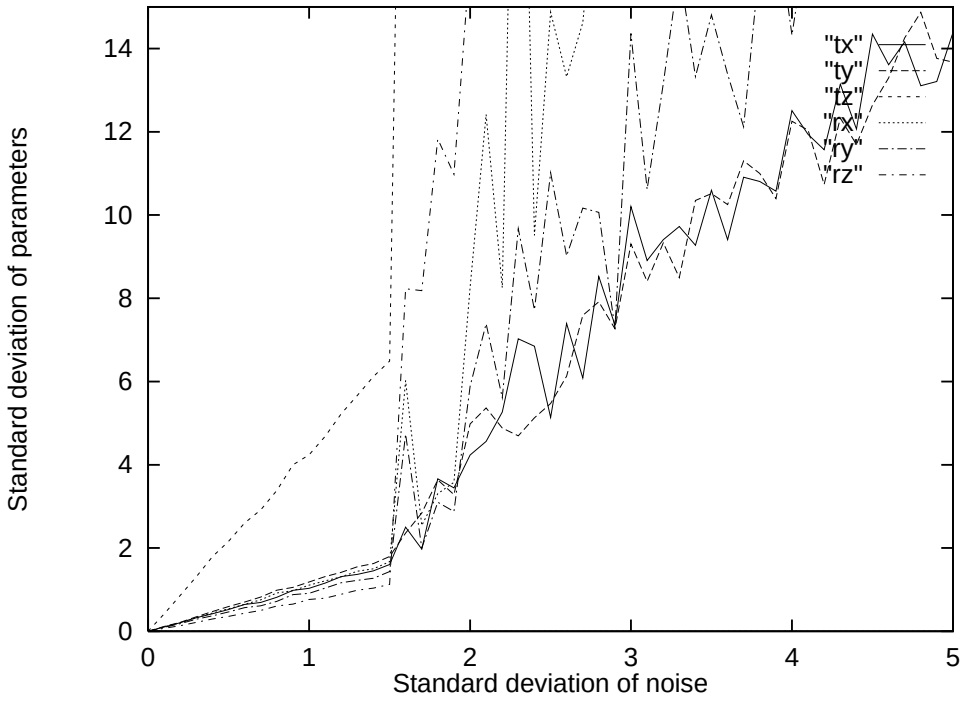
4.4.1.2 Results

The graph in figure 4.9(a) shows how the standard deviation of each parameter t_x, t_y, t_z r_x, r_y, r_z increases as noise is added to the 2D image points. Similarly, the graph in figure

[3]This is also part of freely available software `http://www.cs.cmu.edu/~rgw/`



(a)



(b)

Figure 4.9: (a) Variation in parameters $t_x \ldots r_z$ with noise added to the 2D points. (b) Variation in parameters $t_x \ldots r_z$ with noise added to the 3D points.

4.9(b) shows how the parameters vary when noise is added to the 3D points. It can be seen that the noise has greater effect when added to the 3D points. When the noise has a standard deviation ≥ 1.5 and the noise is added to the 3D points, Tsai's algorithm fails. For both 2D and 3D noise, the t_z parameter is the most effected. This means that for a given gold standard registration, t_z will be the least accurate parameter.

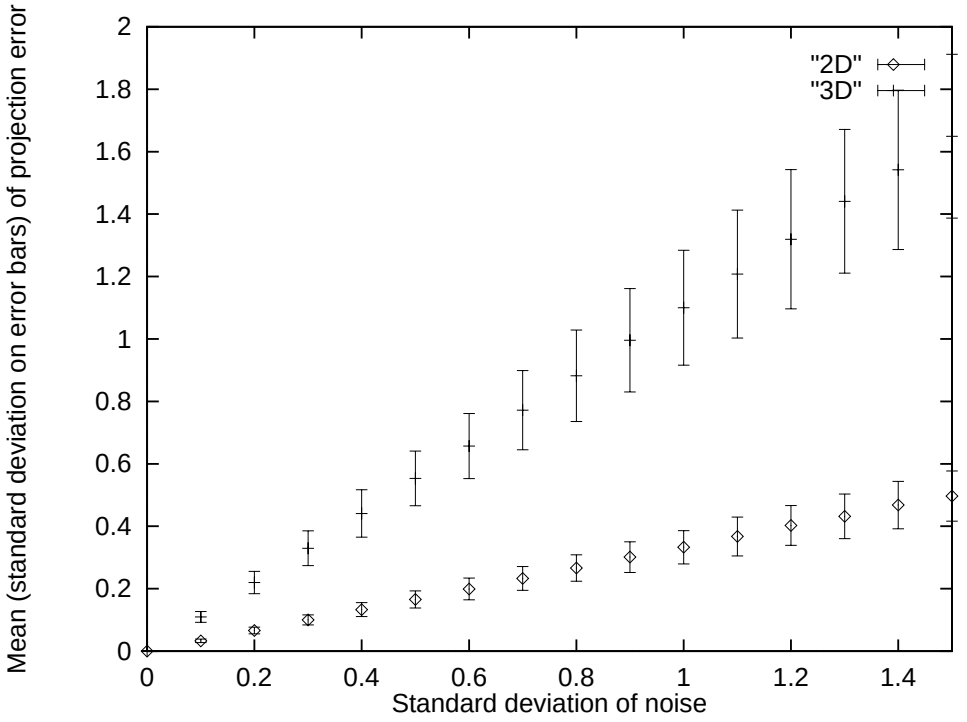
From these graphs in figure 4.9 (a) and (b) we can deduce that in order for all the parameters to have a standard deviation < 1, the standard deviation of the noise on the 2D pixels must be < 0.7 pixels, and the standard deviation of the noise in the 3D points must be < 0.2mm.

The graphs in figure 4.10(a)(b) show the mean and standard deviation (on errorbars) of (a) projection error and (b) the 3D error as the noise level is increased. The difference in projection error and 3D error is immediately apparent. Recall that the parameter most affected by noise is t_z. An error in the parameter t_z will cause a large 3D error but a much smaller projection error. This explains why graph (b) has larger errors.

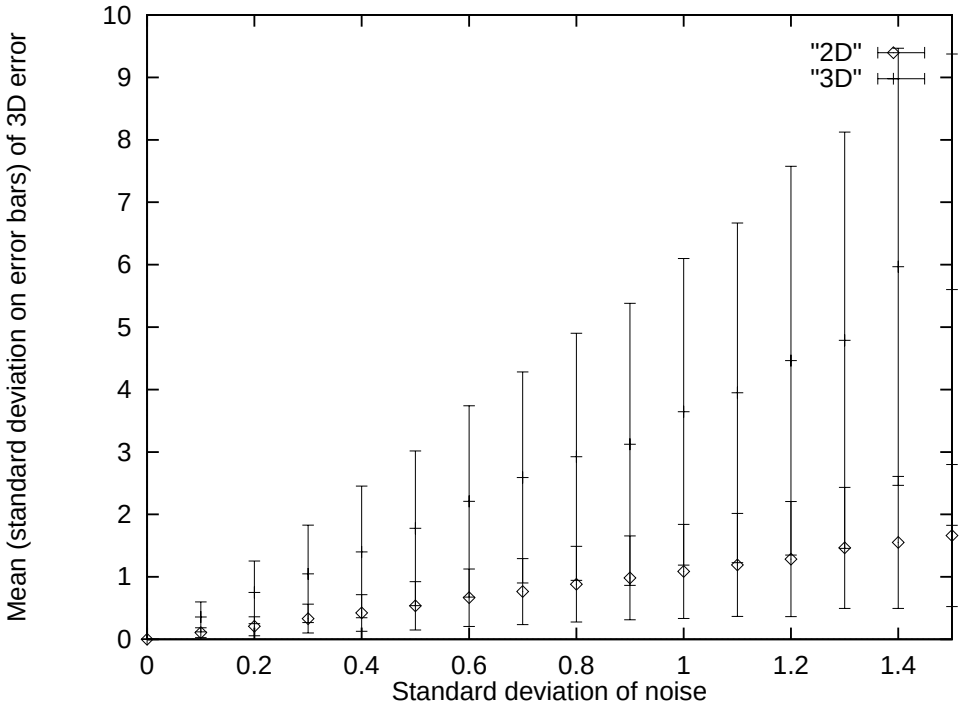
13 of the fiducials visible in the image in figure 4.7(a) were used for this experiment. Tsai's calibration was performed for every combination of 12 points from the 13 and the mean projection error was calculated as 0.25 mm. Referring to the graph in figure 4.10(a) and assuming that errors are caused entirely by noise on the 3D points suggests that the standard deviation of the noise is likely to be approximately 0.2 mm. This would suggest that, using the graph in figure 4.10(b) that the corresponding 3D error is approximately 0.75 mm. In addition, from the leave one out test, the standard deviation of the parameters $t_x, t_y, t_z, r_x, r_y, r_z$ was 0.07, 0.07, 0.19 mm and 0.06, 0.06, 0.05 degrees respectively.

4.4.1.3 Conclusions

It was concluded that the gold standard used throughout this chapter is of sufficient accuracy for the experiments that follow. The leave one out test revealed that the mean projection error was 0.25 mm and the corresponding mean 3D projection error was likely to be approximately 0.75 mm. For most clinical applications, an accuracy of approximately 1mm would be acceptable. With this gold standard, 3D errors of the order of 0.75 mm would be the best that can be reliably calculated with respect to this quality of gold standard.



(a)



(b)

Figure 4.10: (a) Mean and standard deviation (on error bars) of projection error as noise is added to 2D or 3D points. (b) Likewise for 3D error.

4.4.2 Testing Which Lighting Model To Use

4.4.2.1 Methods

A CT scan (Philips TOMOSCAN SR 7000 0.488 × 0.488 × 1.0 mm, 512 × 512 × 142 voxels) was taken of a plastic skull phantom. The video image shown in figure 4.7 was used for this experiment and was chosen to contain a combination of facial features and the side of the head. All the video images used in this chapter were taken using a Pulnix TM6EX camera with a 50 mm Cosmicar lens, grabbed with a Matrox Magic (RGB) frame grabber and converted to 768 × 576, 8-bit grey scale images. The small black spherical markers are 5mm painted aluminium ball bearings. These markers are used to produce a gold standard, and are not used by the algorithm to perform the registration. The markers were manually edited out from the CT scan, and video images using ANALYZE (Biomedical Imaging Resource, Mayo Foundation, Rochester, MN, USA.). Due to memory limitations for subsequent processing, the CT scan was smoothed using a Guassian filter of standard deviation $\sigma = 1$ mm and resampled using tri-linear interpolation [Press *et al.*, 1992] to half the resolution. Using the marching cubes algorithm in VTK [Schroeder *et al.*, 1997] an isosurface (surface model) was extracted. The isosurface value was the mean average of the air and phantom intensity values (700). Initially the surface model contained 528,548 triangles. This surface was decimated [Schroeder *et al.*, 1992] until it contained 88,384 triangles. The decimation was performed to reduce rendering time. The gold standard registration was calculated as in section 4.3.10. Misregistrations of size $\delta t = \pm 4$ mm and degrees were calculated and the registration algorithm as described in sections 4.3.8 to 4.3.7 was used to register the video image to the surface model.

This test was repeated for each of the three lighting methods described in section 4.3.7, i.e. moving, fixed, and optimising. This produced 64 results for each of the 3 lighting methods, giving 192 registrations. Each registration was assessed as a success or failure (as in section 4.3.9) and the mean projection error and 3D error was calculated using the successful registrations.

4.4.2.2 Results

Graphs (a) and (b) shown in figure 4.11 illustrate an important point. (a) and (b) show that for a moving light source i.e. a light source that moves with the rendering camera, the distribution of post-registration errors is bi-modal. The post-registration projection

Solution	Post-Registration Extrinsic Parameters					
	t_x	t_y	t_z	r_x	r_y	r_z
1 ($\delta t = -4$)	0.17 (0.19)	-1.07 (0.48)	-2.57 (0.36)	1.25 (0.42)	0.07 (0.20)	-0.07 (0.06)
2 ($\delta t = +4$)	-0.23 (0.08)	-0.01 (0.19)	4.41 (0.29)	0.51 (0.17)	-0.32 (0.09)	0.03 (0.05)

Table 4.1: Mean (standard deviation) registration parameters for a moving light source model.

Lighting Model	Projection Error (mm) Mean (StdDev)	3D Error (mm) Mean (StdDev)	Percentage Success
Moving	1.08 (0.39)	2.65 (0.46)	53
Fixed	1.92 (0.96)	3.40 (1.11)	44
Optimising	1.91 (0.77)	3.37 (0.74)	55

Table 4.2: Mean (Standard Deviation) projection and 3D error for each of the lighting source models described in section 4.3.7. $\delta t = \pm 4$ mm and degrees.

error is clustered above and below 1 mm. For the 3D error the results cluster around approximately (visually) 2.5 and also 4.5 mm.

This clustering corresponds to whether the initial misregistration of the parameter t_z was $+4$ or -4mm. Recall that t_z corresponds to translations along the rendering camera's optical axis. This is illustrated in table 4.1. If an increment of $\delta t = +4$ was added to t_z then the parameter t_z hardly changes during the registration, and the other five parameters all converge towards zero. When $\delta t = -4$ all the parameters converge towards zero, but the mean value for t_y, t_z and r_x are all > 1 i.e. inaccurately registered. In figure 4.11(c)-(f), it can be seen that in general performance is rather variable.

The fact that the algorithm does not correct for translations along the rendering cameras optical axis is illustrated by comparing projection and 3D errors. Table 4.2 shows the mean (standard deviation) projection and 3D errors for each lighting model described in section 4.3.7. The mean (standard deviation) projection and 3D errors of the starting position from the gold standard was 7.44 (0.92) mm and 9.36 (0.63) mm respectively.

Note that the arithmetic mean was calculated despite the fact that the distributions are bi-modal. In table 4.2 the mean and standard deviation is calculated using successful registrations. However the success rate is low (44% to 55%). A successful registration was one where all six of the parameters improved towards the gold standard (see section

4.3.9). The reason that so many registrations fail is because the translation parallel to the camera's optical axis started with an offset of $\delta t = \pm 4$ mm and degrees, and if the parameter t_z moves to $|t_z| > 4$ then the results would be classified as a failure even though the rendering may still look visually aligned, and have a low projection error. The 3D error is still large due to the fact that t_z is does not converge well towards the gold standard pose.

Using a Student's paired t-test on the successful registrations for the fixed and moving lighting model shows that the distribution of projection errors is significantly ($p < 0.0001$) different. Comparing the moving and optimising lighting model with a Student's t-test shows that the distribution of projection errors is significantly ($p < 0.0001$) different. The moving lighting model has a lower mean projection error and lower mean standard deviation and is therefore more accurate and robust than the optimising light source for this experiment.

4.4.2.3 Conclusions

It can be concluded that the moving lighting model performed significantly better than the optimising or fixed lighting model. This is to be expected as the video images are taken with a light source that is approximately aligned with the video camera and thus the rendering scene should mimic that. The algorithm is matching the overall shading pattern across the video image with the overall shading pattern across the rendered image. If the rendering light source used the fixed or optimising lighting model, then the shading pattern of the rendered image could vary throughout the course of the registration and eventually be very different from the video image. For the remainder of this chapter, the moving light source is used.

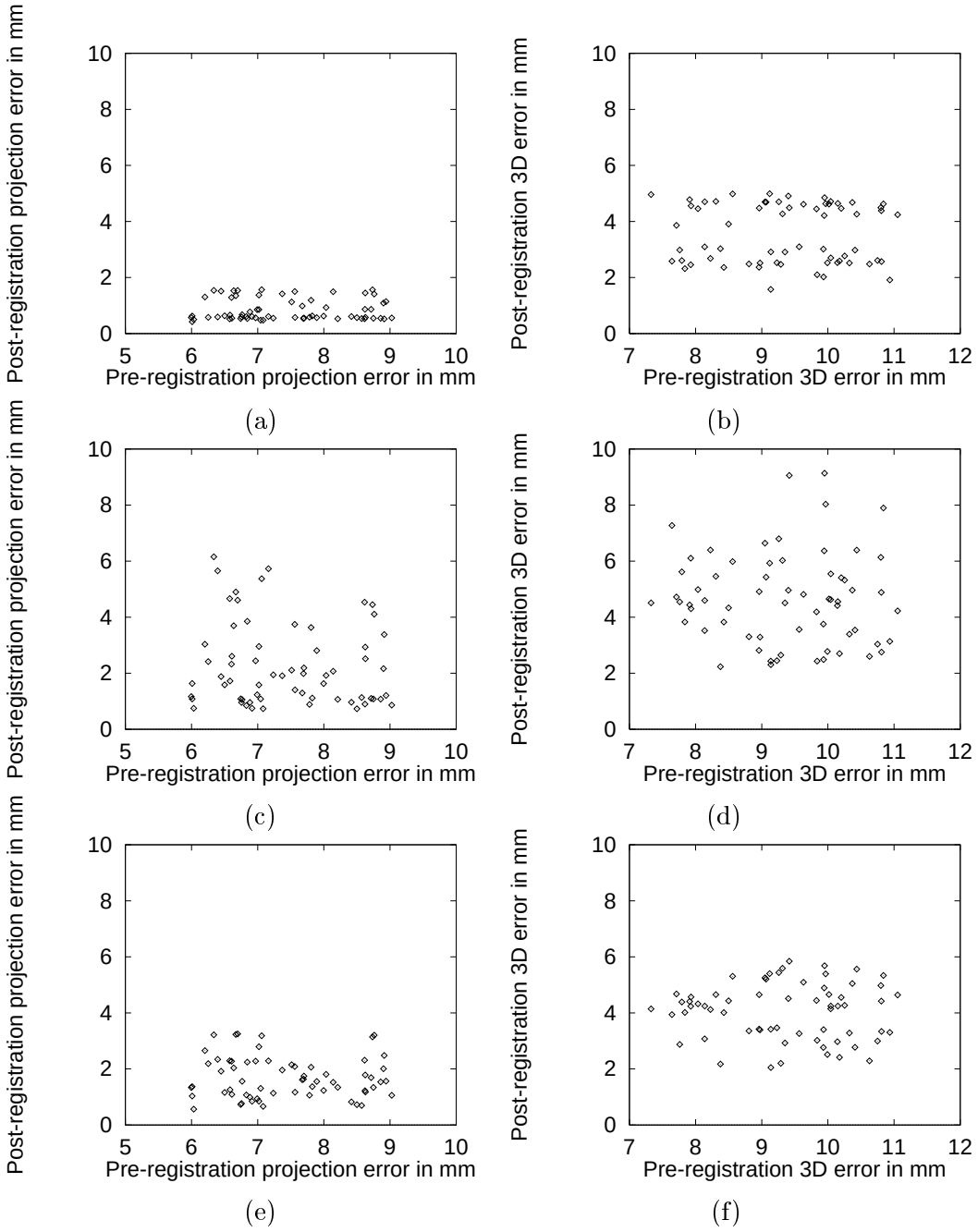


Figure 4.11: The left column shown projection error in mm and the right column shows 3D error, for (a)(b) a moving light model, (c)(d) a fixed light model and (e)(f) an optimising light model. Misregistration size δt was ± 4 mm and degrees.

4.4.3 Testing Accuracy, Robustness And Range Of Capture

4.4.3.1 Methods

The same surface model from the previous section (4.4.2) and both video images shown in figure 4.7 were used to test the accuracy, robustness and range of capture of this algorithm. The gold standard was calculated for both video images, using the method described in section 4.3.10 i.e. localising the fiducials and using Tsai's camera calibration to calculate the camera's intrinsic and extrinsic (t_x, t_y, t_z, r_x, r_y, and r_z) parameters. For both video images, and for each misregistration size of $\delta t = \pm 4, 8, 12, 16$ mm and degrees, the algorithm was used to register the surface model to the corresponding video image. The registrations were classified as a success if all the extrinsic parameters converged towards the gold standard values, and the mean and standard deviation, projection and 3D errors calculated for successful registrations. Thus, given two images, and for each image, 64 registrations for each value of δt gives a total of 512 registrations.

4.4.3.2 Results

Table 4.3 (a) shows the mean (standard deviation) projection and 3D errors for image 4.7(a), and likewise table 4.3 (b) shows the mean (standard deviation) projection and 3D errors for image 4.7(b). The accuracy of the registration is still poor. This can be seen in table 4.3(a) and (b). Table 4.3(a) has consistently lower projection and 3D errors than table 4.3(b), but varied success rates. The success rate has been included mainly for comparison with later chapters. From the previous section we know that the algorithm is failing to recover the parameter t_z which affects the success rate values, and is also confirmed by the fact that the 3D errors are very different to the projection errors for both tables.

4.4.3.3 Conclusions

It is difficult to make conclusions as it is already known that thus far the algorithm is not recovering the translational parameter t_z. So far, a maximum success rate of 83% was achieved, with $\delta t = \pm 8$ mm and degrees. For the two images tested, this value of δt gave mean projection errors of 1.25 mm and 3.86 mm. A mean projection error of 1.25 mm appears visually as a small error. The variation from 1.25 mm - 3.86 mm suggests quite markedly different performance from video image to video image.

	Pre-Registration		Post-Registration		
	Projection Error (mm)	3D Error (mm)	Projection Error (mm)	3D Error (mm)	Percentage
δt	Mean (StdDev)	Mean (StdDev)	Mean (StdDev)	Mean (StdDev)	Success
4	7.44 (0.92)	9.36 (0.92)	1.08 (0.39)	2.65 (0.46)	53
8	14.76 (1.79)	18.60 (1.98)	1.25 (0.55)	6.20 (1.22)	83
12	21.68 (2.81)	27.48 (3.11)	1.80 (1.89)	9.03 (2.29)	61
16	29.30 (3.92)	36.60 (4.48)	1.62 (0.57)	11.53 (2.66)	36

(a)

	Pre-Registration		Post-Registration		
	Projection Error (mm)	3D Error (mm)	Projection Error (mm)	3D Error (mm)	Percentage
δt	Mean (StdDev)	Mean (StdDev)	Mean (StdDev)	Mean (StdDev)	Success
4	7.51 (0.93)	9.38 (0.67)	2.26 (0.85)	3.86 (0.78)	53
8	14.92 (1.86)	18.65 (1.95)	3.86 (1.87)	8.17 (1.55)	73
12	22.40 (2.81)	28.04 (2.96)	5.12 (2.10)	10.95 (1.72)	66
16	29.48 (3.85)	36.88 (4.05)	6.17 (3.28)	14.53 (2.77)	52

(b)

Table 4.3: (a) Mean (standard deviation) projection errors, 3D errors and success rate for each δt, for the image shown in figure 4.7(a) and table (b), likewise for image 4.7(b).

4.4.4 Testing Performance With Changing Field Of View

4.4.4.1 Methods

The two video images used in the previous section and shown in figure 4.7 were also used to test performance with changing field of view. Each of the two images was masked to omit various parts of the image, as shown in figure 4.12. The same surface model as in section 4.4.2 was used. From previous experiments, the gold standard extrinsic parameters were known. For each video image, 64 misregistrations of size $\delta t = \pm 8$ mm and degrees were performed and the algorithm described in section 4.3.8 was used to register the model to the video image. Registrations were classified as success or failure as in section 4.4.2 and the mean projection and 3D errors calculated from the successful registrations for each image.

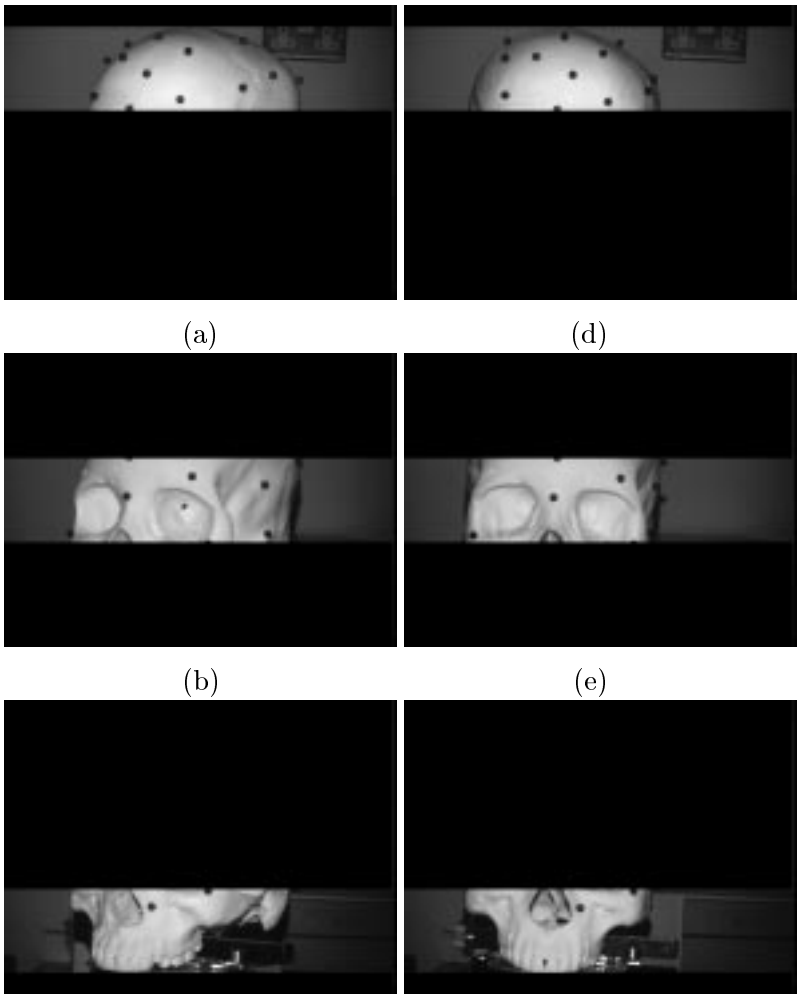


Figure 4.12: The two images of figure 4.7 were masked to omit information. (a)(b)(c) are masked versions of the image in figure 4.7(a), and (d)(e)(f) are masked versions of the image in figure 4.7(b).

4.4.4.2 Results

Table 4.4 shows the mean (standard deviation) projection and 3D errors for each image shown in figure 4.12. It can be seen that images 4.12 (a) and (d) have a reasonable success rate of 67% and 70% respectively whereas the remaining four images have an unacceptably low success rate of $\leq 14\%$. The images that produce successful registrations are two images of the top of the head, where in fact the skull phantom is relatively smooth and featureless.

Figure 4.13 shows the three masked versions of image 4.7(a), and the corresponding masked images of a rendering of the surface model, at the gold standard registration.

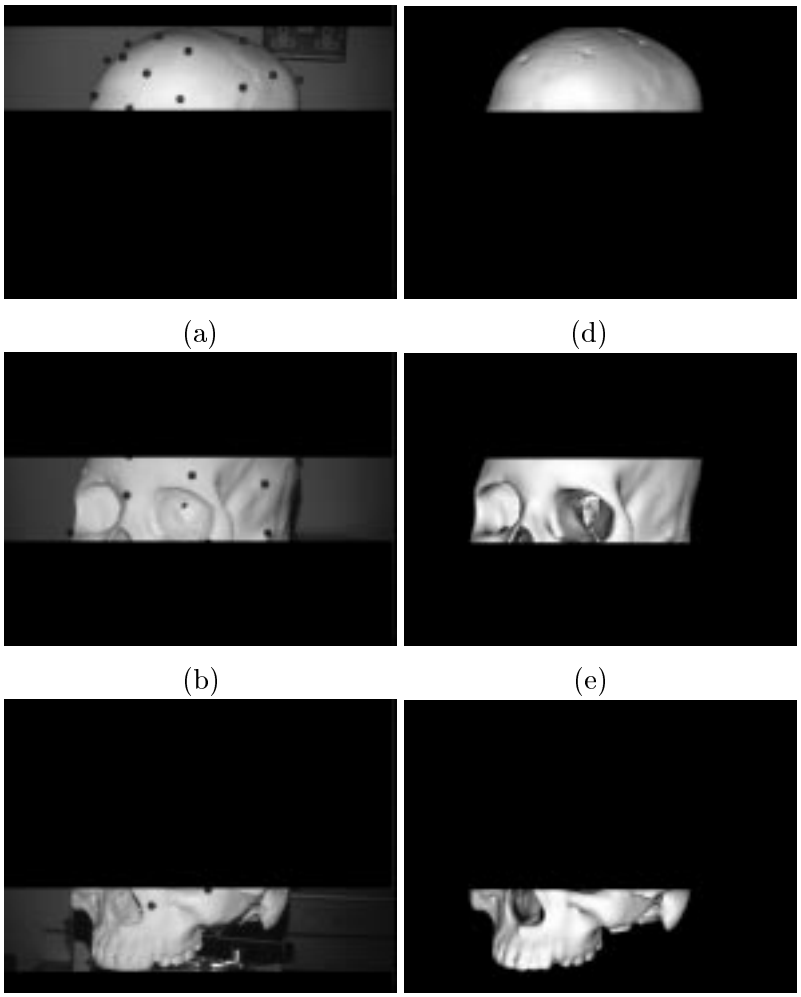


Figure 4.13: (a)(b) and (c) show image 4.7(a) masked in three different ways, and (d)(e) and (f) show similarly masked rendered images of the skull phantom at the gold standard registration.

Image	Projection Error (mm) Mean (StdDev)	3D Error (mm) Mean (StdDev)	Percentage Success
4.12(a)	1.95 (0.94)	7.32 (1.01)	67
4.12(b)	2.68 (1.13)	7.89 (0.75)	13
4.12(c)	6.53 (2.49)	10.49 (2.76)	14
4.12(d)	3.38 (1.07)	8.12 (0.86)	70
4.12(e)	5.21 (2.65)	9.67 (2.18)	14
4.12(f)	7.50 (1.27)	11.63 (1.62)	11

Table 4.4: Mean (Standard Deviation) projection and 3D error for each of the field of view images in figure 4.12. $\delta t = \pm 8$ mm.

Note that during registration the fiducial markers are also masked out of the video images. Comparing the video images with their corresponding rendered images reveals that for the middle and bottom pairs (b)(e) and (c)(f), the shading of the surface model in the rendering is remarkably different from the video image, whereas for the top pair (a)(d), the rendering is much more featureless, but the shading is most similar to the video image. Comparing table 4.4 with table 4.3(a) and (b) for $\delta t = \pm 8$ mm and degrees reveals that the mean accuracy and precision using the masked images 4.12(a) and (d) has significantly decreased.

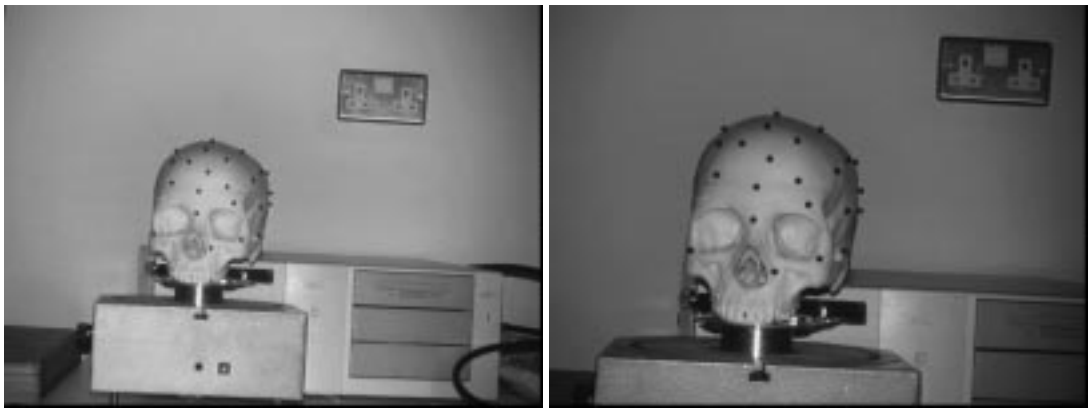
4.4.4.3 Conclusions

The masked images of the top of the skull phantom registered much more successfully than those of the middle or bottom of the skull phantom. However, comparing the rendering at the gold standard position (figure 4.13) with the video images shows that for the top pair of images (figure 4.13 (a) and (d)) the shading of the rendering is similar to the video image. For the middle pair of images, there are some marked differences. In the video image, figure 4.13 (b) the skull phantom's eye sockets are similar intensities, whereas in the equivalent rendering 4.13 (e) each eye socket has a different range of intensities. Mutual information was thought to be sufficiently flexible to match video images with rendered surface models, and for the experiments in sections 4.4.2 and 4.4.3 show that mutual information can successfully register video images and a rendered surface model. However the experiments of section 4.4.4 suggest that mutual information may not be sufficiently robust in cases where parts of the rendered image look markedly different from the video image.

4.4.5 Testing Performance With Changing Focal Length

4.4.5.1 Methods

Four video images were taken with different focal lengths, and are shown in figure 4.14. The same surface model as section 4.4.2 was used. The gold standard registrations were calculated as described in section 4.3.10. For each video image, misregistrations of size $\delta t = \pm 8$ mm and degrees were added to the gold standard extrinsic parameters and the algorithm used to register the surface model to the corresponding video image. Registrations were classified as success or failure as in section 4.4.2 and the mean projection and 3D errors calculated from the successful registrations for each image.



(a) (b)

(c) (d)

Figure 4.14: Four example video images used for focal length experiments. (See text section 4.4.5).

4.4.5.2 Results

Table 4.5 shows the mean (standard deviation) projection and 3D errors for each of the video images in figure 4.14. The four video images are a sequence where the skull phantom was moved closer to the video camera between each image grab. From table 4.5 it can be seen that image 4.14(b) gives the highest accuracy and precision for projection errors. Table 4.3(b) shows the registration performance for the image (b) in figure 4.7, which is an image of the front of the skull. The images used in this experiment to test the performance with respect to changing focal length are also images of the front face of the skull phantom. Thus comparing table 4.5 for image 4.14(c) shows comparable performance with table 4.3 for $\delta t = \pm 8$ mm and degrees.

Image	Pre-Registration		Post-Registration		Percentage
	Projection Error (mm) Mean (StdDev)	3D Error (mm) Mean (StdDev)	Projection Error (mm) Mean (StdDev)	3D Error (mm) Mean (StdDev)	Success
4.14(a)	14.97 (2.64)	18.60 (1.93)	2.82 (1.31)	7.05 (1.26)	50
4.14(b)	14.91 (1.88)	18.50 (1.98)	2.03 (1.15)	6.75 (1.24)	59
4.14(c)	14.88 (1.39)	18.79 (1.70)	4.13 (1.81)	7.91 (2.22)	57
4.14(d)	14.71 (1.73)	18.72 (2.06)	5.81 (1.08)	9.17 (1.42)	75

Table 4.5: Mean (Standard Deviation) projection and 3D error for each of the images shown in figure 4.14 where $\delta t = \pm 8$ mm.

4.4.5.3 Conclusions

It can be concluded that changing the focal length produces a significant change in the registration performance. The performance is affected by the distance of the skull phantom from the camera and the size and resolution of the images being used. The pixel size of the video image places a fundamental limit on the resolution of the video images. The image size i.e. 768×576 pixels affects the performance of the similarity measure. Mutual information is evaluated only for pairs of pixels that have an intensity greater than 0. Thus, the size of the rendering of the skull phantom in the rendered image will affect how much information gets placed in the histogram and is used to calculate mutual information. The search strategy calculates the derivative of mutual information with respect to each of the extrinsic camera parameters, and then tries to maximise the mutual information to achieve registration. However, the smoothness of the search space is determined by the image data, and also the step size taken of the optimisation algorithm. Consider the case where the optimisation strategy takes a ± 3 mm step size when calculating the derivative with respect to a translational parameter. If the focal length is long then this ± 3 mm shift may correspond to one pixel when projected onto the image plane. If the focal length is short, a ± 3mm shift may correspond to many pixels. Consequently the value of mutual information could change dramatically, leading to noisy estimates of derivatives and hence poor optimisation. The fact that the registrations using images in figure 4.14(c) and (d) show worse performance than the image in figure 4.14(b) suggests that the parameter space might in fact be less smooth when registering these images. For images (c) and (d) there may be more counts in the histogram, but the histogram could be varying rapidly with each pose tested, resulting in worse overall registration performance.

4.4.6 Comparison Of Similarity Measures

4.4.6.1 Methods

Both video images (a) and (b) in figure 4.7 were taken along with the same surface model, described in 4.4.2. The gold standard registrations were calculated as described in section 4.3.10. For each video image, misregistrations of size $\delta t = \pm 8$ mm and degrees were added to the gold standard extrinsic parameters and the algorithm used to register the surface model to the corresponding video image. Registrations were classified as success or failure as in section 4.4.2 and the mean projection and 3D errors calculated from the successful registrations for each image. The experiment was performed using mutual information (MI), normalised mutual information (NMI), normalised cross correlation (NCC) and gradient correlation (GC).

4.4.6.2 Results

The results for each similarity measure are shown in tables 4.6 and 4.7. The main observation is that with the similarity measures MI, NMI, NCC and GC, the algorithm produces smaller projection errors than 3D errors. This is again due to the algorithm failing to recover t_z, the translations parallel to the video camera's optical axis. The measure GC performs worst of those measure tested. GC measures the correlation of the video and rendered images vertical and horizontal gradients (see section 3.5.4.1). As the rendered image depicts a smooth surface model, using a smoothly varying Lambertian reflection model, it will have few clear edges, whereas the video image will have many spurious edges caused by noise. Gradient Correlation is an insufficient measure to align these types of gradient image. Of the remaining measures MI performs more accurately and robustly than NMI which similarly outperforms NCC. This confirms the choice of MI as a similarity measure for the previous experiments in sections 4.4.2 to 4.4.5. However, it is probably worthless comparing the measures to any further detail, as they all fail to recover t_z. A method for recovering all six parameters is developed in the next chapter. In addition, there is still a marked difference from image (a) to image (b).

4.4.6.3 Conclusions

The comparison of similarity measures confirms the initial choice of mutual information as the similarity measure of choice when compared with normalised mutual information, normalised cross correlation and gradient correlation.

Similarity Measure	Projection Error (mm) Mean (StdDev)	3D Error (mm) Mean (StdDev)	Percentage Success
(Pre-registration)	14.76 (1.79)	18.60 (1.98)	
MI	1.25 (0.55)	6.20 (1.22)	83
NMI	1.42 (0.46)	5.13 (1.66)	64
NCC	1.56 (0.22)	4.18 (1.44)	39
GC	4.37 (2.17)	6.56 (2.85)	39

Table 4.6: Mean (standard deviation) projection and 3D errors, for each similarity measure tested, for the image shown in figure 4.7(a).

Similarity Measure	Projection Error (mm) Mean (StdDev)	3D Error (mm) Mean (StdDev)	Percentage Success
(Pre-registration)	14.92 (1.86)	18.65 (1.95)	
MI	3.86 (1.87)	8.17 (1.55)	73
NMI	5.72 (2.59)	8.78 (2.29)	58
NCC	3.74 (1.10)	5.11 (1.33)	50
GC	8.22 (4.05)	10.68 (4.19)	30

Table 4.7: Mean (standard deviation) projection and 3D errors, for each similarity measure tested, for the image shown in figure 4.7(b).

4.5 Summary

From section 4.4.1.2 it was concluded that the gold standard is of sufficient accuracy for the experiments. The 3D error of the gold standard itself is of the order of 0.75 mm. The registration experiments showed that the moving lighting model performed significantly better than the optimising or fixed lighting models. However the main problem with this algorithm was that the algorithm does not recover the translation t_z which is the translational component parallel to the video cameras optical axis. The algorithm achieved a maximum success rate of 83% with the video image shown in figure 4.7(a), resulting in a mean (standard deviation) projection error of 1.25 (0.55) mm. The performance varies significantly with different video images, focal lengths and fields of view. Thus further work is required to improve the robustness. Mutual information was found to be the best similarity measure of those tested for the task of registering CT scans and video images of a skull phantom.

The registrations in this chapter took on average, approximately 10 minutes each. This is because of implementation, and methodology. The gradient ascent search strategy is known to take many iterations to reach an optimum. Furthermore, in order to improve the range of capture of the algorithm, a multi-resolution search strategy was implemented (see section 4.3.6). In this implementation, the lower resolution searches are not quicker to perform, as they are only used to increase robustness. In addition, the software implemented in VTK was designed to be modular and easily extendable, not fast. At this stage, the software is too slow for practical use. The speed of the software can be improved, and this topic is addressed in later chapters.

In this chapter, the mutual information of a single rendered and video image pair, optimised using a gradient ascent search strategy was found to be insufficient to register a video image with a 3D image. The first and foremost reason for this is that in all experiments the algorithm consistently failed to register with respect to translations parallel to the cameras optical axis. This is to be addressed in the next chapter. Other issues as to whether the similarity measure can be improved are developed throughout the thesis.

Chapter 5

Multiple View Registration

5.1 Introduction

The previous chapter described an algorithm to register a 3D medical image to a single video image. The main problem with the algorithm was found to be that the algorithm failed to recover translations along the optical axis of the camera. In this chapter the algorithm is extended to be able to take multiple optical images and register them all to a single 3D medical image. It is assumed that the transformation between each video camera coordinate system is known. This chapter represents an incremental change to the algorithm of the previous chapter through two simple methods for extending the similarity measure to cope with multiple views. An extension to multiple views has also been proposed by Leventon *et al.* [Leventon *et al.*, 1997] for images of a model car. His method is tested and compared with the two other novel extensions to the mutual information framework of the previous chapter.

This chapter describes the extension to multiple views in detail, and demonstrates the improvement in performance over the mono view algorithm. The experiments test (1) which of three multiple view methods are preferable, (2) registration robustness, (3) registration accuracy, (4) range of capture, (5) performance with changing field of view, (6) performance with changing focal length of the video camera and (7) a comparison of similarity measures for images of a volunteer's face. The multiple view results are then compared with the similar experiments of chapter 4.

5.2 Aim

The aim of this chapter is to investigate whether the mutual information of two or more video image and rendered image pairs, optimised using a gradient ascent search strategy is sufficient to register the video images to a 3D volume image.

5.3 Methods

The algorithm remains unchanged except for a novel modification of the similarity measure to enable the similarity of multiple video and rendered image pairs to be computed. This is described below.

5.3.1 Novel Extension To Multiple Views

Consider the case where N video images, denoted by $V_v, v = 1, 2, \ldots N$ are acquired of an object and where the transformation from one camera coordinate system to another is known. When this is the case, all N video images can be matched to one 3D image simultaneously. A set of rendered images can be produced, where the transformation between each virtual rendering camera is the same as between each video camera. The rendered images are denoted by $R_v, v = 1, 2, \ldots N$ and video image V_v should match rendered image R_v at registration. Let $\mathcal{V}_v$ denote a random variable describing the distribution of intensity values in image V_v and likewise $\mathcal{R}_v$ denote a random variable describing the distribution of intensity values in image R_v. Four different approaches to solving this registration problem are discussed below.

5.3.1.1 High Dimensional Histograms

The registration between the video and rendered images can be expressed as

$$\arg\max\ [I(\mathcal{V}_1; \ldots; \mathcal{V}_N; \mathcal{R}_1; \ldots; \mathcal{R}_N)] \tag{5.1}$$

This formulation assumes that each image is represented by a separate random variable, and the quantity to be maximised is information in one image that is well explained by all the other images. However, with N video and rendered images pairs, a 2N-dimensional joint histogram will be required to evaluate the underlying probability distributions. As N increases, so does computational cost, and the estimates become less reliable as the high dimensional histogram becomes more sparsely populated. In the general case, this method will be prohibitively expensive, and is not considered further in this thesis.

5.3.1.2 Multiple 2D Histograms

An alternative is to use

$$\arg\max \left[I(\mathcal{V}_1; \mathcal{R}_1) + I(\mathcal{V}_2; \mathcal{R}_2) + \ldots + I(\mathcal{V}_N; \mathcal{R}_N) \right] \tag{5.2}$$

where the mutual information between each pair of images is added so that the quantity to be maximised is the sum of the mutual information of each rendered and video image pair. This means that N, 2D joint histograms will be required. This is computationally less expensive, and provided that each histogram is well populated, the estimates of $I(\mathcal{V}_v; \mathcal{R}_v)$ will be reliable, and hence an algorithm based around equation (5.2) should perform well. This is called "adding the mutual information for each rendered and video image pair" and will be denoted with the single word 'adding'.

5.3.1.3 Single 2D Histogram

Alternatively, if it can be assumed that the relationship between rendered image intensities and video image intensities is the same across all pairs of rendered and video images, then the intensities from all video and rendered images can be combined into a single joint probability distribution of intensities, which characterises the relationship between the video and rendered images. The mutual information can then be calculated from the joint probability distribution, and will be maximised as alignment is reached. This is called "combining all the information into a single histogram", and is denoted by 'combining'.

5.3.1.4 Alternating Between Video Images

Finally, we call Leventon's method alternating between video images [Leventon *et al.*, 1997]. This is denoted by 'alternating'. For each video image in turn, the algorithm computes the gradient of mutual information with respect to the six transformation parameters, and makes a single step in that direction. The images in this chapter are of a plastic skull phantom, or a volunteer.

5.4 Experiments

The first part of this chapter described the necessary modification to the algorithm described in chapter 4 to enable multiple video images to be registered simultaneously to a single 3D volume. This section tests the performance of the modified algorithm as follows.

- **Testing Which Multiple View Method To Use**. For misregistrations of size $\delta t = \pm 8$ mm and degrees, and for each of the three multiple view methods adding, combining and alternating, described above, the algorithm was used to register two video images to a CT scan of a skull phantom. See section 5.4.1
- **Testing What Angular Disparity To Use**. Using a pair of video images, and the two best multiple view methods, adding and combining, the algorithm was tested with respect to the angular separation between views. Angles tested were $5, 10, 30, 50, 70$ and 90 degrees and misregistration size was $\delta t = \pm 8$ mm and degrees. See section 5.4.2.
- **Testing How Many Video Views To Use.** Using combinations of $2, 3, 4$ and 5 video views, the combining multiple view method was tested to determine a limit on the necessary number of views. See section 5.4.3.
- **Comparison With Mono View Algorithm.** The combining multiple view method was then compared with the mono view method, for accuracy, robustness, range of capture, performance with respect to field of view and focal length. See sections 5.4.4 to 5.4.6.2.
- **Registration For The Stereo Operating Microscope.** The combining multiple view method was then applied to register multiple video taken from an operating microscope. This was presented in [Clarkson *et al.*, 1999a]. See section 5.4.7.
- **Comparison Of Similarity Measures.** For misregistrations of size $\delta t = \pm 8$ mm and degrees, the similarity measures mutual information (MI), normalised mutual information (NMI), normalised cross correlation (NCC) and gradient correlation (GC) were used to register four video images of a volunteer to a reconstructed surface model. See section 5.4.9.

(a) (b)

Figure 5.1: Video images of the skull phantom used for the multiple view experiments.

5.4.1 Testing Which Multiple View Method To Use

5.4.1.1 Methods

The surface model of section 4.4.2 and two video images were taken of the plastic skull phantom. See figure 5.1. The two views of the skull phantom depicted in figure 5.1 differ by a 45 degree rotation of the skull. The gold standard registration for each view was calculated by localising the fiducials and using Tsai's algorithm [Tsai, 1987] as described in section 4.3.10. This yields a set of intrinsic and extrinsic camera parameters for each video camera. The gradient ascent search strategy remains the same as in the previous chapter. The algorithm was tested for misregistration sizes of $\delta t = \pm 8$ mm and degrees. Each registration was classified as a success or failure. A successful registration is one where none of the extrinsic parameters moves further away from the known gold standard values than when it started. For each successful registration the projection error and 3D error were calculated. The mean and standard deviation projection and 3D errors were calculated for each multiple view method.

5.4.1.2 Results

Table 5.1 shows the mean (standard deviation) projection and 3D errors for each multiple view method tested. It can be seen that in this experiment the alternating method is less robust, less accurate and less precise than the combining or adding method. The combining and adding methods produce similar results. Recall that with the mono view algorithm, the mean 3D error was usually of the order of the misregistration size δt e.g. 8mm, as the algorithm failed to recover the offset t_z along the camera's optical axis. In

Multiple View Method	Projection Error (mm) Mean (StdDev)	3D Error (mm) Mean (StdDev)	Percentage Success
Pre-registration	14.97 (1.86)	18.69 (1.98)	
Combining	2.56 (0.15)	3.82 (0.20)	100
Adding	2.45 (0.22)	3.62 (0.27)	100
Alternating	3.53 (1.13)	5.37 (1.22)	73

Table 5.1: Mean (Standard Deviation) projection and 3D error for each of the different multiple view methods described in section 5.4.1. $\delta t = \pm 8$ mm and degrees.

table 5.1 we can see that the mean 3D error has improved when compared to the mono view algorithm. From these tests, the alternating method was rejected, as the other two methods, adding and combining performed significantly better.

5.4.1.3 Conclusions

When comparing the different multiple view registration methods it can be seen that adding the mutual information of each rendered and video image pair does not give a significant performance increase over combining all the information into one histogram. This is to be expected as each view contains a view of the same surface skull phantom, and the light source is known to be fixed relative to the camera. Thus the relationship between video and rendered image intensities is likely to be similar across all views. There might be other lighting geometries for which adding the mutual information from each view might prove superior to the combining case. However the alternating method performed much worse than the adding or combining method. For the adding and combining method, the algorithm would find the maximum of mutual information and terminate when the gradient ascent search could find no better set of parameters. However, for the alternating method, the algorithm would approach the maximum, and then fail to converge. The alternating method would calculate the best step to take for a given view, and take that step, which would make the registration improve with respect to one view. It would then move onto the next view, whereupon it would take another step. However, improving the parameters with respect to one view seemed to make the current estimate of the parameters worse with respect to another view. Thus the alternating algorithm seemed to oscillate between different views. A maximum number of iterations had to be set to force the algorithm to terminate. Furthermore, the alternating algorithm was in general more likely to fail than either the adding or combining method.

Images	Projection Error (mm) Mean (StdDev)	3D Error (mm) Mean (StdDev)	Percentage Success
(0,5)	6.72 (1.75)	7.90 (2.18)	85
(0,10)	5.26 (1.47)	6.23 (1.77)	97
(0,30)	2.70 (0.65)	3.07 (0.82)	100
(0,50)	2.46 (0.18)	3.50 (0.25)	100
(0,70)	3.70 (0.43)	5.16 (0.38)	97
(0,90)	-	-	0

(a)

Images	Projection Error (mm) Mean (StdDev)	3D Error (mm) Mean (StdDev)	Percentage Success
(0,5)	6.22 (1.90)	7.29 (2.31)	92
(0,10)	3.98 (1.23)	4.81 (1.72)	100
(0,30)	1.39 (0.15)	1.73 (0.28)	100
(0,50)	2.07 (0.14)	3.21 (0.19)	100
(0,70)	4.01 (0.33)	5.55 (0.29)	97
(0,90)	-	-	0

(b)

Table 5.2: Mean (standard deviation) projection errors, 3D errors and success rate for each angle of disparity, for each multiple view method (a) adding and (b) combining $\delta t = \pm 8$ mm and degrees. Video images are shown in figure 5.1.

5.4.2 Testing What Angular Disparity To Use

5.4.2.1 Methods

The surface model from section 4.4.2 was used with a series of video images that represented a rotation of up to 90 degrees. The first image was that shown in figure 5.1(a) and is labelled as image 0 as it is the reference image. Further video images were taken where the skull was rotated by 5, 10, 30, 50, 70 and 90 degrees. Each image was labelled according to its angle of rotation from the reference image. The gold standard registration for each view was calculated by localising the fiducials and using Tsai's algorithm [Tsai, 1987] as described in section 4.3.10. Registrations were performed using pairs of video image simultaneously. Pairs of images tested were images $(0, 5), (0, 10), (0, 30), (0, 50), (0, 70)$ and $(0, 90)$. Registration to each pair of images and for the two multiple view meth-

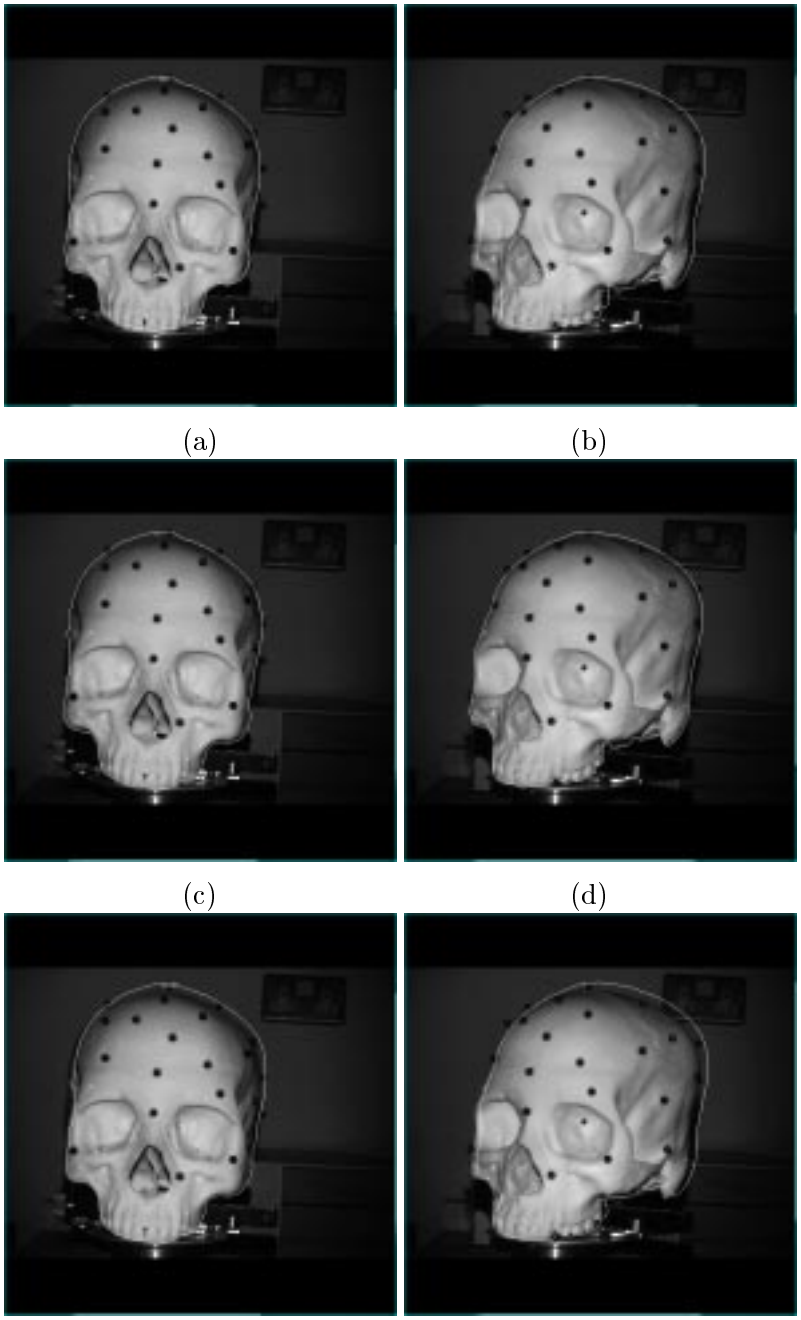
ods combining and adding were tested using a misregistration size of $\delta t = \pm 8$ mm and degrees. After each successful registration, the projection and 3D error was calculated. The mean and standard deviation projection and 3D errors were calculated for each pair of images.

5.4.2.2 Results

The mean (standard deviation) projection and 3D errors for each pair of images can be found in table 5.2. Table 5.2(a) shows the results for cases in which the information from each rendered and video image pair is added and table 5.2(b) shows the errors for cases in which the information from each rendered and video image pair is combined.

It can be seen that the combining method has a higher success rate for the pairs of images (0,5) and (0,10). Both methods completely fail for the image pair (0,90). From this experiment, the (0,30) pair of images and the combining method has the lowest mean projection and 3D errors i.e. 1.39 (0.15) and 1.73 (0.28) mm respectively with a 100 % success rate. Table 4.3(a) showed the mean (standard deviation) of the projection and 3D errors for the mono case was 1.25 (0.55) and 6.20 (1.22) mm respectively. Table 4.3(b) showed the mean (standard deviation) of the projection and 3D errors for the second mono case was 3.86 (1.87) and 8.17 (1.55) mm respectively. The (0,30) pair of video images and the combining method therefore has a lower 3D error than the mono cases, and a comparable projection error.

Figure 5.2 illustrates mono and multiple view registration results. The top row is a mono view registration result. The outline of the rendered surface model is displayed as a white line, overlayed onto the video image. The image in figure 4.7(a) shows a registration result for the mono view algorithm of the previous chapter. The recovered extrinsic parameters of this registration are shown in table 5.3 in the row labelled 'Mono'. The main registration error is along the optical axis of the camera. In image (a), which was the image used for the registration the rendered overlay appears well aligned. The image in figure 5.2(b) is another 'overlay image' from a camera that is rotated by 30 degrees, from image (a) but showing the registration result produced when registering to image (a). The errors in the mono view registration are apparent as the rendering appears shifted to the right relative to the skull in the video image.



(a) (b)

(c) (d)

(e) (f)

Figure 5.2: Registration results using (a) and (b), mono video image, (c) and (d) two images separated by 30 degrees, (e) and (f) two images separated by 70 degrees (see text section 5.4.2.2).

Solution	Post-Registration Extrinsic Parameters					
	t_x	t_y	t_z	r_x	r_y	r_z
Gold Standard	0	0	0	0	0	0
Mono	-0.09	-0.13	8.20	0.57	-0.31	-0.02
Stereo, 30 Degrees	0.53	0.89	-0.34	0.46	0.15	-0.13
Stereo, 70 Degrees	2.99	-1.19	3.35	2.34	0.01	-2.03

Table 5.3: Examples of post-registration extrinsic parameters for mono and stereo results. See section 5.4.2.2.

Images (c) and (d) are results from a stereo view registration. The angle of disparity between the views is 30 degrees. Both views are accurately aligned giving lower projection and 3D errors than the mono view algorithm. Images (e) and (f) are also results from a stereo view registration. The angle of disparity between the view is 70 degrees. Neither view is accurately aligned. The actual registration results are shown in table 5.3. The gold standard position is represented by 0 for all $t_x \ldots r_z$. This table shows that the mono view algorithm fails to recover t_z. The stereo algorithm with 30 degrees disparity recovers all parameters close to their gold standard values, and the stereo algorithm with 70 degrees recovers all parameters, but not very accurately.

5.4.2.3 Conclusions

The experiments testing what angle to use between two video views (section 5.4.2) showed that an angle difference of 30 degrees gave the best performance. With the angle less that 30, the errors increased and became similar to the mono view performance. With an angle larger than 30 degrees, the errors also increased as registration performance worsened. At a separation of 90 degrees the algorithm failed completely. This could be due to the search space becoming nearly flat, and the search strategy failing. As each new pose was tested, a change in the parameters will produce an improvement in the similarity measure with respect to a single view, and possibly a similar decrease in similarity with respect to another view. If these changes are equal and opposite when the angle of separation approaches 90 degrees then the search space becomes flatter, and the search strategy is more likely to fail.

Set of Images	Projection Error (mm) Mean (StdDev)	3D Error (mm) Mean (StdDev)	Percentage Success
(0,10)	3.98 (1.23)	4.81 (1.72)	100
(0,10,30)	1.44 (0.22)	1.93 (0.32)	100
(0,10,30,50)	2.16 (0.36)	3.08 (0.36)	100
(0,10,30,50,70)	3.68 (0.43)	5.52 (0.58)	94
(0,10,30,50,70,90)	-	-	0

Table 5.4: Mean (standard deviation) projection and 3D error for each number of images. $\delta t = \pm 8$ mm and degrees.

5.4.3 Testing How Many Video Views To Use

5.4.3.1 Methods

The surface model from section 4.4.2 was used and the same video images from section 5.4.2. As before registrations were performed with multiple images, except the combinations were $(0, 10), (0, 10, 30), (0, 10, 30, 50), (0, 10, 30, 50, 70)$ and $(0, 10, 30, 50, 70, 90)$. For each of these five groupings, 64 registrations with $\delta t = \pm 8$ mm and degrees were performed. As the adding and combining multiple view methods had performed similarly, the combining method was used for the remainder of the chapter. Each registration was classified as a success or failure as before and the mean and standard deviation projection and 3D errors for each set of images were calculated from the successful registrations.

5.4.3.2 Results

Table 5.4 shows the mean (standard deviation) projection and 3D errors for each group of images. In this experiment the combining method was used and $\delta t = \pm 8$ mm and degrees. It can be seen that the (0,10,30) set of images results in a mean projection and 3D error of 1.44 (0.22) and 1.93 (0.32) respectively. Note that the set (0,10,30,50,70,90) completely failed for all tests. The 3D errors are in general better than the mono view case, but the projection errors are not necessarily so. Table 5.5 shows that for different combinations of images, a different mean set of parameters is recovered.

5.4.3.3 Conclusions

The experiments testing the required number of images (section 5.4.3) produced similar results to the experiments testing what angle to use between video views (section 5.4.2).

Solution	Post-Registration Extrinsic Parameters					
	t_x	t_y	t_z	r_x	r_y	r_z
Gold Standard	0	0	0	0	0	0
(0,10,30)	0.60	0.32	0.63	1.06	0.24	-0.17
(0,10,30,50,70)	2.20	-2.47	2.34	4.87	2.39	-2.18

Table 5.5: Mean registration parameters using different image combinations.

The experiment only tested a few combinations, i.e. 1 combination of 2,3,4,5 and 6 images, and the errors increased as larger number of images with larger angle separation from the reference image (image 0) were used. From this experiment and the previous experiment it was concluded that the multiple view algorithm appears to be precisely recovering a solution that is offset from the gold standard i.e. there is a systematic error, and the error depends on the number of images and their distribution. It is impossible to test every combination of number of images and their distribution. However, even with this set of images, two points are clear. (1) Mutual information can be used to register accurately, but it does not always work well. (2) If mutual information does not work well, it is hard to determine why. This is a general problem for other registration applications.

5.4.4 Testing Accuracy, Robustness And Range Of Capture

5.4.4.1 Methods

The same surface model was taken, and the pair of video images (0,30) from section 5.4.2, as these two video image produced the most accurate registrations. The gold standard registration for each view was calculated by localising the fiducials and using Tsai's algorithm [Tsai, 1987] as described in section 4.3.10. For misregistration sizes of $\delta t = \pm 4, 8, 12$ and 16, the algorithm registered the surface model to the video images. For successful registrations the mean and standard deviation projection and 3D errors were calculated for each δt.

5.4.4.2 Results

The mean and standard deviation projection and 3D errors for each value of δt are shown in table 5.6. This table should be compared with table 4.3(b) in section 4.4.3.2. Using two views, the 3D errors are much better throughout a range of misregistration

	Pre-Registration		Post-Registration		
	Projection Error (mm)	3D Error (mm)	Projection Error (mm)	3D Error (mm)	Percentage
δt	Mean (StdDev)	Mean (StdDev)	Mean (StdDev)	Mean (StdDev)	Success
4	7.50 (0.93)	9.36 (0.99)	1.36 (0.18)	1.60 (0.27)	100
8	14.97 (1.86)	18.69 (1.98)	1.39 (0.15)	1.73 (0.28)	100
12	22.45 (2.76)	28.18 (2.90)	1.34 (0.17)	1.69 (0.31)	92
16	29.95 (3.62)	38.26 (3.40)	1.42 (0.18)	1.80 (0.33)	73

Table 5.6: (a) Mean (standard deviation) projection errors, 3D errors and success rate for each δt, for the images shown in figure 5.1.

sizes δt than with a mono view. For example in table 4.3(a), for $\delta t = \pm 12$, the mean (standard deviation) projection and 3D errors are 1.80 (1.89) and 9.03 (2.29) respectively for the mono view case. This compares with 1.34 (0.17) and 1.69 (0.31) in table 5.6. Furthermore, the precision is better with two views than with one. With two views, the success rate is 92% and 73% for $\delta t = \pm 12$ and 16 mm and degrees compared to 66% and 52% for one view. For misregistration sizes of $\delta t = \pm 12$ and 16 mm and degrees, the failed solutions did not cluster around any fixed point or local maxima. The failed solutions appeared randomly distributed.

5.4.4.3 Conclusions

The experiments testing the range of capture of the multiple view algorithm (section 5.4.4) showed that using multiple views noticeably increases the range of capture and decreases the 3D errors. Comparing the mono view results in table 4.3 with the multiple view results in table 5.6 shows that the mono view algorithm had a success rate of 36% to 83%, compared with the multiple view performance which gave 73% to 100%. The mono view algorithm failed to recover from misregistrations along the optical axis of the video camera, which gave 3D errors of 2.65 mm to 14.53 mm compared with the multiple view algorithm which gave 3D errors of 1.60 to 1.80 mm for comparable images.

Image	Projection Error (mm) Mean (StdDev)	3D Error (mm) Mean (StdDev)	Percentage Success
4.12(a)(d)	2.11 (0.30)	2.40 (0.35)	100
4.12(b)(e)	2.91 (2.74)	4.12 (3.49)	11
4.12(c)(f)	5.18 (3.14)	7.15 (3.72)	44

Table 5.7: Mean (standard deviation) projection and 3D error for pairs of the field of view images in figure 4.12. $\delta t = 8$ mm.

5.4.5 Testing Performance With Changing Field Of View

5.4.5.1 Methods

The experiments in section 4.4.4 measured the mean and standard deviation projection and 3D error for each of the six images (a) - (f) in figure 4.12. These six images were constructed from a stereo pair of images. The images (a) and (b) were a pair of images where the field of view was masked so that only the top of the image was visible. Image (c) and (d) were a pair where only the middle was visible, and in images (e) and (f) only the bottom was visible. For each pair of images (a)(b), (c)(d) and (e)(f) from figure 4.12, and for misregistration size $\delta t = \pm 8$ mm and degrees, and using the combining multiple view method, the algorithm was used to register the pairs of video images to the surface model. The mean and standard deviation projection and 3D errors were calculated for the successful registrations.

5.4.5.2 Results

In section 4.4.4, each of the images in figure 4.12 were used separately to test the mono view registration performance. The mono view results can be found in table 4.4. Table 5.7 shows the mean (standard deviation) projection and 3D error and success rate for each pair of images. The pairs of images correspond to the top, middle and bottom pairs in figure 4.4. Thus comparing the multiple view method with the mono view method, it can be seen that in general, the registration still only works successfully for the top pair of images (a)(d) in figure 4.4 and this success rate has risen from 67% or 70% for the mono case to 100% for this multiple view experiment. For the middle and bottom pairs, the success rate is only 11% and 44% respectively. The second and third row of results in table 5.7 show that the mean and standard deviation projection and 3D errors are high at 2.91(2.74) and 7.15(3.72) mm.

5.4.5.3 Conclusions

In the previous chapter, the images in figure 4.12(b),(c),(e) and (f) did not register well. Table 5.7 shows that combining these images into pairs (b)(e) and (c)(f) has not made much improvement. This suggests further work is needed in finding for instance, a better similarity measure or search strategy.

5.4.6 Testing Performance With Changing Focal Length

5.4.6.1 Methods

Figure 5.3 shows 8 images, labelled (a) - (h), which were used to test the performance of the multiple view algorithm with respect to changing focal length of the video cameras. The images (a),(c),(e) and (g) in figure 5.3 are the same as the images(a)(b)(c) and (d) in figure 4.4.5. Four pairs of images were taken i.e. (a)(b), (c)(d), (e)(f) and (g)(h) from figure 5.3. Each pair of images (a)(b), ... (g)(h) have the same focal length, but the skull was rotated by 45 degrees. The gold standard registration for each view was calculated by localising the fiducials and using Tsai's algorithm [Tsai, 1987] as described in section 4.3.10. For misregistration size $\delta t = \pm 8$ mm and degrees, the combining multiple view method was used to register the pairs of images to the surface model. For all successful registrations the mean and standard deviation projection and 3D errors were calculated for each pair of images.

5.4.6.2 Results

Table 5.8 shows the mean and standard deviation projection and 3D errors and success rate for each of the pairs of video images shown in figure 5.3. Comparing table 5.8 with table 4.5 it can be seen that the multiple view experiments are more successful with success rates of 95% or above, compared to a success rate of 50% - 75% for the mono view experiments. Looking at the projection and 3D errors in tables 5.8 for the multiple view experiment, and table 5.8, it can be seen that for image pair (g)(h) and image (d) in figure 4.14 the errors are still high, i.e. ≈ 9 mm. For the image pairs (a)(b), (c)(d) and (e)(f), the 3D errors are better than the mono equivalent. For image pairs (a)(b) and (e)(f) using multiple views, projection errors are higher than in the mono case. It is only for image pair (c)(d) in figure 5.3 that both the projection and 3D error show improved performance over the mono view case, i.e. image (b) in figure 4.14.

Image	Pre-Registration		Post-Registration		Percentage
	Projection Error (mm) Mean (StdDev)	3D Error (mm) Mean (StdDev)	Projection Error (mm) Mean (StdDev)	3D Error (mm) Mean (StdDev)	Success
4.14(a)(b)	14.97 (1.86)	18.69 (1.98)	4.41 (0.29)	4.67 (0.28)	100
4.14(c)(d)	14.97 (1.86)	18.69 (1.98)	1.66 (0.01)	1.90 (0.08)	100
4.14(e)(f)	14.97 (1.86)	18.69 (1.98)	4.43 (0.57)	5.22 (0.55)	100
4.14(g)(h)	14.98 (6.26)	18.77 (1.83)	4.58 (0.54)	9.79 (0.77)	95

Table 5.8: Mean (standard deviation) projection and 3D error for each of the focal length images in figure $\delta t = \pm 8$ mm.

5.4.6.3 Conclusions

Table 5.8 shows that extending the algorithm to incorporate multiple views has not solved the problem that with different focal lengths, the registration performance is again variable.

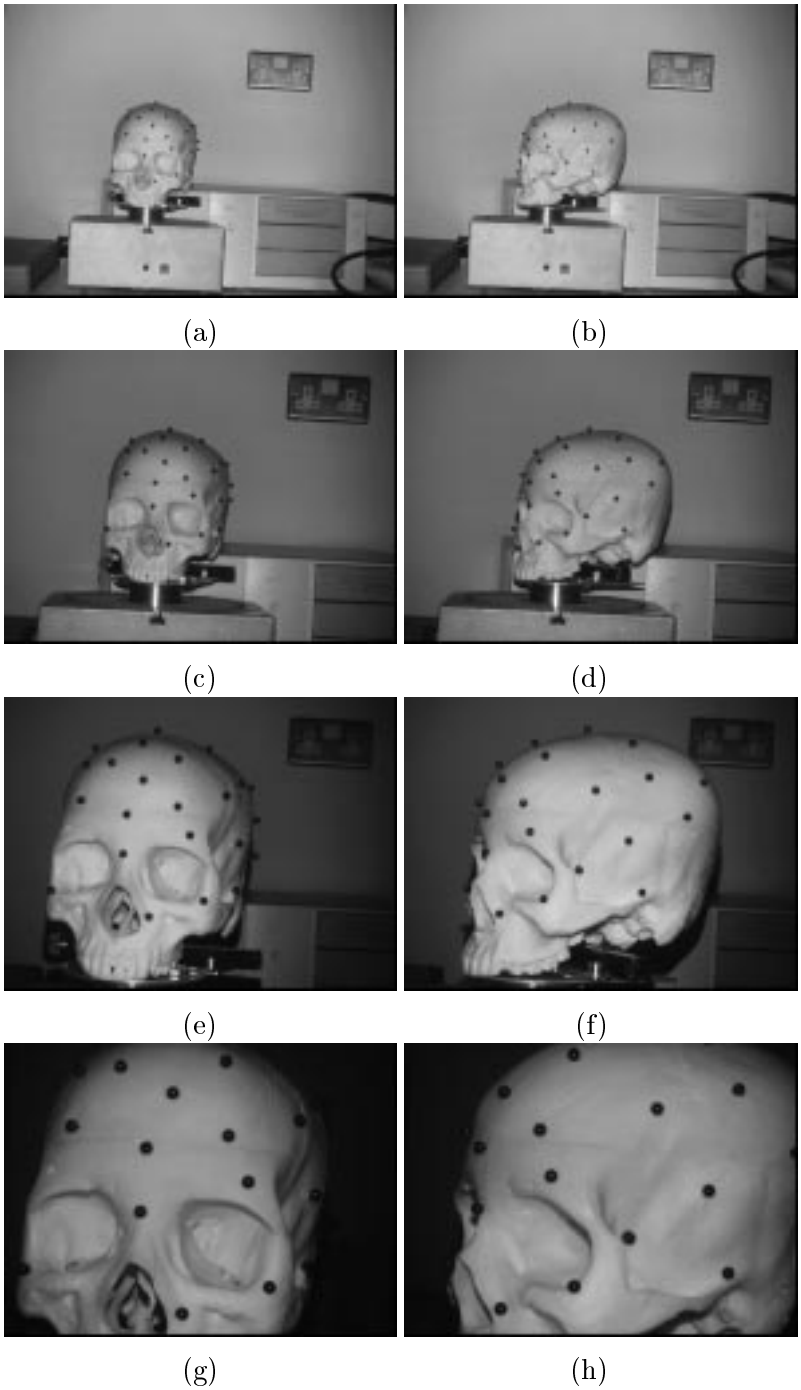


Figure 5.3: Four pairs of video images used for focal length experiments.

5.4.7 Registration For An Operating Microscope

5.4.7.1 Methods

The same surface model as that used for section 4.4.2 was again used for these experiments. Five video images were taken using an operating microscope (LEICA M695). The gold standard was calculated by localising the fiducials as before. However an SVD method [Gonzalez and Woods, 1992] was used to calculate the matrix representing the world to pixel transform rather than explicitly calculating the intrinsic and extrinsic parameters. Each video image only had 7-10 fiducials visible in the field of view which caused Tsai's algorithm to fail to calibrate. The SVD method performed better than Tsai's method with 7-10 point correspondences. The algorithm was tested for misregistration sizes of $\delta t = \pm 4$ mm and degrees. The value of $\delta t = \pm 4$ was chosen because the microscope images have a much smaller field of view, i.e. the object is magnified considerably. Thus if $\delta t = \pm 8$ mm and degrees was chosen the misregistration size relative to the field of view was too large for the algorithm to robustly register. It was assumed that in intra-operative use the algorithm could be initialised e.g. using skin features as fiducial markers, to be within ± 4 mm or degrees from the true registration. Each multiple view method, combining, adding and alternating were used to register all five video views simultaneously to the surface model. For successful registrations, the mean and standard deviation projection and 3D error calculated for each method. As discussed in section 4.3.7, the virtual light source used for the rendering was set to have the same position as the virtual camera. Furthermore, the rays of light emitted from the virtual light source were all parallel to each other and aligned with the optical axis of the virtual camera. This is depicted in figure 4.5(a).

5.4.7.2 Results

Table 5.9 shows a comparison of the three multiple view methods. There still seems to be a residual error of ≈ 3.5 mm, suggesting that there is some bias present. Using the alternating method the algorithm initially failed to converge. The algorithm would align the surface model with one view, and then move onto the next view. As it aligned itself to the next view it would misalign itself with the previous view. The meant that the algorithm did not converge, and a maximum number of iterations (2500 renderings) had to be chosen to stop the algorithm. The results shown are those where this stopping criteria was used.

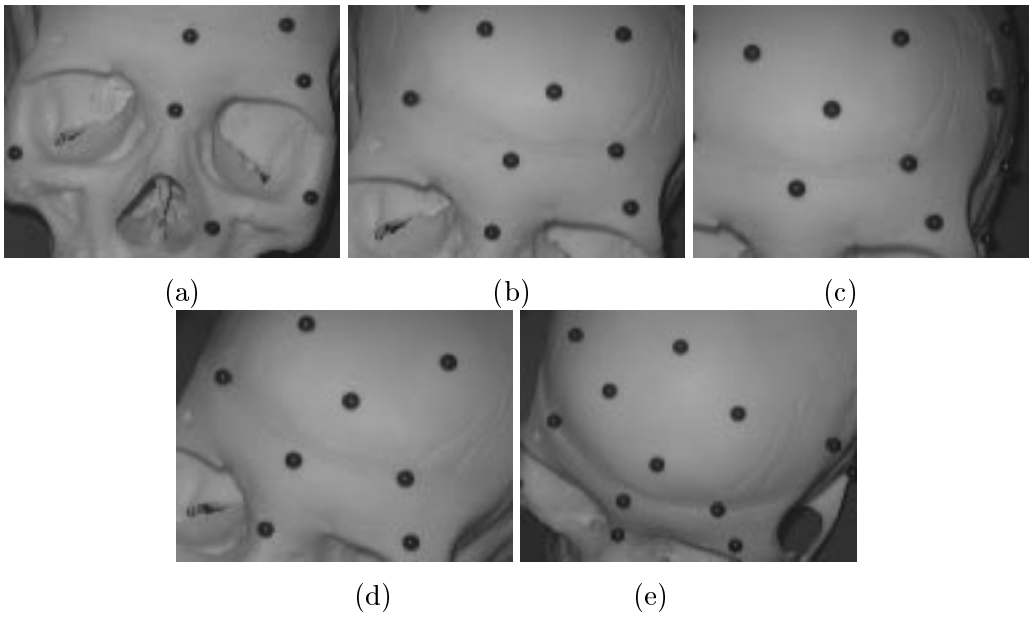


Figure 5.4: Five video images used for multiple view registrations (see text).

Method	Projection Error (mm) Mean (StdDev)	3D Error (mm) Mean (StdDev)	Percentage Success
pre-registration	7.74 (0.55)	9.62 (0.45)	
adding	3.32 (0.52)	3.68 (0.69)	100
combining	3.86 (0.39)	4.12 (0.48)	100
swapping	3.10 (0.41)	3.28 (0.49)	100

Table 5.9: Mean (standard deviation) projection and 3D errors for each multiple view method. The virtual light source was co-incident and co-axial with the virtual camera. $\delta t = \pm 4$ mm and degrees.

5.4.7.3 Conclusions

The experiments with the operating microscope again demonstrated that the adding and combining method performed similarly and significantly better than the alternating method (see table 5.9). The results in table 5.9 show that a systematic error is still present (these results were presented in [Clarkson *et al.*, 1999a]). This systematic error is addressed in the next section.

5.4.8 Calibrating The Light Source Position

5.4.8.1 Methods

It has so far been assumed that the real scene is illuminated by one light source, and that this light source is not only fixed relative to the camera but exactly aligned with it. The assumption of one light source is easily realised in practice. However, the light source in these experiments is not exactly aligned with the video camera, but is slightly offset. The following additional calibration procedure was performed to determine the optimum position of the virtual light source relative to the virtual camera.

The position of the virtual light source can be specified using three coordinates denoted by l_x, l_y and l_z. These represent the x, y, and z position of the virtual light source in the virtual camera coordinate system. The z axis of the camera coordinate system is the camera's optical axis, and thus changing the value of l_z by a small amount does not change the scene illumination significantly, and it is therefore neglected. This calibration stage would ideally be carried out with a dedicated calibration object. In this work, however, the calibration was retrospective, using the five video images in figure 5.4 and the gold standard transformation.

The mutual information of the rendered and video images was maximised by changing the light parameters l_x and l_y, whilst keeping the extrinsic camera parameters fixed at their gold standard calibrated positions. This produced a 'calibrated' light source position, where the position of the rendering light source relative to the rendering camera should more closely mimic the position of the real light source relative to the real camera. The above multiple view registration experiments were then repeated using this additional information. In other words a single two DOF search for l_x and l_y to find the best position of the virtual light source relative to the virtual camera was performed, and then the 64 registration experiments for each multiple view method were repeated.

Following this, the experiments were repeated for different combinations of images. The 'calibrated' light source position was used, and for $\delta t = \pm 4$ mm and degrees the registrations were repeated for every combination of 1,2,3 and 4 images from the 5 images in figure 5.4. After each registration, the registrations were classified as 'successful' or 'failed' as before, and the mean and standard deviation projection and 3D error for each number of images was calculated from the successful registrations.

Method	Projection Error (mm) Mean (StdDev)	3D Error (mm) Mean (StdDev)	Percentage Success
pre-registration	7.74 (0.55)	9.62 (0.45)	
adding	0.93 (0.30)	1.28 (0.35)	100
combining	0.68 (0.26)	1.05 (0.38)	100
swapping	3.10 (0.41)	3.28 (0.49)	100

Table 5.10: Mean (standard deviation) projection and 3D errors for each multiple view method. In this case, the virtual light source position relative to the camera was optimised before the registration took place. $\delta t = \pm 4$ mm and degrees.

Number Of Images	Projection Error (mm) Mean (StdDev)	3D Error (mm) Mean (StdDev)	Percentage Success
(mono) 1	7.80 (1.03)	3.75 (1.18)	75
2	1.49 (0.88)	2.05 (1.09)	96
3	1.06 (0.67)	1.49 (0.88)	99
4	1.55 (0.79)	2.16 (1.12)	100 (319/320)
5	0.68 (0.26)	1.05 (0.38)	100

Table 5.11: Mean (standard deviation) projection and 3D error for mono (1) through to 5 image registration. $\delta t = \pm 4$ mm and degrees.

5.4.8.2 Results

Table 5.10 shows the results when the position of the virtual light was optimised before the registration of the six rigid body parameters. The change detected was $l_x = -27, l_y = -9$mm which was a small change, and not validated. This shift in the light source position seemed reasonable, given the real camera/light setup. Both the projection error and 3D error have significantly decreased for the cases of adding and combining the information. For the alternating method, the oscillation problem still exists. For the combining method, and optimising the light source position before registration, the projection error has decreased to a mean (standard deviation) of 0.68 (0.26) mm and the 3D error to 1.05 (0.38) mm. Table 5.11 shows the mean (standard deviation) projection and 3D errors with the number of images. Three to five video image provides accurate, robust registration, even with images which have a small field of view. Figure 5.5 show the registration results for the combining method and using an optimised light source position. Images (c) - (f) are formed from a mixture of the video image and rendered image. Images (c) and (e) clearly show significant misregistration which has been corrected in images (d) and (f).

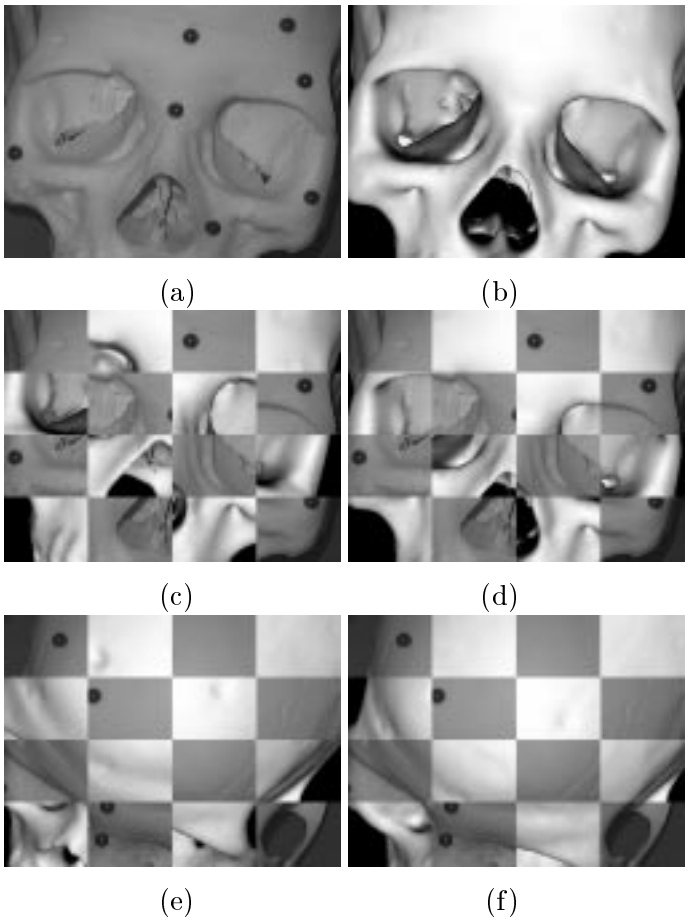


Figure 5.5: Results from multiple view registration. (a) Example video image. (b) Example rendered image. Note the absence of fiducials. (c) mis-registered overlay. (d) registered overlay. (e) mis-registered overlay. (f) registered overlay. The virtual light source position was optimised before registration, and the information was 'combined' into one histogram.

5.4.8.3 Conclusions

In addition to the results in [Clarkson *et al.*, 1999a], table 5.11 shows how the errors change with increasing number of images. Mono view performance is poor with these images, possibly due to the limited field of view. It can be seen that with 3 - 5 images, registration is robust and accurate. No validation of the accuracy of the recovered light source position was performed. The position of the light source should ideally be calibrated using a separate calibration object rather than the images used to test the registration.

5.4.9 Comparison Of Similarity Measures

5.4.9.1 Methods

In section 4.4.6 a comparison of similarity measures revealed that mutual information was the best similarity measure, of those tested, for registering a mono video image of a skull phantom to a surface model. This section introduces the first attempts to register video images of a volunteers face to a surface model of that volunteer. The video images and a rendering of the surface model is shown in figure 5.6. The surface model was acquired using a Tricorder^{TM} S4m system. This system projects a pseudo-random dot pattern onto a subject and captures four video images using four calibrated video cameras. The surface is reconstructed by matching corresponding points in the four views, and triangulating to reconstruct 3D positions. In addition, a further four video images are taken, illuminated with a single plain white light. The Tricorder^{TM} system uses these image to map texture onto the reconstructed surface. The output of the Tricorder^{TM} system is a surface and texture images. Each point on the surface has a 'texture coordinate' which maps the 3D surface position to a 2D texture image. These texture coordinates were used as input to Tsai's camera calibration method to recover the gold standard extrinsic and intrinsic camera parameters for each camera [Tsai, 1987]. The proposed algorithm registers the reconstructed surface to the video images produced with the plain white light illumination. The amount of radial distortion present in the video images was small and hence ignored. Thus for the four cameras the intrinsic and extrinsic parameters were known, and that by design the surface was registered to the video images. This provides an accurate gold standard.

In this case it was known that there was one light source, approximately centred between the four video cameras. The light source position was optimised as a pre-calibration step, similar to the previous section. For the misregistration size of $\delta t = \pm 8$ mm and degrees, and for each similarity measure mutual information (MI), normalised mutual information (NMI), normalised cross correlation (NCC) and gradient correlation (GC) the algorithm was used to register the surface model to the video images. From the successful registrations, the mean and standard deviation projection and 3D errors were calculated. To compare each of the four similarity measures fairly, the total similarity measure for a given pose was the sum of the similarity of each video and rendered image pair, i.e. 'adding' the information as described in section 5.3.1.2.

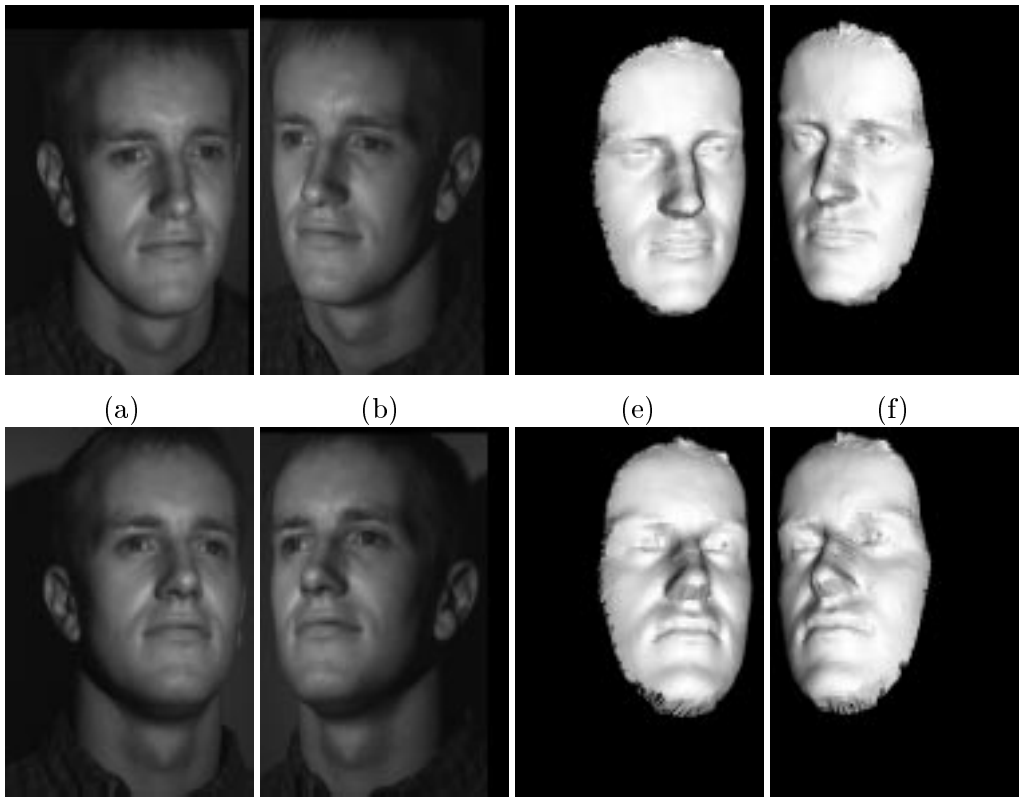


Figure 5.6: The four video images (a),(b),(c),(d) used in section 5.4.9 and a corresponding surface rendering (e),(f),(g),(h) respectively) at the gold standard position.

5.4.9.2 Results

The results are shown in table 5.12. It can be seen that in general the robustness is good, but the accuracy is poor. Of the similarity measures tested, NCC performs the worst in terms of mean projection and 3D errors. Looking at the results for each registration (not shown) reveals that in general the algorithm is still reliably finding a solution that is offset from the aligned position. From these results, the best similarity measure was normalised mutual information, with mean (standard deviation) projection and 3D errors of 1.29 (0.55) and 2.53 (0.27). However, this is not a sufficiently accurate registration. In addition, if the rendering light source is aligned with each rendering camera, as opposed to being fixed relative to each rendering camera then all the registrations fail completely. This is unsurprising as the real camera and light setup is such that the camera is roughly pointing centrally at the face, whilst the video images are top left, top right, bottom left and bottom right views. If the rendering light source is aligned with the rendering camera, then the shading pattern is very different from the video images, and none of the four similarity measures work.

Similarity Measure	Projection Error (mm) Mean (StdDev)	3D Error (mm) Mean (StdDev)	Percentage Success
(Pre-registration)	13.30 (1.96)	17.49 (0.72)	
MI	1.39 (0.60)	2.77 (0.35)	100
NMI	1.29 (0.55)	2.53 (0.27)	100
NCC	1.62 (0.54)	3.13 (0.19)	100
GC	1.46 (0.35)	2.64 (0.45)	100

Table 5.12: Mean (standard deviation) projection and 3D errors for each similarity measure tested.

5.4.9.3 Conclusions

With misregistration sizes of $\delta t = \pm 8$ mm and degrees, the algorithm performed robustly, but not accurately. With the approximate light source position, the algorithm precisely registered to a position that was offset from the gold standard position, i.e. inaccurate. It would be better the develop a method that was not dependent on having a calibrated light source position.

5.5 Summary

For these experiments, the adding and combining methods performed similarly and both were superior to the alternating method. The alternating method did not converge well, when near the solution. From these experiments, the best angular disparity using two views was 30 degrees. Testing the registration performance against the number of images, the best results were achieved with three images separated by a total of 30 degrees. In general the multiple view experiments performed more robustly and precisely than the mono view experiments. The multiple view algorithm usually produced lower 3D errors than the mono view algorithm, but not necessarily lower projection errors. The experiments testing the performance of the multiple view algorithm with changing focal length and field of view showed the same trends as the mono view algorithm and it was concluded that the multiple view algorithm was still failing to work well in these cases. The method shown here of finding the light source position through a separate optimisation procedure serves to demonstrate that even a small shift in the position of the assumed rendering light source produces a significant effect on the registration accuracy. With this method the projection error was reduced to 0.68 (0.26) mm and the

3D error was reduced to 1.05 (0.38) mm. These registration errors have the same order of magnitude as the expected error of the calibration process and the extraction of the surface model from the 3D image.

The registrations in this chapter took on average 10-45 minutes. Clearly, the more views used, the longer the registration takes. It would seem that adding the extra views has significantly improved the accuracy and robustness of the algorithm, but it means that the algorithm is falling further short of the target speed of 3-5 minutes as specified in section 3.7.

Finally, to conclude, these experiments have shown that the method shown here can be sufficient for registering multiple video views to a 3D model with a projection and 3D error of about 1 mm, and with a high level of precision. However this is not always the case. The registration performance was unsatisfactory for images of a volunteers face, and was also dependent on focal length and field of view.

Chapter 6

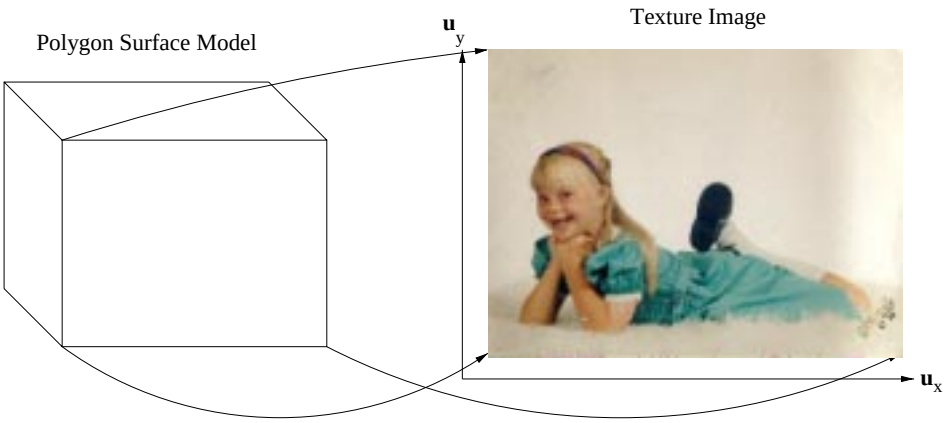
Using Texture Mapping For Tracking

6.1 Introduction

In this chapter, a simple but novel method for tracking an object using texture mapping is proposed. Previous chapters have described an algorithm to register one or more video images to a surface model derived from a 3D volume. However, the experiments have shown in chapter 5 that the algorithm can register well for untextured surfaces such as a skull phantom, but does not register well for more textured surfaces like a human face. Consider a sequence of video images, where an accurate registration between a 3D model, and the first video image has been performed. In this case, pixel grey values in the first video image can be directly associated with points in the 3D model. The registration provides information describing what a 3D point in the surface model should look like in 'real life', and this information is not present in the original 3D tomographic image or surface model. Information from an initial registered video view can be texture mapped onto the model and used to assist registration to subsequent video frames. Tracking in this context is simply registering a 3D image to a sequence of video images. This chapter describes in detail a new tracking algorithm. The tracking algorithm is tested using a mono and multiple view simulation and then a mono and multiple view tracking experiment, tracking a volunteer's face. Finally the tracking performance is compared with a surface based registration algorithm [Maurer Jr. *et al.*, 1996].

6.2 Aim

The aim of this chapter is to investigate whether texture mapping can be used to assist a tracking algorithm and whether it significantly improves the accuracy and robustness of mutual information based tracking when compared with the non-texture mapping algorithm of the previous two chapters.



(a)

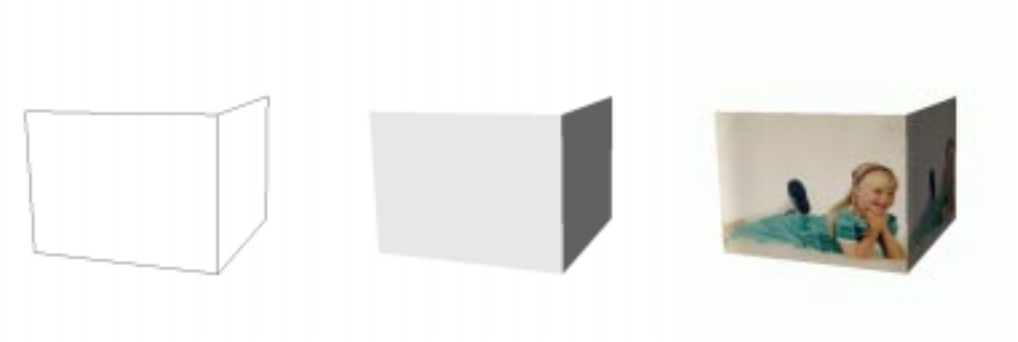
Figure 6.1: Texture coordinates map vertices to texels.

6.3 Methods

6.3.1 Texture Mapping

As the required detail within a rendered image increases, explicitly modelling object surfaces using graphics primitives becomes increasingly less practical. For instance to create a polygon model to represent a wooden floor may require an individually coloured polygon for each grain in the wood. This would be difficult to define, and computationally expensive to render. Texture mapping (or pattern mapping), pioneered by [Catmull, 1975], is a simple approach to map an image onto a surface to provide additional realism.

The technique of colour mapping [Catmull, 1975] is used as an example of one type of texture mapping method, and the method used in this chapter. For other types of texture mapping see [Foley *et al.*, 1990]. To perform colour mapping, a surface is defined as a set of polygons, that is to say, a set of points, with known connectivity. A texture coordinate $\mathbf{u} = (u_x, u_y)^T$ is assigned to each vertex in the surface, see figure 6.1. The figure shows an image of 'Anna' which is used as the texture image. When a polygon is rendered, the colour at each point on the polygon is determined by interpolating between pixels in the texture image (also called texels).



(a) (b) (c)

Figure 6.2: Texture Mapping Example: (a) Geometry is defined as polygons. (b) Geometry is surface rendered. (c) Texture mapping adds detail, with a small overhead.

An example can be found in figure 6.2. Two planes have been defined. These are shown, rendered as a wireframe in figure 6.2(a) and as a solid surface in 6.2(b). The corners of each plane have texture coordinates corresponding to the corners of the image Anna. The texture image is then mapped onto the planes during the rendering. This results in a highly detailed rendering with interactive frame rates on a typical graphics workstation.

6.3.2 Tracking

This chapter describes a tracking algorithm that uses texture mapping and mutual information. However, the intention is to describe the tracking algorithm as an extension to the work of the previous chapters, and to demonstrate that texture information does enable more robust matching, and to measure the performance of the proposed system. The intention is not to develop a video frame rate tracking system.

Three experiments are described in this chapter. A simulation and two tracking experiments with real data. Tracking with single and multiple views is tested. As before, it is assumed that the intrinsic camera parameters of each camera are known. For the multiple view experiments it is also assumed that the rigid body transformation relating each camera's coordinate system is known. Tracking is simply registering a 3D image or surface model to a sequence of video images i.e. repeatedly registering.

6.3.2.1 Notation

Recall from section 2.1, equation (2.1) that the transformation from 3D model coordinates $\mathbf{m} = (m_x, m_y, m_z, 1)^T$ to 2D pixel coordinates $\mathbf{p} = (p_x, p_y, 1)^T$ was accomplished using the equation

$$k\,\mathbf{p} = \mathbf{M}\,\mathbf{m} \tag{6.1}$$

where $\mathbf{M}$ is a 3×4 perspective projection matrix and k is a homogeneous scale factor. In this chapter stereo pairs of cameras are used. Let $c = 1, 2$ denote the camera number. Using these two cameras we acquire or simulate a sequence of video images. Let $v = 1 \ldots N$ denote the video image number. The matrix $\mathbf{M}$ will be different for each camera and for each video image. Therefore let $\mathbf{M}_{c,v}$ be the transformation from 3D scene coordinates to 2D video image pixels for camera c and for video image v. Let $\mathbf{Q}_{c,v}$ be a rigid body transformation from model coordinates to camera coordinates, and $\mathbf{P}_c$ be a projection matrix formed by the intrinsic camera parameters for camera c. The matrix $\mathbf{M}_{c,v}$ can be represented as

$$\mathbf{M}_{c,v} = \mathbf{P}_c\,\mathbf{Q}_{c,v} \tag{6.2}$$

such that

$$k\,\mathbf{p}_{c,v} = \mathbf{M}_{c,v}\,\mathbf{m} \tag{6.3}$$

First tracking experiments using a plastic skull phantom are performed. In this case the 3D coordinate system is defined by the model (CT) coordinate system. $\mathbf{Q}_{c,v}$ is a transformation from 3D model coordinates to 3D camera coordinates, and $\mathbf{P}_c$ is a transformation from 3D camera coordinates into 2D image pixels. The matrix $\mathbf{P}_c$ is calculated using a calibration process and is fixed throughout the tracking process. Consider a sequence of N images denoted by V_v where $v = 1 \ldots N$ taken from camera c. Assume that for both cameras in the system $c = 1, 2$, the initial registration of 3D image coordinates to the video image pixels is known. This means that for V_1, $\mathbf{Q}_{c,1}$ is known. The goal of the tracking is to find the rigid body transformation which, when combined with the initial known registration matrix $\mathbf{Q}_{c,1}$ and camera calibration matrix $\mathbf{P}_c$, transforms 3D image points onto the corresponding 2D video image pixels throughout a sequence of video images. The desired rigid body transformation is represented by $\mathbf{Q}_v$ where

$$\hat{\mathbf{Q}}_{c,v} = \mathbf{Q}_{c,1}\,\mathbf{Q}_v\,. \tag{6.4}$$

$\hat{\mathbf{Q}}_{c,v}$ is the updated rigid body transform produced by our algorithm. The matrix $\mathbf{Q}_v$ is determined by the six extrinsic parameters t_x, t_y, t_z, r_x, r_y and r_z as described in sec-

tion 2.2.8. The matrix $\mathbf{Q}_v$ is the output of the algorithm after each video frame, v, in the sequence. If the gold standard transformation $\mathbf{Q}_{c,v}$ is known then $\hat{\mathbf{Q}}_{c,v}$ should be approximately equal to $\mathbf{Q}_{c,v}$. Thus the tracking problem is to determine the six degrees of freedom t_x, t_y, t_z, r_x, r_y and r_z which updates the transformation from 3D model coordinates to 2D pixel coordinates for each video frame in a sequence.

6.3.3 Why Use Texture Mapping For Tracking?

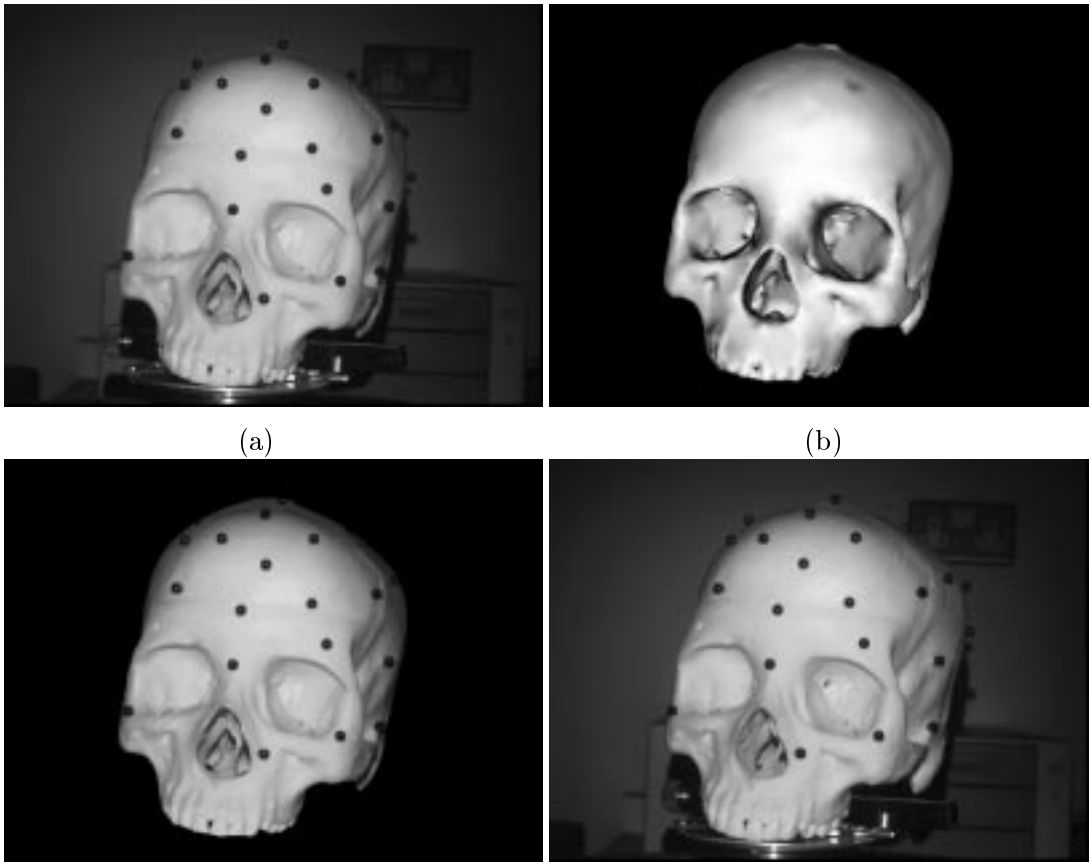
The reason for using texture mapping is best shown by example. In figure 6.3, image (a) shows an example video image of a skull phantom, similar to those used in previous chapters. In addition, image (b) shows a surface model of the skull phantom, registered with the video image. Once registered, the video pixel information can be mapped back onto the surface model, as shown in image (c). Image (d) shows another example video image, where the skull has been rotated by 6 degrees. The mutual information of the plain rendered surface model (b) and the video image (d) is 0.67. The mutual information of the textured rendered surface model (c) and the video image (d) is 0.81. Intuitively, assuming that the texture is mapped onto the correct location on the surface model, and of course that the surface model is an accurate representation of the real object, then the texture mapped rendering should be more similar to subsequent video images of the same object than the plain rendered model. In addition, this makes no assumption about what type of object you are tracking. The surface model can be any shape, and any set of image intensities can be mapped onto the surface. It would be expected that the more features in the video image, then the better tracking performance could be achieved.

6.3.4 Calculating Texture Coordinates

Once a video image is registered, then from section 6.3.2.1, for image number $v = 1$ and camera number $c = 1, 2$ and equation (6.3) then

$$k\,\mathbf{p}_{c,1} = \mathbf{M}_{c,1}\,\mathbf{m} \tag{6.5}$$

where $\mathbf{m} = (m_x, m_y, m_z, 1)^T$ is a 3D surface model point, and $\mathbf{p}_{c,1} = (p_x, p_y, 1)^T$ is a 2D video image point in pixels. Given a surface model, an initial registration matrix $\mathbf{M}_{c,1}$ and a video image, there are several different ways of selecting which 3D points in the surface model are useful for the tracking and hence which polygons need texture mapping onto them. Two methods are described below.



(a) (b)

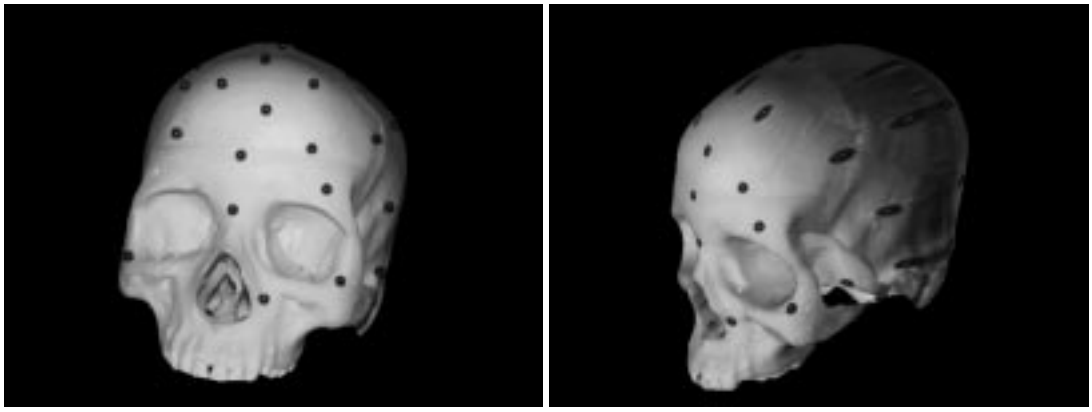
(c) (d)

Figure 6.3: Texture mapping example: (a) The first image in a video sequence. (b) The model is registered to the video image. (c) The video texture is pasted onto the model. (d) The texture mapping makes the model more similar to subsequent video images.

6.3.4.1 Projection Onto The Image Plane

From equation (6.5), each 3D point $\mathbf{m}$ can be projected to its corresponding 2D point $\mathbf{p}$, and this 2D point is the necessary texture coordinate. The first problem with this method is what to do with polygons that do not correspond to a surface nearest to the camera. For instance, the surface model rendered in figure 6.3(b) is a model of the whole skull. The video image in figure 6.3(a) is of the front of the skull. If every 3D point in the surface model is assigned a texture coordinate by multiplying by $\mathbf{M}_{c,1}$ then the video texture corresponding to the front of the face will also be mapped on the back and sides of the skull. This can be seen in figure 6.4.

A further problem exists due to perspective foreshortening and is illustrated in figure 6.5. If a polygon is near parallel to the image plane then texture on the video image from region C will be mapped onto a small area at region D. If however the polygon is



(a) (b)

Figure 6.4: If each 3D point in the surface model is assigned a texture coordinate then (a) texture is mapped onto the model correctly at the front of the skull phantom but (b) if the model is rotated, the texture is observed, 'smeared' across the surface, and pasted incorrectly at the sides and back of the skull phantom.

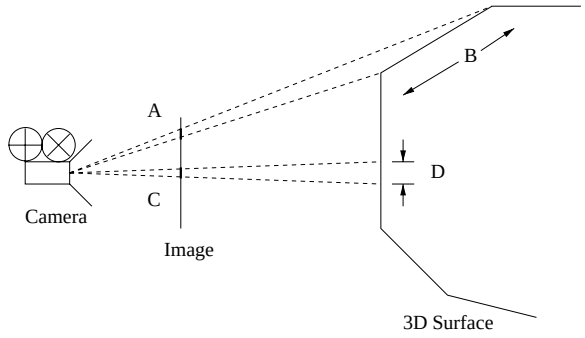


Figure 6.5: The texture map is distorted as it is mapped onto a polygon

at an oblique angle to the image plane, a similar amount of video texture at region A will be mapped onto region B. Thus the texture map is distorted when reprojected onto a 3D object.

This simple projection method can still be used if the surface model only spans roughly the same area as that visible in the video image, or if during tracking, the object is known to move only a small distance over time.

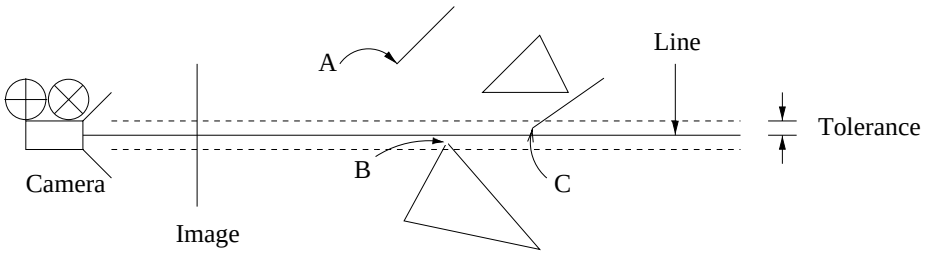


Figure 6.6: The closest surface is extracted by projecting (casting) lines into the 3D world and finding the closest points in the surface model within a tolerance. Here, point A is closest to the camera, but point B is the closest point within tolerance.

6.3.4.2 Back Projection

An alternative method is to select only those polygons which correspond to the front surface with respect to the video camera, i.e. those polygons which actually represent the same surface as viewed in the video image.

Using $\mathbf{M}_{c,1}$, then for a given 2D point, a line can be defined from the camera's optical centre, projecting through the 2D point on the image plane, and continuing into 3D space. The closest 3D model point to this line is found. See figure 6.6. Once a 3D point has been found that is a point in the front most surface, then the texture coordinate can be calculated as above in section 6.3.4.1.

In addition this method can be used to filter out polygons that are at too oblique an angle to the image plane. If each 3D point is stored with a surface normal, then given a vector describing the direction of the camera, each point that is deemed close enough to the camera to be useful for texture mapping, can be discarded if the dot product of the surface normal and the camera direction is below a threshold.

Figure 6.7 demonstrates this method. Image (a) is the texture mapped onto the model, where only the front most polygons are used, and the remaining are discarded. Image (b) shows that if the texture mapped surface model is rotated, then rear most polygons may become visible through gaps in the model. In this case, in image (b), polygons corresponding to the skull's left side can be seen through the left orbit. Image (c) shows that if no thresholding of oblique polygons is performed whilst applying the texture

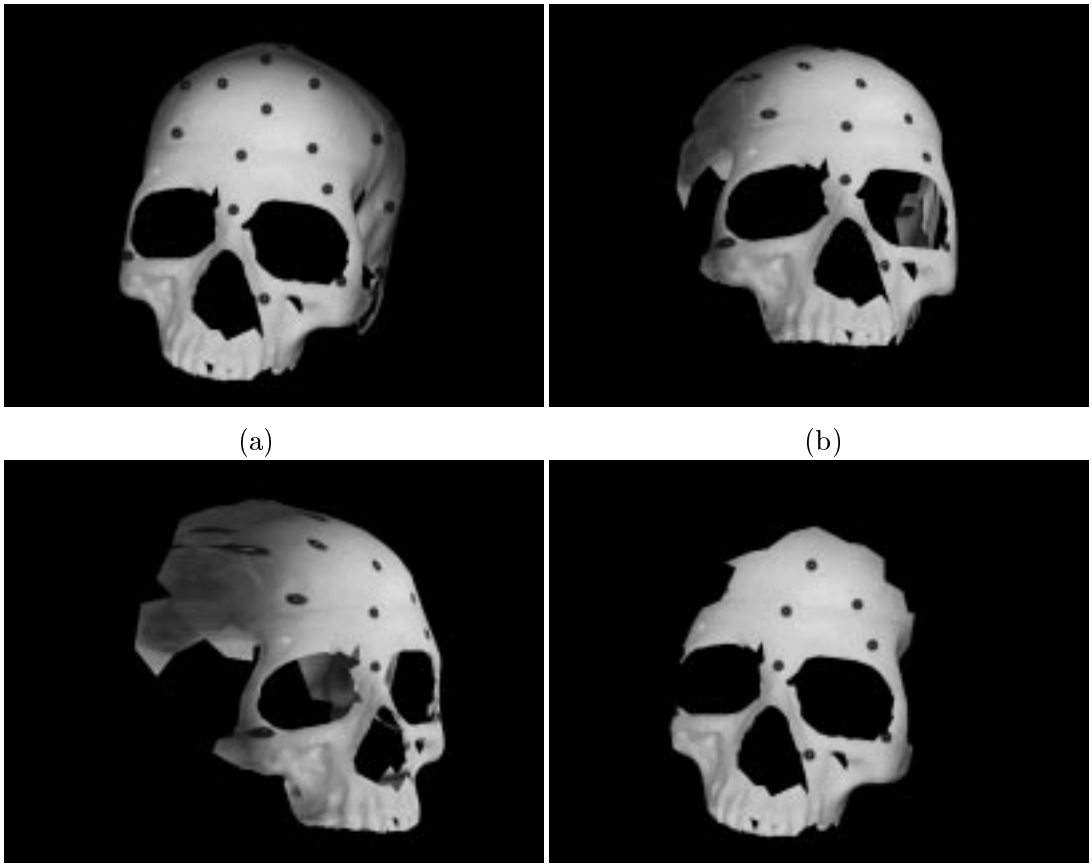


Figure 6.7: Problems with selecting polygons for texture mapping. (a) Texture is mapped onto polygons that are near to the camera. (b) Rear most polygons are not occluded correctly. (c) If the skull is rotated further, rear most polygons are not occluded and texture map is distorted. (d) If oblique polygons are clipped, but model has decreasing number of polygons, and occlusion issues may still exist. See text section 6.3.4.2.

map, then, when the skull model is rotated, polygons with a distorted texture map will be visible. If a threshold is applied to clip polygons that are oblique to the image plane when applying the texture map, then it becomes difficult to select a threshold that removes a good number of polygons. Image (d) shows a texture mapped model, that does not have polygons that are oblique to the image plane, but so many polygons have been removed that important features such as edges have disappeared. It is difficult to find a good tolerance for the ray casting process (figure 6.6) and a threshold for discarding oblique polygons that doesn't also discard too many polygons.

6.3.4.3 Choice Of Method

To generate the texture coordinates for the experiments that follow, the 3D surface model points were projected using equation (6.5) to find the corresponding 2D pixel location. i.e. the method of section 6.3.4.1. This method was chosen because it was simple and quick to calculate, and it was found to be sufficient in the following experiments. The experiments in section 6.4.1 only used small angles of rotations between each image in the tracking sequence and thus the effects described in 6.3.4.1 will not be prominent in the rendered image. Mutual information is known to be robust to occlusion [Viola, 1995] or spurious information which will not help the match. Thus if rotation angles are small, then the texture which has been mapped to the model and which has also been distorted due to perspective foreshortening, will not have a large impact. In section 6.4.2, the surface model only comprises of a section of the front part of the face. This can be seen in figure 6.13(b). Thus in this case, the artifacts described in section 6.3.4.1 will not be produced.

6.4 Experiments

Three experiments were performed to demonstrate the potential of this concept. These experiments are summarised below and then explained in further detail.

- **Tracking Simulation** The surface model of the plastic skull phantom (see section 4.4.2) and two video images were taken. The video image texture was mapped onto the model and a series of 100 pairs of video frames were *simulated* by repeatedly rendering the surface model in a sequence of known poses. The texture mapping and non-texture mapping algorithm were used to track the motion. See section 6.4.1.

- **Tracking A Volunteer** An MR scan and a series of 25 pairs of images taken of a volunteer were taken. The gold standard was provided using a Lockable Acrylic Dental Stent (LADS) which enabled the volunteer and camera movement to be independently tracked using an optical tracking device (Optotrak, Northern Digital). The texture mapping and non-texture mapping algorithms were used to track the motion of the volunteer relative to the camera and compared to the Optotrak gold standard. See section 6.4.2.

(a) (b)

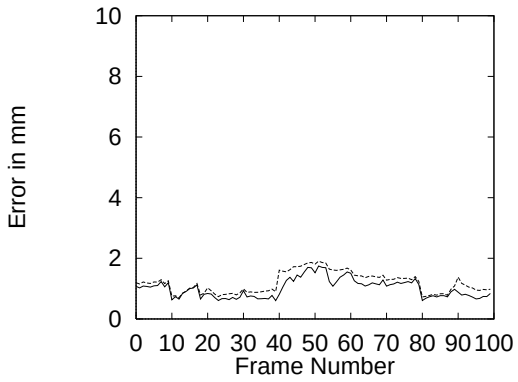
Figure 6.8: Example images: (a) and (b) are the stereo pair used for the skull phantom experiments as described in section 6.4.1.

- **A Comparison With A Surface Based Registration Algorithm** A series of video images was taken using a Tricorder^{TM} S4m system. This system performs a surface based reconstruction from video image information. Thus for a series of images, the first surface can be registered to subsequent reconstructed surfaces using a surface based registration algorithm [Maurer Jr. *et al.*, 1996]. In addition, the first surface can also be registered to the subsequent video images using the proposed texture mapping algorithm and the non texture mapping algorithm from the previous chapter. Both algorithms can be compared with the surface based algorithm. See section 6.4.3.

6.4.1 Tracking Simulation

6.4.1.1 Methods

The same skull phantom as that used in section 4.4.2 was used to perform a tracking simulation, and the two video images shown in figure 6.8. The images differ by a rotation of the skull of 45 degrees. The surface model was registered to the video images by localising the fiducials and using Tsai's algorithm [Tsai, 1987] as described in section 4.3.10. This meant that the intrinsic and extrinsic camera parameters for each view were known. The texture from the video images was mapped onto the surface model. Two virtual rendering cameras were created and 100 images for each camera were then *generated synthetically* by changing the extrinsic parameters, calculating the pose of the 3D model with respect to each camera and producing a texture mapped rendering for each camera. Zero mean, Gaussian noise ($\sigma = 7$) was added to these simulated images. The value of σ was chosen to simulate video image noise. The set of 100 images per camera were a sequence of left/right and up/down rotations, where the change in pose of the 3D



(a)

Figure 6.9: 3D (dotted line) and projection (solid line) Errors for the mono view simulation, with texture mapping, as described in section 6.4.1.

Case	Projection Error (mm)	3D Error (mm)
Mono	4.22	6.20
Stereo	3.04	3.79

(a)

Case	Projection Error (mm)	3D Error (mm)
Mono	1.01	1.19
Stereo	0.83	1.05

(b)

Table 6.1: A comparison of mono view and stereo view performance for the simulation. (a) without texture mapping, (b) with texture mapping.

model with respect to the camera between each frame was one degree. A 'mono view' tracking experiment was then performed by taking the sequence of simulated images for a single camera and using the known initial registration $\mathbf{M}_{1,1}$ to initialise the tracking algorithm. The algorithm was used to recover the transformations $\mathbf{M}_{1,v}$ for $v = 2 \ldots 100$. This experiment was repeated, performing a 'stereo view' tracking experiment by taking the sequence of images for both cameras $c = 1, 2$, and using our algorithm to recover the transformations $\mathbf{M}_{c,v}$ for $v = 2 \ldots 100$. Both mono and stereo experiments were repeated using the non-texture mapping algorithm of chapters 4 and 5.

6.4.1.2 Results

Figure 6.9 (a) shows a graph of the projection and 3D errors in mm for the mono view simulation with texture mapping. Table 6.1(a) and (b) shows a comparison of the mono and stereo view tracking performance with and without texture mapping. For the non texture mapped mono case, the algorithm does not track well. The true motion is a rotation, but the algorithm seems to try and compensate for a rotation with translations. For the non texture mapped stereo case, the algorithm tends to 'lag

behind' when tracking the motion, but performs significantly better than the mono view algorithm. For the texture mapping case, both the mono and stereo experiments worked well. The 3D error for mono view texture mapped case is 1.19 mm and this improves to 1.05 mm for the stereo view texture mapped tracking.

6.4.1.3 Conclusions

The simulation did not include translation along the camera's optical axis, so it was expected that the texture mapped tracking algorithm would work well for both mono and stereo experiments. This was in fact the case. The mean 3D error is dependent on the final step size of the gradient search strategy (see section 4.3.4). This could be improved, but overall with this experiment, the algorithm has achieved accurate, reliable tracking, and the texture mapped tracking performance is clearly better than the non texture mapping experiment.

6.4.2 Tracking A Volunteer

6.4.2.1 Methods

An MRI scan ($1.016 \times 1.016 \times 1.250$ mm, $256 \times 256 \times 150$ voxels) was taken of a volunteer. This was corrected for scaling errors [Hill *et al.*, 1998], and a skin surface extracted using VTK [Schroeder *et al.*, 1997]. A pair of video cameras was fixed with respect to each other and calibrated using SVD [Gonzalez and Woods, 1992], which produces the matrix $\mathbf{P}_c$ for each camera as mentioned in section 2.2.10. A bivariate polynomial deformation field for each camera was calculated to correct for distortion effects. The translational separation of the two cameras was approximately 30 centimetres and the disparity between their optical axes was approximately 45 degrees.

The volunteer was scanned whilst wearing a Lockable Acrylic Dental Stent (LADS) [Edwards *et al.*, 1999c; Edwards *et al.*, 1999b]. This is a device which rigidly attaches to a volunteer or patients upper set of teeth. The LADS has imaging markers, which can be swapped for localiser caps. This enables the precise position of the markers to be measured in an MR or CT image and also in physical space using an optical tracking device (Optotrak, Northern Digital). Thus the LADS is used to register the volunteer's MR scan to physical space. Furthermore the LADS and the video cameras have infra-red LED's (IRED's) rigidly fixed to them. This enables the volunteer and camera's position to be tracked relative to each other, providing an independent gold

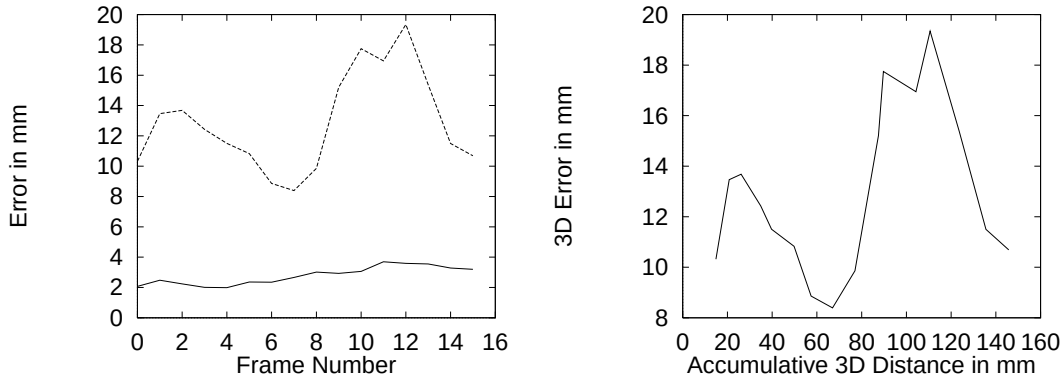
(a) (b)

Figure 6.10: Example images: (a) and (b) are the stereo pair used for volunteer experiments, as described in section 6.4.2.

standard for this experiment. In this volunteer based experiment the matrix $\mathbf{Q}_{c,v}$ shown in equation (6.2), is a transformation from MR coordinates to the camera coordinate system. Using the tracking information produced by the LADS [Edwards *et al.*, 1999a] and the Optotrak, the gold standard transformation $\mathbf{M}_{c,v}$ for each image $v = 1 \ldots 25$ can be calculated. A 'mono view' tracking experiment was performed, by taking the sequence of simulated images for camera $c = 2$ and using the known initial registration $\mathbf{M}_{2,1}$ to initialise the tracking algorithm. The texture tracking algorithm was used to recover the transformations $\mathbf{M}_{2,v}$ for $v = 2 \ldots 25$. Subsequently a 'stereo view' tracking experiment was performed by taking the sequence of images for both cameras $c = 1, 2$, and using the texture tracking algorithm to recover the transformations $\mathbf{M}_{c,v}$ for $v = 2 \ldots 25$. The mono and stereo experiments were also repeated using the non texture mapping algorithm of chapters 4 and 5.

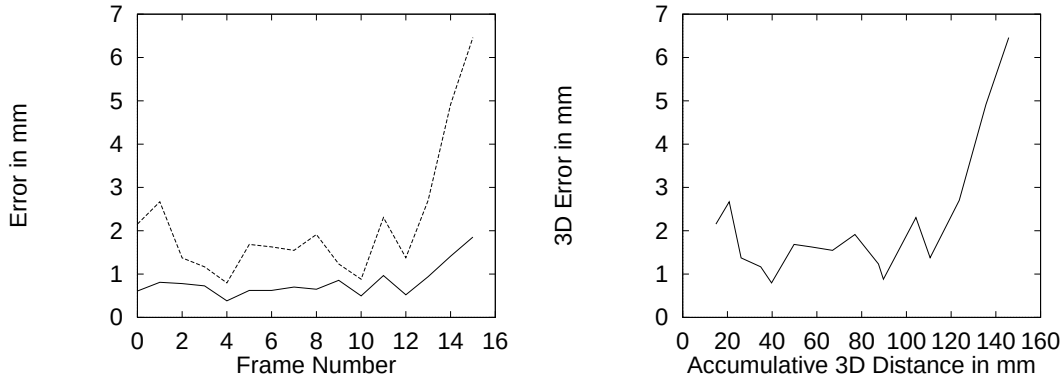
6.4.2.2 Results

Figure 6.10 shows two example video images. The images are a pair taken from the (a) left and (b) right camera. It can be seen that of the two images, one is significantly lower in contrast than the other. Figure 6.11 (a) shows the results for the mono view experiment on the volunteer. This graph shows that projection error and 3D error can be significantly different. Specifically the projection error can be reasonably low while the 3D error is high. A mono view experiment can fail to recover translations along the optical axis of the camera. Figure 6.11 (b) shows the 3D error plotted against the accumulated 3D distance which shows that the camera has moved over 140mm in total. Figure 6.12(a) shows that with stereo views, the tracking performance is much better. Figure 6.12(b) shows the 3D error as a function of accumulative 3D distance moved. Table 6.2 summarises the performance of the mono and stereo view algorithms, both with and without texture mapping.



(a) (b)

Figure 6.11: (a) 3D Error (dotted line) and Projection Error (solid line) for mono view, volunteer, texture tracking experiment. (b) 3D Error plotted against the Accumulated 3D Distance for the mono view volunteer, texture tracking experiment. See section 6.4.2.



(a) (b)

Figure 6.12: (a) 3D Error (dotted line) and Projection Error (solid line) for stereo view, volunteer, texture tracking experiment. (b) 3D Error plotted against the Accumulated 3D Distance for the stereo view, volunteer, texture tracking experiment. See section 6.4.2.

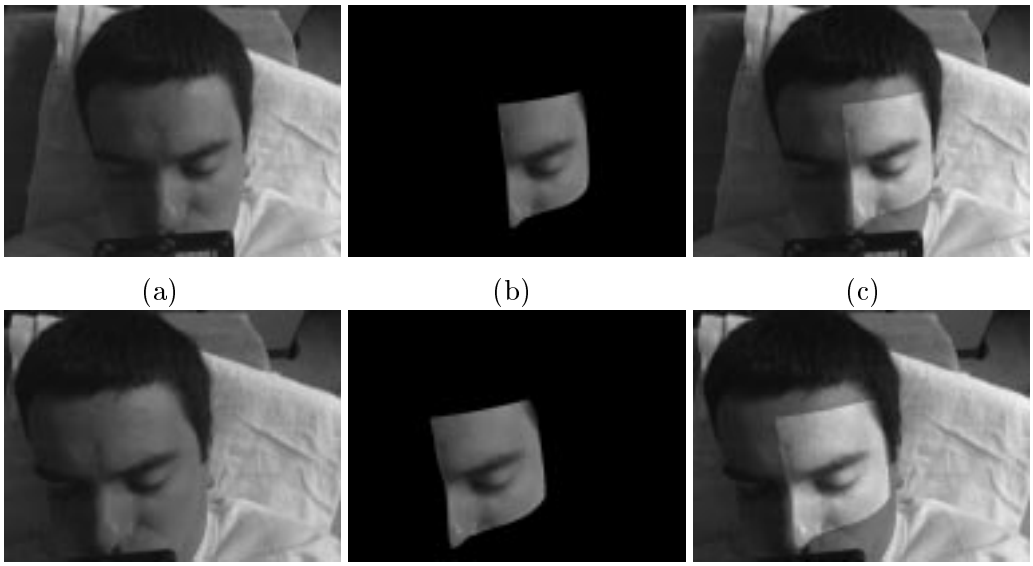
Case	Projection Error (mm)	3D Error (mm)
Mono	91.93	123.59
Stereo	134.93	147.01

(a)

Case	Projection Error (mm)	3D Error (mm)
Mono	2.75	13.03
Stereo	0.74	1.89

(b)

Table 6.2: A comparison of mono view and stereo view performance when tracking a volunteer (a) without texture mapping, (b) with texture mapping.



(a) (b) (c)

(d) (e) (f)

Figure 6.13: Results of volunteer tracking experiment: (a) Video image 1. (b) Texture mapped model. (c) Model registered and overlaid on video image, at the initial pose, before tracking. (d) Video image 12. (e) Texture mapped model at the tracked pose. (f) Model registered and overlaid on video image at the tracked pose.

To summarise, the mono and stereo view, non texture mapping experiments failed completely. For the volunteer tracking experiment with texture mapping, it can be seen that the stereo algorithm performs significantly better than the mono view algorithm. However after 14 frames, corresponding to 140 mm of accumulative 3D movement, the stereo algorithm fails to track. This was attributed to the fact that the difference in the relative position of the camera and volunteers between consecutive frames was too large.

Figure 6.13(a)-(f) shows some example images. Image (a) is the initial camera image. (b) is the surface model, upon which the texture from image (a) is mapped. Image (c) shows the texture mapped surface model overlayed onto the video image (a) illustrating how the texture mapped surface model matches image (a). Image (d) shows a subsequent video frame in the sequence, (e) shows the updated registration position of the surface model and (f) shows how this updated position does indeed match image (d).

6.4.2.3 Conclusions

From the volunteer tracking experiments the mean 3D error was 1.89mm. This must be compared with the accuracy of the gold standard. The Optotrak can track IRED's (infra-red light emitting diodes) attached to the LADS, accurately to within 0.1 - 0.2mm. However the LADS is used as part of an image guided surgery system MAGI (Microscope Assisted Guided Interventions [Edwards *et al.*, 1999c]). MAGI registers MR space to physical space, and independently tracks the volunteer and the video cameras. The video cameras are calibrated to physical space. Thus the overall system accuracy of MAGI is dependent on many factors and was assessed to be 1.6mm [Edwards *et al.*, 1999c]. Thus the accuracy of the texture mapped tracking of 1.89mm is comparable. Table 6.2 shows that the non texture mapping algorithm fails completely, whereas the texture mapping algorithm tracks well up until frame 14. The images were grabbed using a frame grabber that grabbed a single image at each button click. The camera was moved manually relative to the volunteer and after each movement an image was grabbed. The algorithm failed to track at frame 14 as there was too large a movement between frames. A better system could be implemented that continuously grabs video frames, which would result in much smaller relative movement between video frames.

6.4.3 A Comparison With A Surface Based Registration Technique

6.4.3.1 Methods

In this section, an experiment is described in which the Tricorder^{TM} S4m system is used to create input data for the algorithm. As mentioned in the previous chapter, the Tricorder^{TM} S4m system takes sets of video images, and reconstructs a texture mapped surface. A single 'grab' for the S4m captures four video images, from four cameras, with the scene illuminated with a pseudo-random speckle pattern and four video images illuminated with plain white light. As the four video cameras are accurately calibrated, a surface can be reconstructed from the patterned light images, and texture mapped with information from the four plainly lit video images.

The following experiment was devised. A series of 56 sets of images was captured by the S4m system whilst the volunteer moved slowly within the field of view. For each of the 56 sets, the corresponding surface was reconstructed. The first surface was then taken, clipped to remove spurious surface data, and registered to the remaining 55 in

the order they were taken. The algorithm used was an independent implementation [Maurer Jr. *et al.*, 1996] of the iterative closest point algorithm [Besl and McKay, 1992]. The registration from surface one to two, was used as the starting estimate for the registration from surface two to three and so on.

Subsequently, the first clipped surface was taken and registered to the remaining 55 sets of plainly lit video images using the proposed texture mapped tracking algorithm. This was repeated using the non-texture mapped algorithm of the previous chapter. The surface based, texture mapped and non texture mapped algorithm were compared by measuring the 3D error between the texture mapped and surface based transformations, and the non texture mapped and surface based registrations over the sequence of 55 sets of images.

Note that the surface based registration may have errors for two reasons. Firstly, the surface based registration minimises the distance between surfaces, which in itself does not guarantee a correct registration. Consider the case of registering a hemisphere to a sphere of equal radius. The distance between each surface could be zero, but there are still an infinite number of possible, incorrect registrations. However, surface based registration is widely used, and in this case where the two surfaces are generated by the same device, captured within twenty seconds of each other, and have featuredness or curvature like the face, should register well. Secondly, the reconstructed surface is formed from the images that were illuminated with the pseudo-random dot pattern. There is approximately a two to three second delay between the capture of the patterned images and the plainly lit images using the Tricorder^{TM} system. Therefore the volunteer could have moved between the capturing of these two sets of images, and so even if the surface based registration was perfect, it would never match the registration produced by the texture or non-texture mapped algorithms. It is assumed that the movement of the volunteer between the capturing of the images illuminated with the pseudo-random dot pattern and the images illuminated with the plain white light is small compared to errors in the registration algorithm because the time delay is small.

6.4.3.2 Results

Figure 6.14 illustrates the tracking algorithm. The left column represents the first frame in the tracking sequence, the middle column represents the 14th frame, and the right column represents the 36th frame. The Tricorder^{TM} system always captures four images at a time, one from each of four cameras. In this figure, all the images represent the images from the same view, i.e. the top left camera from the volunteers viewpoint. Images (a), (b), and (c) are the plainly lit video images to which the proposed texture mapped tracking algorithm registers. Images (d), (e) and (f) are the surface reconstructions created by the Tricorder^{TM} system, viewed from the same direction.

It can be seen that the surface in image (d) is aligned with image (a), surface (e) aligned with image (b) and surface (f) aligned with image (c). This is because the surfaces were reconstructed directly from similar patterned light video images, and so should fit well. In figure (g), the surface in green was clipped, and shown in red. This red surface was then registered to each reconstructed surface which included the surfaces shown in figures (e) and (f), using a surface based registration [Maurer Jr. *et al.*, 1996]. Figures (h) and (i) show that the surface based registration was successful at frame 14 and 36 as the red surface fits the green surface well. Figure (j) shows a wireframe representation of the surface overlayed on the first video image. Recall that the texture-mapped tracking algorithm matches the texture mapped surface directly to the video images, i.e. an intensity based match. Figure (k) shows that the texture mapped tracking algorithm works well up until frame 14, but figure (l) shows that at frame 36, the algorithm has failed.

The performance can also be assessed by measuring the 3D error between the surface based registration estimate for each video frame, and the texture mapped tracking estimate for each frame. This is shown in the graph in figure 6.15. The texture mapped tracking algorithm tracks well up until frame 14. After frame 14, the algorithm fails between frame 15 and 25, recovers between frame 25 and 31 and fails from 31 to 36. After frame 36 the algorithm was stopped, as the registration was lost. By comparison, the non-texture mapping algorithm fails completely as the error is always > 10 mm and after frame 25, the 3D error increases rapidly.

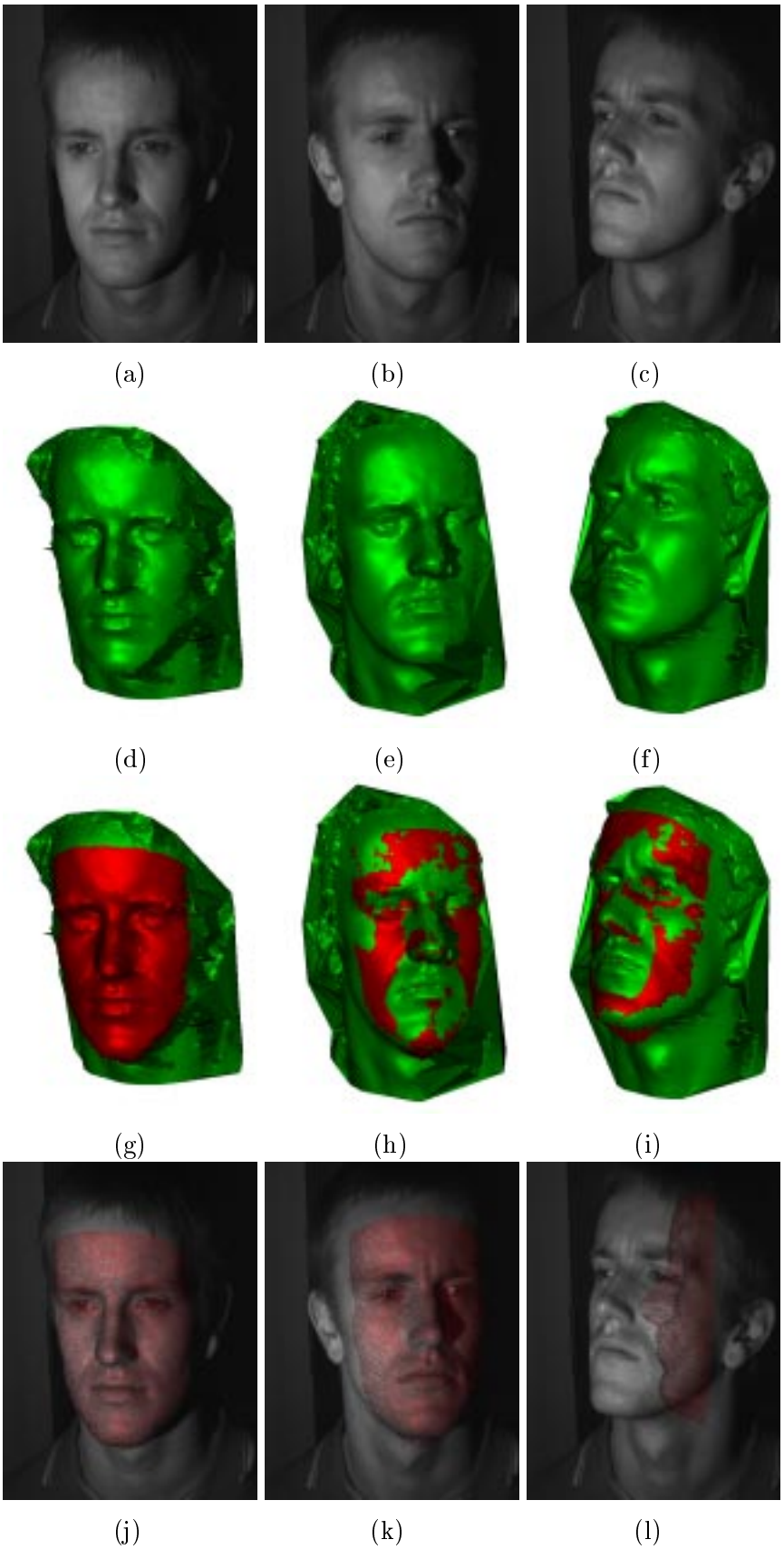
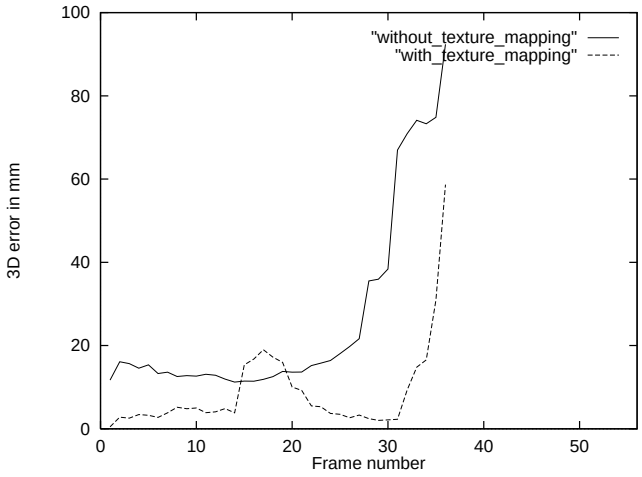


Figure 6.14: Comparing texture mapped and surface based tracking. See text, section 6.4.3.2.



(a)

Figure 6.15: Graph of 3D error in mm between the surface based tracking [Maurer Jr. *et al.*, 1996] and the proposed texture mapped tracking.

6.4.3.3 Conclusions

In the previous experiment it was concluded that the algorithm failed to track because the change in registration transformation between video frames was too large. Here, more care was taken to make the transformation between each frames small. The head movement of the volunteer consisted of a rotation to the left, rotation up, rotation to the right, rotation down, and rotation back to the centre position. The texture from the initial image was mapped onto the surface and the initial surface used to track throughout the sequence. As the video images are taken with one single plain white light source, there is noticeable shading. The illumination of each surface point on the volunteers face will change as he moves relative to the camera. The texture mapped onto the initial surface however, will not. It was concluded that the algorithm failed when the volunteer's head had rotated too far to the left, and to the right. Frame 14, was where the algorithm tracked to, which represents a total rotation of 25 degrees to the left. Frame 25 - 31 represented rotations of 17 - 19 degrees to the right. Therefore it can be concluded that this algorithm only works well for rotations of approximately $\pm$ 20 degrees from the initial position. Beyond this, the shading on the texture map is too different to match to the next video image.

6.5 A Comparison With Other Methods

There exists similar algorithms to the method proposed in this chapter in the low bitrate image coding literature. Steinbach describes a motion analysis and segmentation algorithm of video images for model-less based image coding [Steinbach *et al.*, 1998]. The algorithm uses two successive video frames and reconstructs an unstructured dense set of points using a structure from motion algorithm. The image texture is then mapped onto the points, and used to track these points in subsequent video frames. However Steinbach's method first has to reconstruct a model of the scene. For the cases described in this thesis, an accurate 3D model of the object is available from a 3D medical image. The tracking method uses optical flow, evaluated at multiple resolutions, to compute the apparent motion from frame to frame, whereas the method in this chapter uses a simple extension of the previous registration algorithm. It is difficult to compare these two methods as Steinbach's method relies on a good initial reconstruction, and can only explicitly compute five of the six extrinsic parameters, and the sixth only up to a scale factor. This makes it difficult to verify the actual tracking accuracy. Care has been taken in this chapter to develop a tracking algorithm suitable for medical applications, and to compare its accuracy to the best currently available techniques. Future work might include ways of taking optical flow methods to estimate apparent motion and use it to speed up the tracking algorithm in this chapter.

LaCascia also develops a method for tracking using texture mapping [LaCascia *et al.*, 1998] for possible video conferencing or image coding applications i.e. tracking faces. LaCascia however can only approximate the shape of the face using a cylinder. Again, no careful validation was performed. The algorithm is different to that proposed in this chapter as LaCascia performs the registration in texture map space. The first video image is aligned with the cylinder model, and then a texture map generated by unwarping the cylinder model and storing the resultant image. Subsequent video images are applied to the model using an estimate of the extrinsic camera parameters, and then the model is unwarped and compared to the previous texture map. This means that both images being registered are warped by an unrealistically simplified transformation, which could make LaCascia's algorithm less accurate than the one proposed here.

6.6 Summary

This chapter has described a new tracking algorithm that uses texture mapping to register sequences of multiple video images to a 3D surface model derived from MR/CT. The algorithm was tested with simulated data. This achieved registration with a mean 3D error of 1.05 mm for stereo views. The mono tracking experiments with the volunteer (section 6.4.2) showed that tracking performance is poor if only one camera is used. However tracking was possible by using two camera views. The tracking was tested over a range of motion that might be encountered during for example a neurosurgical or ENT procedure without head immobilization. This work uses a simple gradient ascent search method to maximise the mutual information. This could be improved by using predictive methods such as the Kalman filter. The experiment in section 6.4.3 illustrates that an interesting topic of research would be to investigate whether the texture map could be updated throughout the tracking. Alternatively, it may be necessary to adjust the intensities in the texture map to compensate for changes in shading on the surface of the object of interest, as it moves relative to the light source.

This tracking algorithm typically took about 2-3 minutes to register to each frame. This is quicker than the registration time for the previous two chapters, as the algorithm always starts close to the solution. This algorithm must also perform texture mapped rendering, which is significantly slower than non-texture mapped rendering.

Finally, it can be concluded that the use of texture mapping to assist a tracking algorithm does significantly improve the accuracy and robustness of mutual information based tracking when compared with the non-texture mapping algorithm of the previous two chapters. A subset of this work was presented in [Clarkson *et al.*, 1999b; Clarkson *et al.*, 1999c].

Chapter 7

Photo-Consistency, A Novel Measure Of Image Alignment

7.1 Introduction

Previous chapters have described an algorithm which registers a 3D model to one or more video images. Registration was achieved by producing renderings of the 3D model, and comparing these to the video images using mutual information. The results have shown that the algorithm can register video images with a surface model of a plastic skull phantom with good accuracy [Clarkson *et al.*, 1998; Clarkson *et al.*, 1999a]. However it proved difficult to achieve the same performance when registering video images to surface models of, for example, a human face. In this chapter a new similarity measure is proposed that does not in any way require rendered image intensities. The algorithm requires at least two calibrated video cameras and so cannot be compared to the mono view registration performance in chapter 4. It is in this chapter that the most significant, and novel research of this thesis is described.

7.2 Aim

The aim of this chapter is to investigate whether a similarity measure can be developed which registers a 3D model to two or more optical images without requiring a rendered image to be calculated and whether using such a similarity measure provides more accurate and more reliable registration than the rendering based method of previous chapters.

7.3 Theory

The algorithm described in chapters 4 and 5, registers a 3D model of a plastic skull phantom to one or more video images, and works accurately and robustly. However, as the algorithm produces a rendering of the 3D surface, there are several underlying assumptions:

- The surface has constant albedo or reflectance. This means that it should have no texture or varying colour.
- The surface can be rendered using a simple lighting model, to look sufficiently similar to the video image. Defining 'sufficiently similar' is non-trivial and any lighting model must be overly simple to satisfy a trade off between realism and computational cost.
- Only one surface type is present. i.e. a skin or bone surface

It has also been shown that an accurately calibrated light source position is required to reduce the effects of false maxima or minima in the cost function which lead to poor registration accuracy.

7.3.1 Shape Reconstruction

A fundamental problem in computer vision is the reconstruction of a 3D scene from sensors such as video cameras, or range sensors. A recent paper by Kutulakos and Seitz demonstrates a new method of shape reconstruction [Kutulakos and Seitz, 1998]. The algorithm requires that the scene or shape being reconstructed is finite and opaque. The scene should be imaged by N video cameras, where each camera is calibrated to some world coordinate system. The algorithm proceeds by defining a starting volume, e.g. a cube, which must contain the shape. Through a series of sweeps through the volume, it discards or 'carves away' any voxels which are not 'photo-consistent'. A voxel is called photo-consistent based on the following method: The scene radiance is assumed to follow a locally computable lighting model, e.g. Lambertian. Locally computable means that shadows, inter-reflections and transparencies are not allowed. If the camera configuration is known, i.e. the intrinsic and extrinsic camera parameters are known, then for each 3D voxel, the corresponding 2D pixel coordinate in each of the N video images can be calculated. For all the images that the voxel is visible in, the intensities at each of the projected points should be consistent, i.e. agree with the assumed reflectance model. The

authors assume that the object they are reconstructing exhibits Lambertian reflection. The Lambertian reflectance model states that the observed intensity depends on the cosine of the angle between the surface normal and the vector to the light source, not on the angle between the surface normal and the direction to each video camera (see section 2.5.2). Thus if a point on the reconstructed surface is projected into each of the N video images, and the video image intensity is read, then apart from image noise, the resulting image intensities should be identical. Thus a suitable consistency checking function can take a 3D voxel, calculate the standard deviation of the intensity values at each of the projected 2D pixel locations and discard the surface voxel as non-photo-consistent if the standard deviation is above a threshold. Algorithmic details need to ensure that voxels are visited in the correct order, but the key point is the 'consistency checking' function by which a surface voxel is deemed photo-consistent or not. The algorithm reconstructs a maximally photo-consistent shape. This is illustrated in figure 7.1. Figure 7.1(a) shows a real object, imaged by two video cameras. Figure 7.1(b) shows an example reconstructed surface. The real object was a circle, and so the reconstructed object will have a circular front nearest the cameras. However, the algorithm can say nothing about the voxels that are occluded, and so these are kept. Thus the reconstructed shape is the maximally (biggest) photo-consistent shape and must include all other shapes that are photo-consistent under the assumed lighting model. Figure 7.1(c) shows that adding further cameras improves the quality of the reconstruction as each new video image places constraints on the allowed shape. This algorithm reconstructs using a least commitment principle, as voxels are only removed if they are definitely *not* photo-consistent. Thus the reconstructed shape will be the maximal shape assuming no *a priori* information of what the object should look like. The quality of the reconstruction is determined by the voxel size used, and how well the real scene does fit the chosen lighting model.

To summarise, this method takes a set of video images and produces a shape which is registered to each video image. It does not, however, make any prior assumptions of what the shape should look like. The registration problem is to take a known shape and register it to one or more video images. The ideas of Kutulakos and Seitz provided the inspiration for a new similarity measure which is described below.

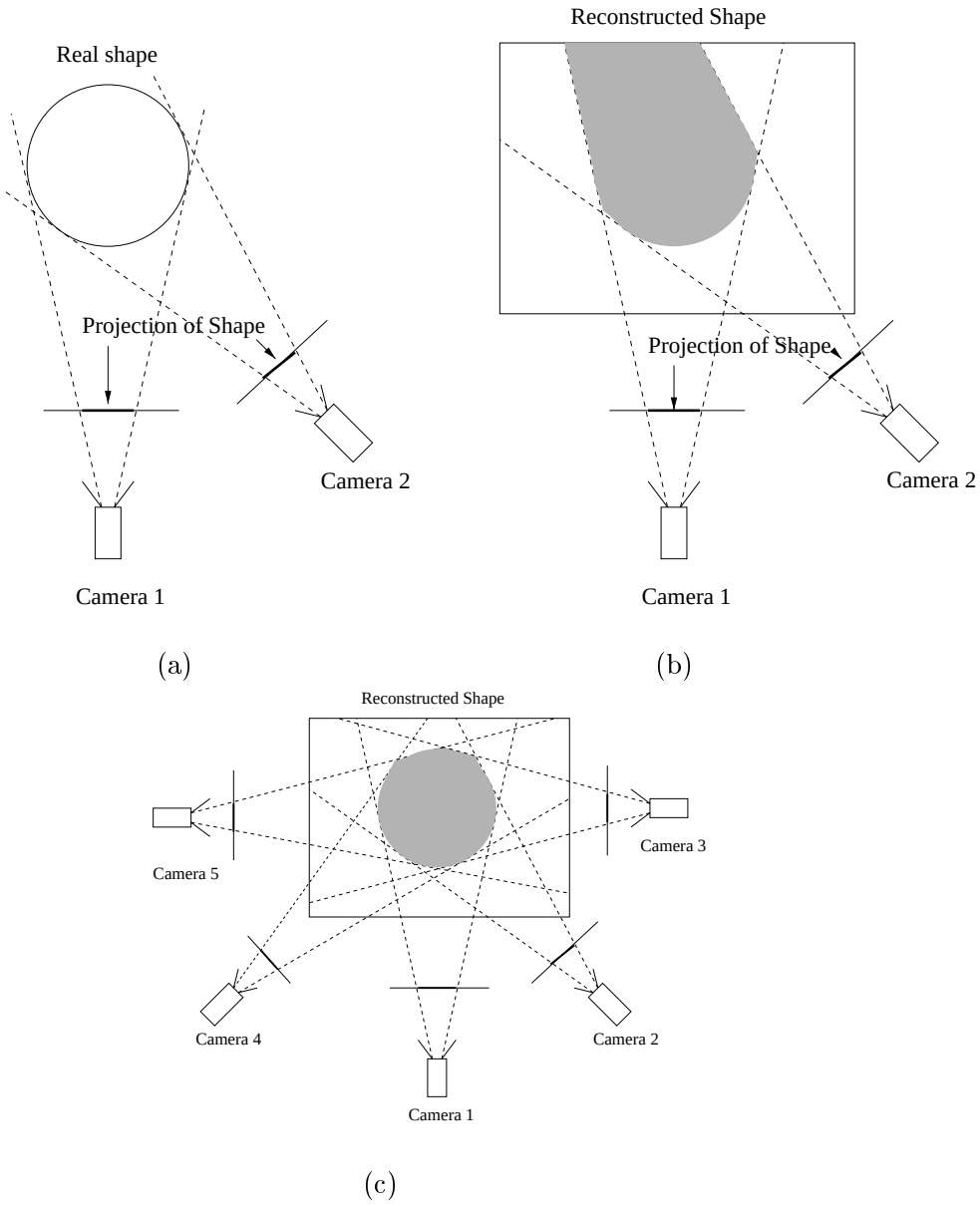


Figure 7.1: (a) Two cameras take images of a real object. (b) A maximal photo-consistent shape is reconstructed. (c) With more views, a more accurate model is produced. See text section 7.3.1.

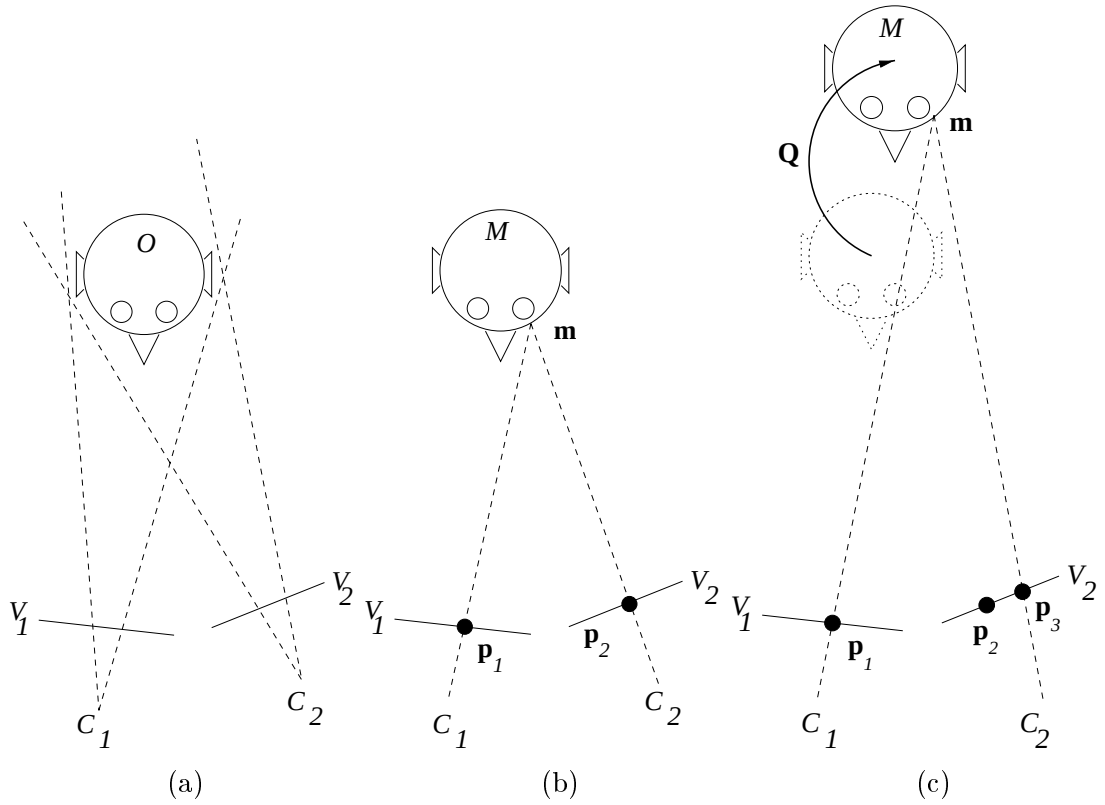


Figure 7.2: Diagram to illustrate the photo-consistency measure for registration. See text section 7.4.1.

7.4 Methods

7.4.1 A New Similarity Measure

The similarity measure arose from the observation that if photo-consistency can be used to deduce which points on a surface are consistent with N video cameras, then given an accurately defined surface, photo-consistency might be used as a measure of alignment to the N video images.

Figure 7.2 illustrates how the similarity measure works.(a) Two video cameras C_1 and C_2 produce video images V_1 and V_2 of a real object O. (b) Each model point $\mathbf{m}$ of a surface model M projects onto image point $\mathbf{p}_1$ and $\mathbf{p}_2$. If the model is registered to the video images, then the intensity values at $\mathbf{p}_1$ and $\mathbf{p}_2$ should be photo-consistent. (c) If the model is misregistered by a transformation $\mathbf{Q}$, then model point $\mathbf{m}$ projects onto $\mathbf{p}_1$ and $\mathbf{p}_3$, which are likely to be less photo consistent than case (b).

7.4.2 The Consistency Checking Criteria

The consistency check function has the task of describing how consistent the set of N pixels is, across N views. For simplicity, consider a point which is not occluded in any of the N views. To define a consistency checking function, lighting and camera geometry must be considered and also the reflectance of the surface. Several consistency check functions are illustrated by example.

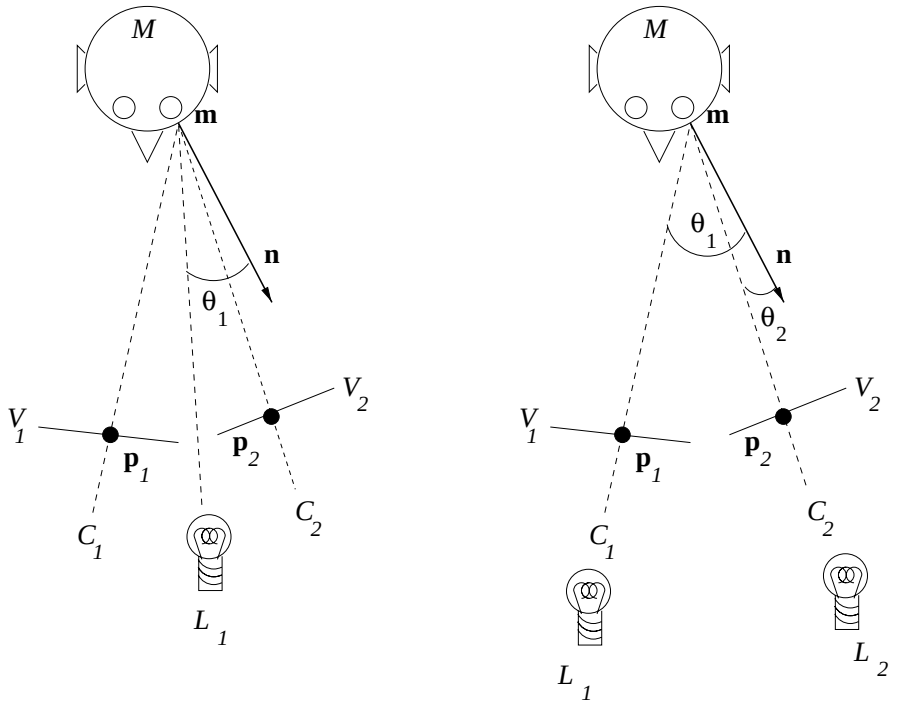
7.4.2.1 Calibrated Cameras, Uncalibrated Lights, Lambertian Reflectance

Consider the case where N video cameras take images of an object, and where the extrinsic and intrinsic parameters of each camera are known. Furthermore assume that one or more lights are present, where their positions are not known, but they are fixed relative to the object. If the surface is assumed to exhibit Lambertian reflection, the reflectance at any point on the surface depends on the cosine of the angle θ_1 between the light source and the surface normal, not on the direction to the camera (see section 2.5.2). This is illustrated in figure 7.3(a). In this case the image intensities at $\mathbf{p}_1$ and $\mathbf{p}_2$ should be identical, apart from image noise.

7.4.2.2 Calibrated Cameras, Calibrated Co-Axial Lights, Lambertian Reflection

Consider the case in figure 7.3(b). Video image V_1 is taken using only light source L_1, which is rigidly attached to camera C_1 and emits light co-axial to the camera viewing direction. Video image V_2 is taken using only light source L_2, which is similarly attached to camera C_2. This second image V_2 could be taken by moving the first camera C_1 and light L_1 and tracking it. Assuming that the model M is registered to the video images, and that the real surface exhibits Lambertian reflectance, then model point $\mathbf{m}$ will project to a bright pixel in V_2 as the angle θ_2 is small. However it will appear dimmer in image V_1 as angle $\theta_1 < \theta_2$. In this case, the surface normal $\mathbf{n}$ at each model point $\mathbf{m}$ must be used to calculate angles θ_1 and θ_2.

Let the surface colour at model point $\mathbf{m}$ be I_a, and let the light source have an intensity of 1. If a Lambertian model is assumed, the observed intensity v_1 in video image V_1 should be $v_1 = I_a \cos\theta_1$. Likewise the image intensity v_2 in image V_2 should be $v_2 = I_a \cos\theta_2$. Therefore, given two pixel intensities, in two images, the following squared distance



(a) (b)

Figure 7.3: (a) Calibrated cameras, uncalibrated lights, Lambertian reflectance. (b) Calibrated cameras, calibrated lights aligned co-axially with cameras, Lambertian reflection (see text).

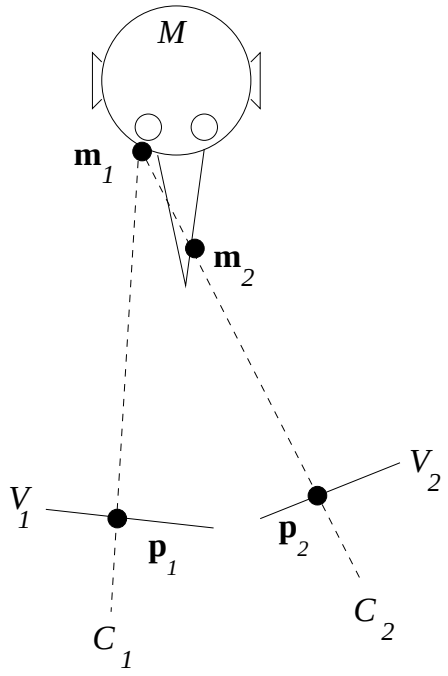
$d^2(v_1, v_2)$ in intensity space should be minimised at registration

$$d^2(v_1, v_2) = (v_1/\cos\theta_1 - v_2/\cos\theta_2)^2 \tag{7.1}$$

If either θ_1 or θ_2 approach 90 degrees then equation (7.1) becomes unstable, and so a threshold must be used to define a limiting angle e.g. 45 degrees.

7.4.2.3 Other Camera, Light And Reflectance Scenarios

The previous two examples illustrate the idea that if a lighting model is assumed, and knowledge of the pose of cameras and lights relative to each other, then a measure of consistency between image intensities can be defined. In principle this model can be any locally computable lighting model i.e. a model that does not include shadowing, transparency or inter-reflections. Thus in principle, it would be possible to include the effects of specular reflection into the consistency calculations. Specular reflection occurs when the angle between the surface normal and the vector to the light source is approximately equal and opposite to the angle between the surface normal and the



(a)

Figure 7.4: It is necessary to determine which points are visible in which views. In (a), model point $\mathbf{m}_1$ projects to pixel $\mathbf{p}_1$ in image V_1, but does not project to pixel $\mathbf{p}_2$ in image V_2 as the model point $\mathbf{m}_2$ is closer to camera C_2 than point $\mathbf{m}_1$.

vector to the viewer or camera. If the camera and light were positioned so that a point was imaged with specular reflection, then this point could, for instance, be ignored as specular reflection is usually a similar colour as the light source rather than the material being imaged. The images in this chapter are either simulated or are of a volunteer's skin surface. The former has Lambertian reflection by construction, and the experiments in this chapter show that a volunteer's skin surface is well approximated by a Lambertian reflectance model.

7.4.3 Eliminating Occluded Points

It is also necessary to determine which points are visible in which views. Figure 7.4 illustrates the problem. Model point $\mathbf{m}_1$ projects to pixel $\mathbf{p}_1$ in image V_1, but does not project to pixel $\mathbf{p}_2$ in image V_2 as the model point $\mathbf{m}_2$ is closer to camera C_2 than point $\mathbf{m}_1$. Thus each point must be checked for visibility before using it to calculate a similarity measure. A given point must project onto at least two video images in order for any kind of consistency checking to be possible. For the experiments in this chapter, except the last, the algorithm was implemented so that if a point did not project onto all the available video views, it was ignored.

7.4.4 Similarity Measures Based On Photo-Consistency

With a surface model, and knowledge of the geometry of the lighting and camera arrangement, a cost function for a given pose must be defined. First, two cost functions are described for the scenario described in section 7.4.2.1, where assuming Lambertian reflection, and the fact that a single light source position is fixed with respect to the cameras, the intensity of a projected model point should be identical in each view.

Let the set of all optical images be denoted by V_n where $n = 1 \dots N$ is an index labelling each optical image. Let the set of all model surface points that are visible in all optical views be denoted by $\mathbf{m}_i$ in homogeneous coordinates, where $i = 1 \dots I$ is an index labelling these I points. To evaluate the similarity measure, each model point is projected into each optical image using

$$k\,\mathbf{p}_{i,n} = \mathbf{M}_n\,\mathbf{m}_i \tag{7.2}$$

Here, $\mathbf{p}_{i,n}$ is a homogeneous coordinate in optical image n, projected from model surface point i, $\mathbf{M}_n$ is the 3×4 perspective projection matrix, calculated from the extrinsic and intrinsic parameters of optical image n, which projects $\mathbf{m}_i$ onto $\mathbf{p}_{i,n}$ and k is a homogeneous scale factor. The optical image intensity at point $\mathbf{p}_{i,n}$ is given by $v_{i,n}$. The arithmetic mean $\overline{v_i}$ of the pixel values associated with a given point is calculated as

$$\overline{v_i} = \frac{1}{N}\sum_{n=1}^{N} v_{i,n} \tag{7.3}$$

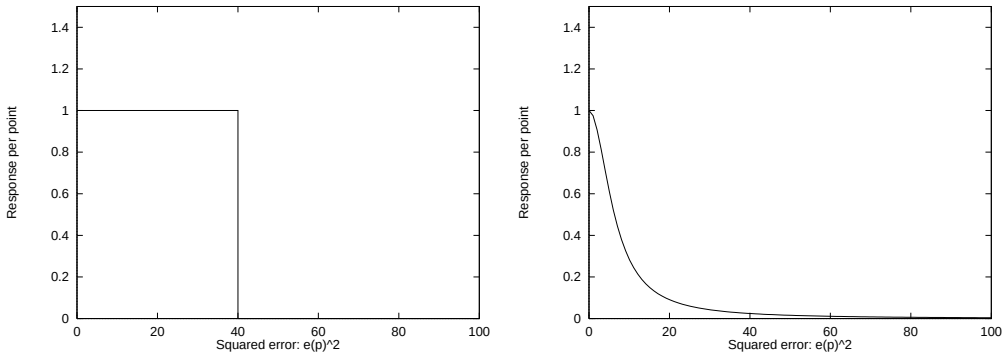
and the mean sum of squared differences

$$e_i^2 = \frac{1}{N-1}\sum_{n=1}^{N}(v_{i,n} - \overline{v_i})^2 \tag{7.4}$$

A similarity measure, the sum of squared differences of photo-consistency, $\mathrm{PC}_{\mathrm{squared}}$ can now be defined as

$$\mathrm{PC}_{\mathrm{squared}} = \frac{1}{I}\sum_{i=1}^{I} e_i^2 \tag{7.5}$$

In other words, a point in the surface model is projected into each optical image, and the intensity read and the squared error e_i^2 calculated. The similarity measure, $\mathrm{PC}_{\mathrm{squared}}$, is the sum of the squared error evaluated for each model point in the surface and normalised (divided) by the number of points. With more than two optical images, $\mathrm{PC}_{\mathrm{squared}}$ would be the sum of the variance of the intensity values that a given 3D point projects to. An alternative measure would be to set a threshold e on the squared error and define



(a) (b)

Figure 7.5: Graphs of (a) $\mathrm{PC}_{\mathrm{good}}$ and (b) $\mathrm{PC}_{\mathrm{inverse}}$ for a threshold e, where $e^2 = 40$.

whether a set of pixel intensities were consistent or not. Defining a function $Good(i)$ as

$$Good(i) = \begin{cases} 1 & : \quad e_i^2 < e^2 \\ 0 & : \quad e_i^2 >= e^2 \end{cases} \tag{7.6}$$

an alternative cost function, the sum of good photo-consistent points, $\mathrm{PC}_{\mathrm{good}}$ can be defined as

$$\mathrm{PC}_{\mathrm{good}} = \frac{1}{I} \sum_{i=1}^{I} Good(i) \tag{7.7}$$

However, consider the graphs in figure 7.5(a) and (b). Graph (a) shows the response for a single point, using $Good(i)$ from equation 7.6. It was felt that a more continuous response function for each point would be preferable, as it would provide a smoother overall cost function, more suitable for optimisation. Thus, another similarity measure, the sum of inverse squared differences of photo-consistency, $\mathrm{PC}_{\mathrm{inverse}}$ is defined

$$\mathrm{PC}_{\mathrm{inverse}} = \frac{1}{I} \sum_{i=1}^{I} \frac{e^2}{e^2 + e_i^2} \tag{7.8}$$

where e is again a threshold, and e_i^2 is the squared error defined earlier in equation (7.4). The response per point for this function is shown in figure 7.5(b). The value of e can be set to a value calculated from the typical noise level for image intensity values.

A similarity measure can be defined for the scenario described in section 7.4.2.2, where a light source exists for each camera, or a light source is fixed to the camera, and hence moves with the camera as it is tracked. In this case, the mean of the distance function defined in equation 7.1 should be zero. Thus the squared error can be calculated as

$$e_i'^2 = \frac{1}{(N-1)^2} \sum_{n=1}^{N} \sum_{m=1}^{N} d^2(v_{i,n}, v_{i,m}) \tag{7.9}$$

The similarity measure $\mathrm{PC}_{\mathrm{squared}}$ is then modified for this different lighting scenario, and is denoted using $\mathrm{PC}'_{\mathrm{squared}}$. Let

$$\mathrm{PC}'_{\mathrm{squared}} = \frac{1}{I}\sum_{i=1}^{I} e_i'^2 \tag{7.10}$$

Again, using the threshold e, and the squared error measure in equation (7.9), the similarity measure $\mathrm{PC}_{\mathrm{inverse}}$ is also modified for this different lighting scenario, and is denoted using $\mathrm{PC}'_{\mathrm{inverse}}$. Let

$$\mathrm{PC}'_{\mathrm{inverse}} = \frac{1}{I}\sum_{i=1}^{I} \frac{e^2}{e^2 + e_i'^2} \tag{7.11}$$

Other similarity measures can be defined. The common framework would be an assumed lighting model, a measure of how photo-consistent the intensities are for a given point projected into each view, and an overall similarity measure. The lighting model can in principle be any locally computable lighting model i.e. no transparency, no shadows or inter-reflections. The measure of consistency will be based on the assumed lighting model, and the relative position of the lights and cameras. The overall similarity measure can be based around a sum of squared error, variance, or robust estimator.

This new idea, using photo-consistency between video views can be combined with the information theoretic framework presented in the previous chapters. Mutual information can be used to measure how consistent the video image intensities are that all the points project to. Each model point $\mathbf{m}_i, i = 1 \ldots I$ can be projected into the N video images using equation 7.2, resulting in N pixel coordinates $\mathbf{p}_{i,n}, n = 1 \ldots N$ and N corresponding pixel intensities $v_{i,n}, n = 1 \ldots N$. These N pixel intensities are then plotted in an N dimensional histogram where each dimension represents intensities from video image V_n. Let $\mathcal{V}_n, n = 1 \ldots N$ be random variables representing the probability distribution of the pixel intensity values from video image $V_n, n = 1 \ldots N$ respectively. The mutual information of $\mathcal{V}_n, n = 1 \ldots N$ can then be calculated as

$$\mathrm{PC}_{\mathrm{mutual}} = I(\mathcal{V}_1; \mathcal{V}_2; \ldots \mathcal{V}_N) = H(\mathcal{V}_1) + H(\mathcal{V}_2) + \ldots + H(\mathcal{V}_N) - H(\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_N) \tag{7.12}$$

To distinguish this method from the method of chapters 4 to 6, this similarity measure is called $\mathrm{PC}_{\mathrm{mutual}}$. In chapter 5, three methods for calculating mutual information for more than two random variables were examined. The first was by using high-dimensional histograms where the number of random variables equals the number of histogram dimensions necessary to calculate the mutual information (section 5.3.1.1). The second

Abbreviation	Description	Equation
$PC_{squared}$	Sum of squared differences of photo-consistency with a light fixed relative to the object	(7.5)
PC_{good}	Sum of good, photo-consistent points with a light fixed relative to the object	(7.7)
$PC_{inverse}$	Sum of inverse squared differences of photo-consistency with a light fixed relative to the object	(7.8)
$PC'_{squared}$	As $PC_{squared}$ but with a light fixed to each camera	(7.10)
$PC'_{inverse}$	As $PC_{inverse}$ but with a light fixed to each camera	(7.11)
PC_{mutual}	Photo-consistency, measured with mutual information	(7.12)

Table 7.1: A summary of the notation used for the six photo-consistency based similarity measures

method was to add the mutual information of video and rendered image pairs, and the third method was to combine all the information from video and rendered images into one 2D histogram. These latter two methods were legitimate approximations for two reasons. Each video image had a corresponding rendered image and care was taken to make sure that the distribution of rendered image intensities and video image intensities was similar for each pair of images.

When using mutual information as a similarity measure to measure the photo-consistency of the sets of video intensities that a given set of points projects to, no such assumptions can be made. Therefore the mutual information should be calculated by first computing an N dimensional joint probability distribution. As mentioned in chapter 5 this will be increasingly unreliable and computationally expensive as N increases. Therefore, PC_{mutual} will only be suitable when N is small. However, PC_{mutual} does not place any constraints on the functional relationship between video image intensities and hence may have the advantage of potentially wide applicability. Equation 7.8 defines $PC_{inverse}$ for when a surface has Lambertian reflection, and the light source is fixed relative to the N cameras. Equation 7.11 however, defines a similar function $PC'_{inverse}$ for when a surface has Lambertian reflection, and each optical image is taken with a light source attached to the camera. Thus a different function is necessary for each lighting arrangement. With PC_{mutual}, the potential exists to measure the consistency of any statistical relationship between the video intensities that a point projects to. So, in the experiments that follow, PC_{mutual} is tested for two different lighting scenarios on simulated images, where it would be expected that PC_{mutual} would not be affected by the changing lighting conditions. Table 7.1 summarises the naming of the photo-consistency based similarity measures.

7.5 Experiments

The following experiments were performed.

- **Simulations.** Using the minimum of two camera views, several images were created by overlaying a rendered image into a video image. Different lighting conditions, and surface textures were used to demonstrate the algorithm. With misregistration size $\delta t = \pm 4$ mm and degrees, each of the photo-consistency based similarity measures were used to register a surface model of a skull phantom to various images. As the results show, $\mathrm{PC}'_{\mathrm{squared}}$, $\mathrm{PC}'_{\mathrm{inverse}}$ and $\mathrm{PC}_{\mathrm{mutual}}$ did not work well, and so they were also tested for misregistration sizes of $\delta t = \pm 1, 2$ and 3 mm and degrees. See section 7.5.1.

- **Tricorder™ Surface To Four Video Images.** With misregistration sizes of $\delta t = \pm 4, 8, 12$ and 16 mm and degrees, and for similarity measures $\mathrm{PC}_{\mathrm{squared}}$ and $\mathrm{PC}_{\mathrm{inverse}}$, a Tricorder™ reconstructed surface model of the face of a volunteer was registered to four video images. See section 7.5.2.

- **MR Surface To Four Video Images.** Subsequently a surface model was extracted from an MR scan. With misregistration sizes of $\delta t = \pm 4, 8, 12$ and 16 mm and degrees, and for similarity measures $\mathrm{PC}_{\mathrm{squared}}$ and $\mathrm{PC}_{\mathrm{inverse}}$, the MR surface was registered to the same four video images of section 7.5.2. See section 7.5.3.

- **Tricorder™ Or MR Surface To Two Video Images Using** $\mathrm{PC}_{\mathrm{inverse}}$. Two video images is the minimum number of images for photo-consistency based registration. For misregistration sizes of $\delta t = \pm 4, 8, 12$ and 16 mm and degrees, six different pairings of the video images in section 7.5.2 were registered to the Tricorder™ scan of section 7.5.2 and the MR scan of section 7.5.3 using $\mathrm{PC}_{\mathrm{inverse}}$. See section 7.5.4.

- **Tricorder™ Or MR Surface To Two Video Images Using** $\mathrm{PC}_{\mathrm{mutual}}$. For misregistration sizes of $\delta t = \pm 4, 8, 12$ and 16 mm and degrees, different combinations of two video images taken from section 7.5.2 were registered to the Tricorder™ scan of section 7.5.2, and the MR scan of section 7.5.3 using $\mathrm{PC}_{\mathrm{mutual}}$. See section 7.5.5.

- **Robustness To Noise.** Zero mean Gaussian additive noise with standard deviation $\sigma = 1, 2, 4, 8, 16, 32$ and 64 intensity values was added to the four video images (video images are 8 bit = 256 intensity values). For misregistration sizes of $\delta t = \pm 8$ mm and degrees, the noise corrupted video images were registered to the MR scan used in section 7.5.3, using $\mathrm{PC}_{\mathrm{inverse}}$. See section 7.5.6.

- **Surface Resolution And Z-Buffer Requirements.** The MR surface models were repeatedly registered to the four video images of section 7.5.2 using $\mathrm{PC}_{\mathrm{inverse}}$. The algorithm was tested with respect to the number of points used and the frequency of z-buffer checking. These results do depend on the search strategy, but give important qualitative information concerning the possible speed of the algorithm, as reducing points and z-buffer checking greatly decreases the computational cost. See section 7.5.7.

- **Registering 10 Tricorder$^{\mathrm{TM}}$ And 5 MR Datasets** For misregistration sizes of $\delta t = \pm 8$ mm and degrees, 15 different surfaces, taken from volunteers were registered to their corresponding video images using $\mathrm{PC}_{\mathrm{inverse}}$. See section 7.5.8.

- **A Comparison With A Surface Based Registration Algorithm** To conclude this chapter, the tracking experiment from section 6.4.3 was repeated, this time, the photo-consistency based registration algorithm was compared with the surface based algorithm. This completes the comparison of the non-texture mapping algorithm from chapter 5, the texture mapping algorithm from chapter 6, the photo-consistency based algorithm from this chapter, with a surface based registration algorithm [Maurer Jr. *et al.*, 1996]. See section 7.5.9.

7.5.1 Simulations

7.5.1.1 Methods

Eight images were created. The images contained a real video image background and a rendering of a skull phantom surface at a known pose. Figure 7.6 shows these images. The left column of images are the left camera view and the middle column, the right camera view. The right column of images is a diagram showing the lighting arrangement used to produce the rendered image. The surface was rendered with Lambertian reflection.

Figures 7.6 (a) and (b) are two images, each showing a rendering of the skull phantom, used throughout chapter 4, overlaid on a video image. The diagram (c) illustrates that the rendering light source used when rendering images (a) and (b) was aligned with the rendering camera for each view. In this case, $\mathrm{PC}'_{\mathrm{squared}}$ and $\mathrm{PC}'_{\mathrm{inverse}}$ should be used to register the surface model to these two images. Figures 7.6 (d) and (e) are two images, each showing a rendering of the same skull phantom overlaid on the same video image. The diagram (f) illustrates that the rendering light source used when rendering images (d) and (e) was positioned in between each camera. Here, $\mathrm{PC}_{\mathrm{squared}}$ and $\mathrm{PC}_{\mathrm{inverse}}$ should be used to register the surface model to these two images. Figures (g) and (h) are produced with the same rendering light and camera positions as (a) and (b) but with an additional random dot pattern texture mapped to the surface of the skull phantom. Figures (i) and (j) are produced with the same rendering light and camera positions as (d) and (e) but with an additional random dot pattern texture mapped to the surface of the skull phantom.

For image pairs (d)(e) and (i)(j), and similarity measures $\mathrm{PC}_{\mathrm{squared}}$ and $\mathrm{PC}_{\mathrm{inverse}}$, misregistration sizes of $\delta t = \pm 4$ mm and degrees were added to the gold standard extrinsic parameters and the algorithm used to register the surface model to the video images. For image pairs (a)(b) and (g)(h), and similarity measures $\mathrm{PC}'_{\mathrm{squared}}$ and $\mathrm{PC}'_{\mathrm{inverse}}$, misregistration sizes of $\delta t = \pm 4$ mm and degrees were added to the gold standard extrinsic parameters and the algorithm used to register the surface model to the video images. As will be seen in the results in section 7.5.1.2, this did not work well, and so misregistration sizes of $\delta t = \pm 1, 2$ and 3 were also tested for $\mathrm{PC}'_{\mathrm{squared}}$ and $\mathrm{PC}'_{\mathrm{inverse}}$. For image pairs (a)(b), (d)(e), (g)(h) and (i)(j), and similarity measure $\mathrm{PC}_{\mathrm{mutual}}$, misregistration sizes of $\delta t = \pm 1, 2, 3$ and 4 mm and degrees were added to the gold standard extrinsic parameters and the algorithm used to register the surface model to the video images.

As before, successful registrations are defined as those where all the extrinsic parameters finished nearer the gold standard values than the initial displacement (δt) used to test the algorithm. For successful registrations, the mean and standard deviation, projection error and 3D error were calculated using every point on the surface model. The projection error was measured with respect to each video view and the arithmetic mean of the projection errors for each view taken as an overall projection error.

The similarity measures were implemented using VTK [Schroeder *et al.*, 1997], utilising the OpenGL$^{\text{TM}}$ graphics libraries. The surface model is defined by polygons. Thus the surface can be rendered using standard computer graphics techniques. OpenGL computes a z-buffer whilst rendering [Foley *et al.*, 1990]. The 'z' refers to the z_c coordinate mentioned in section 2.2.3, and is the distance of a point from the camera's centre of projection along an axis parallel to the camera's optical axis. The z-buffer is an image, the same size as the eventual rendered image, where each pixel describes the distance of the closest point to the camera. Therefore, if a rendering is produced, the z-buffer can be stored. In contrast to the previous chapters, the rendered image intensities are not used. These new similarity measures only use depth information and the video image intensities.

Referring to figure 7.4, when a point $\mathbf{m}_1$ is projected into image V_2, the distance from the camera C_2 can be calculated, and compared with the z-buffer. If the value is the same, then $\mathbf{m}_1$ can be used to calculate the similarity measure. In this case however, the value in the z-buffer would be the distance of point $\mathbf{m}_2$ because this point is closer to camera C_2 than $\mathbf{m}_1$. Therefore, $\mathbf{m}_1$ is not visible in image V_2. This method requires a z-buffer to be computed for each image, and for each iteration and may be unnecessary. Section 7.5.7 studies the performance with respect to the number of points used and how often z-buffering is performed. The algorithm remains the same as chapter 5 except for the similarity measures, and the amount of blurring. Initial tests for these new similarity measures omitted the blurring to see if the blurring was in fact necessary. The algorithm in this chapter performs better than in the previous chapters and the blurring was simply left out. For $\text{PC}_{\text{mutual}}$, the histogram size was set to 64×64.

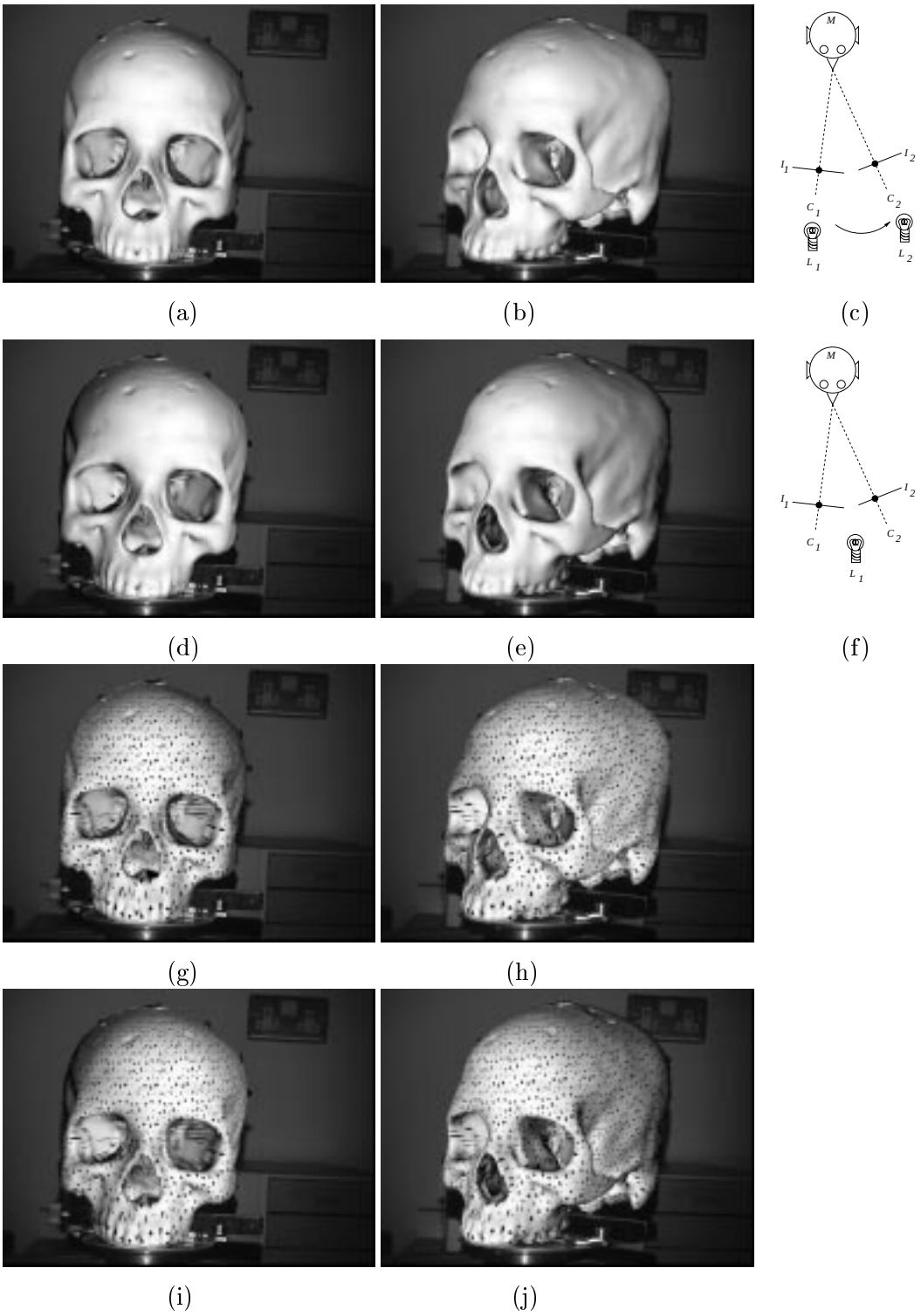


Figure 7.6: Simulation images (see text, section 7.5.1).

7.5.1.2 Results

Table 7.2 shows the mean and standard deviation projection and 3D errors for $\mathrm{PC}'_{\mathrm{squared}}$ and $\mathrm{PC}'_{\mathrm{inverse}}$. The images (a), (b), (g) and (h) were produced with a rendering light source that was aligned coaxially with each rendering camera. See figure 7.6(c). For misregistration sizes of $\delta t = \pm 1, 2, 3$ and 4 mm and degrees, the similarity measures $\mathrm{PC}'_{\mathrm{squared}}$ and $\mathrm{PC}'_{\mathrm{inverse}}$ work poorly. Given that these images are simulations, the accuracy and success rate are low. $\mathrm{PC}'_{\mathrm{squared}}$ can be seen to be working significantly better than $\mathrm{PC}'_{\mathrm{inverse}}$. Both similarity measures have a peak at the correct registration, but around the registration area the search space is rough.

Table 7.3 shows the performance of $\mathrm{PC}_{\mathrm{mutual}}$ for all the image pairs for misregistration sizes $\delta t = \pm 1, 2, 3$ and 4 mm and degrees. It can be seen that for image pairs (a)(b) and (g)(h) the algorithm works poorly. Consider the joint probability distribution of images intensities in figure 7.7(A) and (B). Distribution (A) shows the intensities of image (a) plotted again the corresponding intensities in image (b). Distribution (B) shows the intensities of image (d) against the corresponding intensities in image (e). Recall that the images are those shown in figure 7.6 where images (a) and (b) were created such that the rendering light source was aligned with the rendering camera, and images (d) and (e) were produced such that the rendering light source was placed in between each camera. Distribution (B) shows that at registration, intensities in image (d) should match intensities in image (e). However, distribution (A) shows that even at registration, whilst there is a relationship, it is certainly more complicated, showing two definite trends.

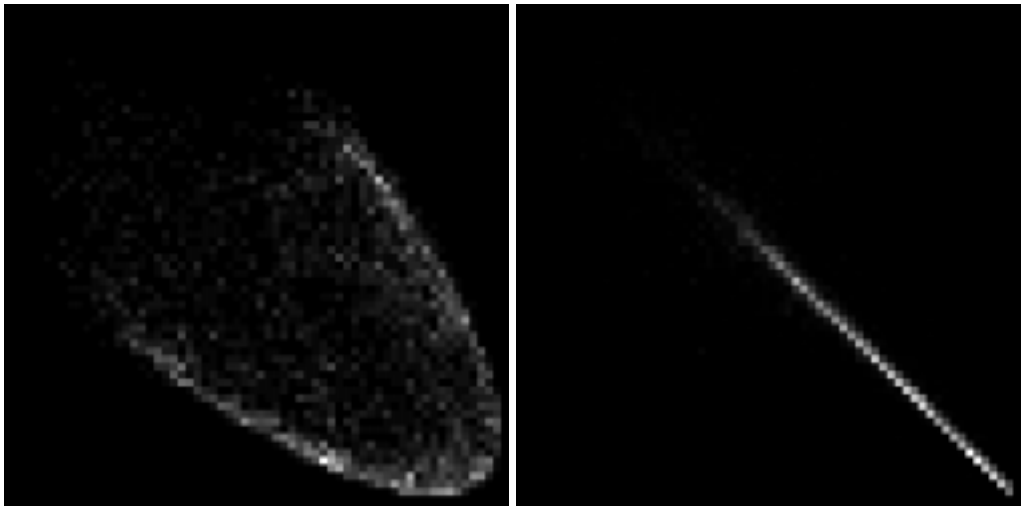
Table 7.4 shows the mean and standard deviation projection and 3D errors for $\mathrm{PC}_{\mathrm{squared}}$ and $\mathrm{PC}_{\mathrm{inverse}}$. The images (d), (e), (j) and (k) were produced with a rendering light source that was fixed between the two cameras. See figure 7.6(f). In all cases, robustness is $\geq 75\%$. It can also be seen that $\mathrm{PC}_{\mathrm{inverse}}$ is more accurate than $\mathrm{PC}_{\mathrm{squared}}$. Recall that for equation (7.8), e is a threshold which can be set to equal the noise level in the video images. Table 7.4 shows results for $\mathrm{PC}_{\mathrm{inverse}}$, with two values for e. When $e^2 = 1$, the success rate is lower, at 75 or 78%. When e is increased, the success rate increases to 92 and 98%. Here the higher value was preferable, as the images used are a mixture of rendered image and video image information which does have noise. Increasing the parameter e effectively smoothes the shape of the cost function in parameter space.

Similarity	Misregistration Size δt	Images	Projection Error (mm) Mean (StdDev)	3D Error (mm) Mean (StdDev)	Percentage Success
$PC'_{squared}$	1	(a)(b)	0.71 (0.29)	0.88 (0.29)	60
$PC'_{squared}$	2	(a)(b)	1.32 (0.58)	1.62 (0.67)	73
$PC'_{squared}$	3	(a)(b)	1.83 (0.73)	2.22 (0.84)	80
$PC'_{squared}$	4	(a)(b)	2.98 (1.04)	3.87 (1.18)	64
$PC'_{inverse}$	1	(a)(b)	0.84 (0.48)	1.13 (0.64)	34
$PC'_{inverse}$	2	(a)(b)	2.21 (1.12)	2.74 (1.37)	14
$PC'_{inverse}$	3	(a)(b)	1.30 (0.29)	2.57 (0.92)	3
$PC'_{inverse}$	4	(a)(b)	7.55 (1.39)	9.46 (1.05)	9
$PC'_{squared}$	1	(g)(h)	0.63 (0.25)	0.69 (0.29)	88
$PC'_{squared}$	2	(g)(h)	0.99 (0.52)	1.10 (0.58)	94
$PC'_{squared}$	3	(g)(h)	1.16 (0.90)	1.32 (1.09)	78
$PC'_{squared}$	4	(g)(h)	2.05 (1.02)	2.31 (1.30)	58
$PC'_{inverse}$	1	(g)(h)	0.98 (0.39)	1.24 (0.54)	23
$PC'_{inverse}$	2	(g)(h)	2.70 (1.18)	3.38 (1.45)	16
$PC'_{inverse}$	3	(g)(h)	5.06 (1.97)	6.14 (1.85)	5
$PC'_{inverse}$	4	(g)(h)	7.37 (1.42)	9.35 (0.62)	3

Table 7.2: Mean (standard deviation) projection and 3D errors for simulations using $PC'_{squared}$ and $PC'_{inverse}$ for each misregistration size $\delta t = \pm 1, 2, 3$ and 4 mm and degrees.

7.5.1.3 Conclusions

These initial tests aim to demonstrate proof of concept. Two different lighting scenarios have been demonstrated, and five different similarity measures. The following conclusions can be made. The first scenario, used a rendering light source that was aligned with each camera. This did not work well for any of $PC'_{squared}$, $PC'_{inverse}$ or PC_{mutual}. The performance of the algorithm suggested that a maximum was present at the correct registration solution, but that the surrounding search space was not smooth. Further research could look at ways of improving this, possibly with more robust search strategies. The second scenario used a single rendering light source fixed relative to the cameras. In this scenario, $PC_{squared}$ and $PC_{inverse}$ performed better. The robustness was 75% to 98%, with 3D errors ranging 0.72 - 1.93mm. PC_{mutual} performed more accurately, precisely and successfully than $PC_{squared}$ and $PC_{inverse}$ for $\delta t = \pm 4$ mm and degrees. However it also performed poorly with images (a)(b) and (g)(h). In principle PC_{mutual}



(A) (B)

Figure 7.7: Joint probability distribution of image intensities used to calculate $\mathrm{PC}_{\mathrm{mutual}}$. Distribution (A) shows the intensities of image (a) plotted against the corresponding intensities in image (b). Distribution (B) shows the intensities of image (d) against the corresponding intensities is image (e). Both distributions (A) and (B) have an origin at the top left, and both images are produced at the 'gold standard' registration. Images, (a),(b),(d), and (e) are from figure 7.6.

should be unaffected by the relationship between different image intensities. However in practice, with a more complicated function, $\mathrm{PC}_{\mathrm{mutual}}$ performed as badly as $\mathrm{PC}'_{\mathrm{squared}}$ and $\mathrm{PC}'_{\mathrm{inverse}}$. Tables 7.2, 7.3, and 7.4 all show that adding the texture mapped dot pattern to the surface of the skull phantom, did not make a significant difference. This suggests that the algorithm may perform similarly well with surfaces exhibiting smoothly varying shading as well as significant texture.

Similarity	Misregistration Size δt	Images	Projection Error (mm) Mean (StdDev)	3D Error (mm) Mean (StdDev)	Percentage Success
PC_{mutual}	1	(a)(b)	1.07 (0.29)	1.21 (0.27)	20
PC_{mutual}	2	(a)(b)	1.90 (0.41)	2.14 (0.50)	19
PC_{mutual}	3	(a)(b)	3.34 (1.30)	3.74 (1.51)	13
PC_{mutual}	4	(a)(b)	9.49 (0.66	10.26 (0)	2
PC_{mutual}	1	(d)(e)	0.34 (0.13)	0.27 (0.14)	100
PC_{mutual}	2	(d)(e)	0.31 (0.36)	0.36 (0.38)	100
PC_{mutual}	3	(d)(e)	0.32 (0.47)	0.37 (0.39)	100
PC_{mutual}	4	(d)(e)	0.42 (0.73)	0.47 (0.77)	94
PC_{mutual}	1	(g)(h)	1.13 (0.33)	1.26 (0.37)	42
PC_{mutual}	2	(g)(h)	2.12 (1.22)	2.43 (1.33)	25
PC_{mutual}	3	(g)(h)	3.08 (1.34)	3.54 (1.80)	11
PC_{mutual}	4	(g)(h)	7.73 (1.60)	9.55 (1.27)	5
PC_{mutual}	1	(i)(j)	0.22 (0.07)	0.24 (0.08)	98
PC_{mutual}	2	(i)(j)	0.23 (0.12)	0.26 (0.13)	97
PC_{mutual}	3	(i)(j)	0.26 (0.24)	0.28 (0.24)	91
PC_{mutual}	4	(i)(j)	0.65 (1.47)	0.70 (1.56)	77

Table 7.3: Mean (standard deviation) projection and 3D error for simulations using PC_{mutual} for each misregistration size $\delta t = \pm 1, 2, 3$ and 4 mm and degrees.

Similarity	Images	Projection Error (mm) Mean (StdDev)	3D Error (mm) Mean (StdDev)	Percentage Success
pre-registration		7.70 (1.18)	9.59 (0.86)	
$PC_{squared}$	(d)(e)	1.60 (0.90)	1.93 (1.11)	92
$PC_{inverse}$ $(e^2 = 1)$	(d)(e)	1.14 (1.37)	1.34 (1.60)	75
$PC_{inverse}$ $(e^2 = 40)$	(d)(e)	0.62 (0.49)	0.72 (0.72)	98
$PC_{squared}$	(i)(j)	1.07 (1.02)	1.24 (1.16)	98
$PC_{inverse}$ $(e^2 = 1)$	(i)(j)	1.04 (1.33)	1.16 (1.47)	78
$PC_{inverse}$ $(e^2 = 40)$	(i)(j)	0.77 (1.37)	0.86 (1.53)	92

Table 7.4: Mean (standard deviation) projection and 3D errors for simulations using $PC_{squared}$ and $PC_{inverse}$ for misregistration size $\delta t = \pm 4$ mm and degrees.

7.5.2 Registration Of A Tricorder Surface Model To Four Video Images

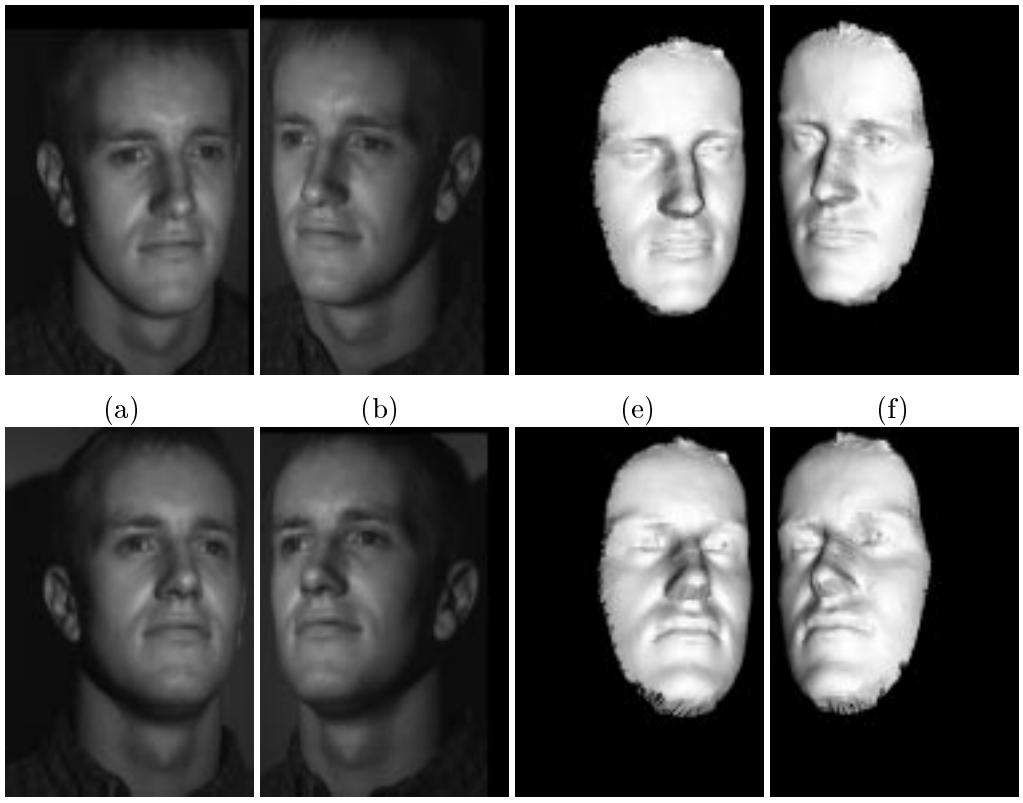
7.5.2.1 Methods

A surface model of a face was acquired using a Tricorder^{TM} S4m system. This system projects a pseudo-random dot pattern onto a subject and captures four video images using four video cameras. The surface is then reconstructed by matching corresponding points in the four views, and triangulating to reconstruct 3D positions. In addition, four video images are captured without patterned light, for the purpose of creating a texture map for the reconstructed surface.

For this dataset, called 'matt', the cameras were calibrated using Tsai's camera calibration method [Tsai, 1987] as described in section 5.4.9.1. The amount of radial distortion present in the video images was small and hence ignored. Thus for the four cameras the intrinsic and extrinsic parameters were known. All four video images are taken using one light source. Thus, assuming the skin exhibits Lambertian reflection, a given point on a surface model should project onto identical image intensities in the video images, apart from noise.

The threshold of e for the measure $\mathrm{PC}_{\mathrm{inverse}}$ was calculated by taking a sequence of two sets of 30 images for each camera. The two sets were of a black and white calibration object, and of the volunteer sitting as still as possible. For each pixel in the the images of the calibration object, and the volunteer, the variance of the pixel intensity values over time was calculated. The mean of the variances for each pixel in the images was calculated. For the calibration object the variance was 2.94, which represents the typical noise level for the video cameras, as the scene did not change. For the volunteer, the variance was 39.8. The threshold e was set so that $e^2 = 40$, as in section 7.5.1.2, the larger value gave better performance.

For each of the similarity measures $\mathrm{PC}_{\mathrm{squared}}$ and $\mathrm{PC}_{\mathrm{inverse}}$, and for each of the 64 possible misregistrations of size $\delta t = \pm 4, 8, 12$ and 16 mm and degrees, the reconstructed surface was registered to the four video images shown in figure 7.8(a)(b)(c) and (d). For each successful registration, the projection error and 3D error was measured using each point in the surface model. The projection error was measured with respect to each video view and the arithmetic mean of the projection error for each of the four views was calculated. The video images are shown in figure 7.8 along with a surface rendering at the gold standard pose for each view.



(a) (b) (e) (f)

(c) (d) (g) (h)

Figure 7.8: The four video images (a),(b),(c),(d) used in section 7.5.2 and a corresponding surface rendering (e),(f),(g),(h) respectively) at the gold standard position. Note this is the same figure and image as figure 5.6 used in section 5.4.9.

7.5.2.2 Results

Tables 7.5 and 7.6 show the mean (standard deviation) projection and 3D errors for each δt for $\text{PC}_{\text{squared}}$ and $\text{PC}_{\text{inverse}}$ respectively. Tables 7.5 and 7.6 show that both $\text{PC}_{\text{squared}}$ and $\text{PC}_{\text{inverse}}$ perform accurately and robustly for $\delta t = \pm 4, 8$ and 12 mm and degrees. The mean 3D error ranges from 1.19 mm and 1.59 mm, with 100 percent success rate. In general, $\text{PC}_{\text{squared}}$ is more accurate, but $\text{PC}_{\text{inverse}}$ more robust.

	Pre-Registration		Post-Registration		
	Projection Error (mm)	3D Error (mm)	Projection Error (mm)	3D Error (mm)	Percentage
δt	Mean (StdDev)	Mean (StdDev)	Mean (StdDev)	Mean (StdDev)	Success
4	6.65 (0.98)	8.75 (0.35)	1.07 (0.19)	1.20 (0.15)	100
8	13.30 (1.96)	17.49 (0.72)	1.06 (0.19)	1.19 (0.15)	100
12	19.94 (2.95)	26.23 (1.11)	1.08 (0.19)	1.20 (0.15)	100
16	26.51 (3.84)	34.86 (1.37)	1.20 (0.71)	1.35 (0.83)	68

Table 7.5: Mean (standard deviation) projection and 3D error for each misregistration size $\delta t = \pm 4, 8, 12$ and 16, using $\mathrm{PC}_{\mathrm{squared}}$.

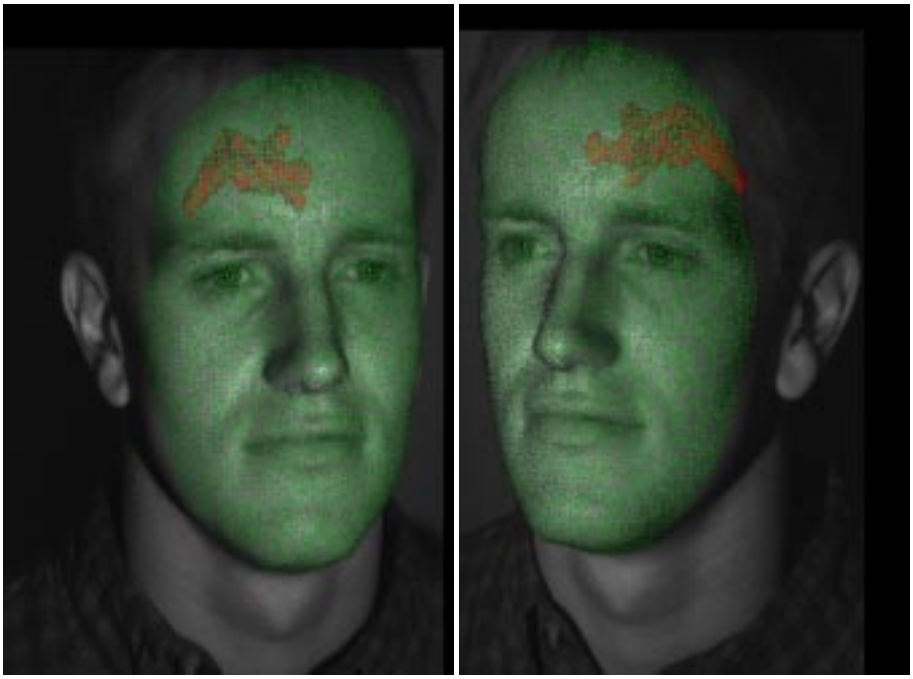
	Pre-Registration		Post-Registration		
	Projection Error (mm)	3D Error (mm)	Projection Error (mm)	3D Error	Percentage
δt	Mean (StdDev)	Mean (StdDev)	Mean (StdDev)	Mean (StdDev)	Success
4	6.65 (0.98)	8.75 (0.35)	1.30 (0.54)	1.50 (0.55)	100
8	13.30 (1.96)	17.49 (0.72)	1.37 (0.56)	1.59 (0.60)	100
12	19.94 (2.95)	26.23 (1.11)	1.23 (0.46)	1.45 (0.46)	100
16	26.51 (3.84)	34.86 (1.37)	1.35 (0.56)	1.57 (0.59)	100

Table 7.6: Mean (standard deviation) projection and 3D error for each misregistration size $\delta t = \pm 4, 8, 12$ and 16, using $\mathrm{PC}_{\mathrm{inverse}}$.

An example of a registration result can be seen in figure 7.9. The Tricorder surface is overlaid as a green wire frame mesh at the registered pose. The skin surface from an MR data set was registered to the Tricorder$^{\mathrm{TM}}$ data using a surface based method [Maurer Jr. *et al.*, 1998]. This enables a visualisation of the ventricles (a) and (b) to be overlaid as a solid red surface.

7.5.2.3 Conclusions

Registering video images to a Tricorder$^{\mathrm{TM}}$ surface demonstrates the effectiveness of this method. Using both $\mathrm{PC}_{\mathrm{squared}}$ and $\mathrm{PC}_{\mathrm{inverse}}$, the algorithm worked effectively for misregistrations of $\delta t = \pm 12$ mm and degrees (see tables 7.5 and 7.6), and even robustly up to $\delta t = \pm 16$ mm and degrees for $\mathrm{PC}_{\mathrm{inverse}}$. This demonstrates recovery of the registration pose from a significant offset. The mean 3D error using this surface ranges from 1.19 to 1.59 mm for $\delta t = \pm 4, 8, 12$ mm and degrees. Visually this corresponds to an accurate registration.



(a) (b)

Figure 7.9: Overlays of the surface model, and a segmentation of the ventricles. The surface used for registration is shown in wire frame green and the segmented surface in solid red.

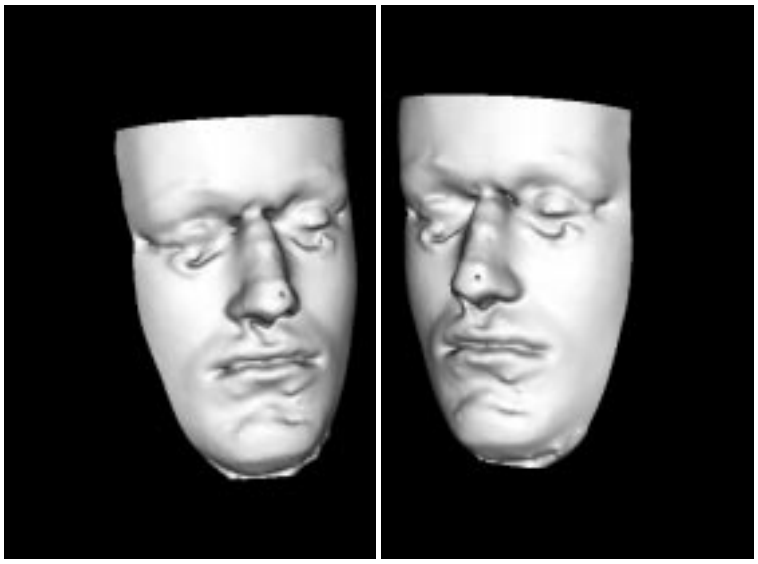
7.5.3 Registration Of An MR Scan To Four Video Images

7.5.3.1 Methods

The Tricorder^{TM} S4m surface model used thus far in this chapter represents a best case scenario. The experiment is realistic in that a 3D surface is being registered to video images, but the surface was originally derived from similar video images by using an independent method, and thus should fit optimally. For image guided surgery it is more likely that a surface derived from an MR scan will need to be registered to several video images. The MR scan might be performed days or weeks before the video images are captured. An MR scan (Gradient Echo 256 $\times$ 256 $\times$ 132, 1.0 $\times$ 1.0 $\times$ 1.3 mm voxels) was taken and a skin surface of the face was extracted using ANALYZE (Biomedical Imaging Resource, Mayo Foundation, Rochester, MN, USA.), and an iso-surface model created using the marching cubes, and smoothing algorithms within VTK [Schroeder *et al.*, 1997]. The iso-surface threshold was the mean average of a typical skin and air intensity value. Figure 7.10 shows the extracted surface. The MR scan was taken three months before the video images. Artifacts around the eyes make segmentation difficult and can be seen in figure 7.10. A single registration was performed using $\text{PC}_{\text{squared}}$ giving a solution of -2.35, -1.46, 0.89, 2.71, -0.10, -4.25 for the six parameters $t_x \ldots r_z$ respectively. This registration appeared visually accurate. No gold standard for this data exists, however the algorithm was tested by assuming that this initial solution was at least 'near' the correct solution. Misregistrations of size $\delta t = \pm 4, 8, 12$ and 16 mm and degrees were made from this initial solution. The algorithm then registered the MR surface to the four video images shown in figure 7.8 using $\text{PC}_{\text{squared}}$ and $\text{PC}_{\text{inverse}}$. Successful registrations were counted as those where none of the extrinsic parameters finished further away from the above initial solution than the initial misregistration size δt.

7.5.3.2 Results

Tables 7.7 and 7.8 show the standard deviation of the recovered extrinsic parameters for $\text{PC}_{\text{squared}}$ and $\text{PC}_{\text{inverse}}$ respectively. No gold standard exists, however, the registrations appeared visually acceptable. Tables 7.7 and 7.8 show that both $\text{PC}_{\text{squared}}$ and $\text{PC}_{\text{inverse}}$ work precisely, i.e. a small standard deviation for each recovered parameter $t_x \ldots r_z$. $\text{PC}_{\text{inverse}}$ is more robust than $\text{PC}_{\text{squared}}$, with a success rate of 100 and 89% for $\delta t = \pm 12$ and 16 mm and degrees, compared with 80 and 42% for $\text{PC}_{\text{squared}}$. $\text{PC}_{\text{inverse}}$ precisely recovers the extrinsic parameters, even when $\delta t = \pm 16$ mm and degrees.



(a) (b)

Figure 7.10: (a) and (b) Two different views of the MR surface used for test data in section 7.5.3.

	Registration Parameter						Percentage
δt	t_x	t_y	t_z	r_x	r_y	r_z	Success
4	0.23	0.12	0.21	0.25	0.38	0.68	100
8	0.21	0.12	0.20	0.24	0.41	0.70	100
12	0.17	0.13	0.19	0.26	0.39	0.66	80
16	3.31	2.90	1.27	0.66	2.82	3.05	42

Table 7.7: Standard deviation of recovered extrinsic parameters for each misregistration size δt and using $\mathrm{PC}_{\mathrm{squared}}$.

7.5.3.3 Conclusions

Tables 7.7 and 7.8 both show that for the MR surface, registration is both precise and robust for misregistrations of sizes $\delta t = \pm 4$ and 8 mm and degrees. $\mathrm{PC}_{\mathrm{inverse}}$ performs robustly and accurately for misregistrations of sizes $\delta t = \pm 12$ and 16 mm and degrees. For the Tricorder$^{\mathrm{TM}}$ and MR surface, $\mathrm{PC}_{\mathrm{inverse}}$ has performed more robustly than $\mathrm{PC}_{\mathrm{squared}}$ and sufficiently accurately. In addition, using the method described in section 7.5.2.1, a value for e^2 (used in equation (7.8) to calculate $\mathrm{PC}_{\mathrm{inverse}}$) can be calculated from the variance of pixel intensities over time, thereby allowing $\mathrm{PC}_{\mathrm{inverse}}$ to be adjustable to tolerate more or less noise.

	Registration Parameter						Percentage
δt	t_x	t_y	t_z	r_x	r_y	r_z	Success
4	0.69	0.25	0.15	0.14	0.78	0.26	100
8	0.66	0.25	0.12	0.12	0.78	0.24	100
12	0.74	0.28	0.14	0.12	0.81	0.27	100
16	0.78	0.30	0.16	0.13	0.92	0.25	89

Table 7.8: Standard deviation of recovered extrinsic parameters for each misregistration size δt and using $\mathrm{PC}_{\mathrm{inverse}}$.

7.5.4 Registration Of An MR Scan Or Tricorder$^{\mathrm{TM}}$ Surface Model To Two Video Images Using $\mathrm{PC}_{\mathrm{inverse}}$

7.5.4.1 Methods

The video images shown in figure 7.8(a),(b),(c) and (d) were paired into (a)(b), (a)(c), (b)(d), (c)(d), (a)(d) and (b)(c). For each pair of images, and for each misregistration size of $\delta t = \pm 4, 8, 12$ and 16 mm and degrees, the algorithm, using $\mathrm{PC}_{\mathrm{inverse}}$ as a similarity measure was used to register the Tricorder$^{\mathrm{TM}}$ surface model from section 7.5.2 to the video images. Each registration was classified as a success or failure as before, and the mean and standard deviation projection and 3D errors were calculated for successful registrations. The same video images were also registered to the MR surface from the previous section 7.5.3. For each successful registration, the standard deviation of the recovered extrinsic parameter values was calculated.

7.5.4.2 Results

Referring to figure 7.8, the pairings of images (a), (b), (c) and (d) can be split into three groups, horizontal image pairs (a)(b) and (c)(d), vertical pairs (a)(c) and (d)(b) and diagonal pairs (a)(d), (b)(c). The results for each pair of images for misregistration size $\delta t = \pm 8$ mm and degrees are shown in table 7.9 for the Tricorder$^{\mathrm{TM}}$ surface and in table 7.10 for the MR surface. These two tables are now used to summarise the performance using two views.

It can be seen that for both Tricorder$^{\mathrm{TM}}$ and MR surfaces, the horizontal and diagonal pairings perform more robustly than the vertical pairings. For horizontal and diagonal pairings, the accuracy and precision is similar to that with four views. The mean 3D errors range from 1.49 to 1.59 mm. For vertical pairings however, the mean (standard

Image Pair	Image Grouping	Projection Error (mm) Mean (StdDev)	3D Error (mm) Mean (StdDev)	Percentage Success
pre-registration		13.30 (1.96)	17.49 (0.72)	
(a)(b)	horizontal	1.19 (1.00)	1.59 (1.13)	100
(c)(d)	horizontal	1.20 (0.45)	1.53 (0.38)	100
(a)(c)	vertical	2.97 (1.56)	3.95 (1.86)	81
(b)(d)	vertical	4.52 (1.78)	5.50 (2.12)	98
(a)(d)	diagonal	1.34 (0.60)	1.56 (0.60)	100
(b)(c)	diagonal	1.25 (0.48)	1.49 (0.54)	98

Table 7.9: Mean (standard deviation) projection and 3D errors for registrations of a Tricorder^{TM} surface to pairs of images from figure 7.8 using $PC_{inverse}$, for $\delta t = \pm 8$ mm and degrees.

Image Pair	Image Grouping	Registration Parameter						Percentage Success
		t_x	t_y	t_z	r_x	r_y	r_z	
(a)(b)	horizontal	0.33	0.20	0.20	0.26	0.42	0.32	100
(c)(d)	horizontal	0.72	1.06	0.67	0.44	0.64	1.28	100
(a)(c)	vertical	1.03	0.25	0.45	0.40	0.83	0.65	36
(b)(d)	vertical	0.68	1.72	0.38	0.58	0.87	1.29	11
(a)(d)	diagonal	0.32	0.29	0.20	0.19	0.40	0.30	100
(b)(c)	diagonal	0.44	0.58	0.21	0.44	0.63	0.60	100

Table 7.10: Standard deviation of recovered extrinsic parameter values for registration of an MR surface to pairs of images from figure 7.8, using $PC_{inverse}$ for $\delta t = \pm 8$ mm and degrees.

deviation) 3D errors for a Tricorder^{TM} are 3.95 (1.86) and 5.50 (2.12) mm, which indicates poor accuracy and precision. For the MR surface, and vertical pairs of images, the success rate is also low, but the standard deviation of the parameters is similar to the horizontal and diagonal pairings. However, inspection of each registration results reveals that the vertical pairings that were classified as successful (i.e. only 11 or 36% in table 7.10) converge to a false maximum of the cost function. This false maximum was only a small offset from the visually correct solution. The recovered extrinsic parameters of the failed registrations were a long way from the visually correct solution.

Tables 7.11 to 7.14 show the mean and standard deviation, projection and 3D errors in mm, for registration of a Tricorder^{TM} surface to different combinations of video images from figure 7.8 for each misregistration size $\delta t = \pm 4, 8, 12$ and 16 mm and degrees.

It can be seen that the trends observed in table 7.9 for misregistration sizes of $\delta t = \pm 8$ mm and degrees, still hold for misregistration sizes $\delta t = \pm 4, 12$ and 16 mm and degrees. Tables 7.11 and 7.12 show that for horizontal pairings (a)(b) and (c)(d), performance is still good, even up to misregistration sizes of $\delta t = \pm 16$ mm and degrees. The success rate is 100% throughout, and the mean 3D error over these two tables is 1.54mm.

Tables 7.13 and 7.14 show that the success rate, and accuracy decreases slightly for diagonal pairings, and tables 7.15 and 7.16 show that the success rate and accuracy is the worst for vertical pairings. This degradation of performance, when compared with the horizontal pairings is independent of misregistration size δt.

The tables 7.17 to 7.20 show the standard deviations of the post-registration extrinsic parameter values $t_x \ldots r_z$ for registration of an MR surface to different combinations of video images from figure 7.8 for each misregistration size $\delta t = \pm 4, 8, 12$ and 16 mm and degrees.

Tables 7.17, 7.18, 7.19 and 7.20 show that horizontal pairings give the most accurate and successful registrations, with diagonal pairings slightly worse. For misregistration sizes of $\delta t = 4$ and 8 mm and degrees, the success rate is 100% and the standard deviations of the recovered extrinsic parameters ranges from 0.18 to 1.28, which is low. For vertical pairings, tables 7.21 and 7.22 show very poor performance. The fact that the performance is poor for each misregistration size of $\delta t = \pm 4, 8, 12$ and 16 shows that the similarity measure is failing completely.

	Pre-Registration		Post-Registration		
	Projection Error (mm)	3D Error (mm)	Projection Error (mm)	3D Error (mm)	Percentage
δt	Mean (StdDev)	Mean (StdDev)	Mean (StdDev)	Mean (StdDev)	Success
4	6.65 (0.98)	8.75 (0.35)	1.12 (0.54)	1.49 (0.55)	100
8	13.30 (1.96)	17.49 (0.72)	1.19 (1.00)	1.59 (1.13)	100
12	19.94 (2.95)	26.23 (1.11)	1.21 (1.09)	1.60 (1.20)	100
16	26.51 (3.84)	34.86 (1.37)	0.92 (0.40)	1.29 (0.36)	100

Table 7.11: Mean (standard deviation) projection and 3D errors for registrations of a Tricorder^{TM} surface to images (a) and (b) from figure 7.8 using $PC_{inverse}$, for $\delta t = \pm 4, 8, 12$ and 16 mm and degrees.

	Pre-Registration		Post-Registration		
	Projection Error (mm)	3D Error (mm)	Projection Error (mm)	3D Error (mm)	Percentage
δt	Mean (StdDev)	Mean (StdDev)	Mean (StdDev)	Mean (StdDev)	Success
4	6.65 (0.98)	8.75 (0.35)	1.26 (0.52)	1.60 (0.48)	100
8	13.30 (1.96)	17.49 (0.72)	1.20 (0.45)	1.53 (0.38)	100
12	19.94 (2.95)	26.23 (1.11)	1.24 (0.49)	1.55 (0.42)	100
16	26.51 (3.84)	34.86 (1.37)	1.32 (0.47)	1.64 (0.41)	100

Table 7.12: Mean (standard deviation) projection and 3D errors for registrations of a Tricorder^{TM} surface to images (c) and (d) from figure 7.8 using $PC_{inverse}$, for $\delta t = \pm 4, 8, 12$ and 16 mm and degrees.

	Pre-Registration		Post-Registration		
	Projection Error (mm)	3D Error (mm)	Projection Error (mm)	3D Error (mm)	Percentage
δt	Mean (StdDev)	Mean (StdDev)	Mean (StdDev)	Mean (StdDev)	Success
4	6.65 (0.98)	8.75 (0.35)	1.24 (0.47)	1.45 (0.45)	100
8	13.30 (1.96)	17.49 (0.72)	1.34 (0.60)	1.56 (0.60)	98
12	19.94 (2.95)	26.23 (1.11)	1.46 (1.11)	1.69 (1.29)	100
16	26.51 (3.84)	34.86 (1.37)	1.98 (3.59)	2.29 (4.16)	100

Table 7.13: Mean (standard deviation) projection and 3D errors for registrations of a Tricorder^{TM} surface to images (a) and (d) from figure 7.8 using $PC_{inverse}$, for $\delta t = \pm 4, 8, 12$ and 16 mm and degrees.

	Pre-Registration		Post-Registration		
	Projection Error (mm)	3D Error (mm)	Projection Error (mm)	3D Error (mm)	Percentage
δt	Mean (StdDev)	Mean (StdDev)	Mean (StdDev)	Mean (StdDev)	Success
4	6.65 (0.98)	8.75 (0.35)	1.32 (0.63)	1.55 (0.70)	100
8	13.30 (1.96)	17.49 (0.72)	1.25 (0.48)	1.49 (0.54)	98
12	19.94 (2.95)	26.23 (1.11)	1.45 (1.30)	1.71 (1.43)	98
16	26.51 (3.84)	34.86 (1.37)	2.24 (3.38)	2.59 (3.80)	100

Table 7.14: Mean (standard deviation) projection and 3D errors for registrations of a Tricorder^{TM} surface to images (b) and (c) from figure 7.8 using $PC_{inverse}$, for $\delta t = \pm 4, 8, 12$ and 16 mm and degrees.

	Pre-Registration		Post-Registration		
	Projection Error (mm)	3D Error (mm)	Projection Error (mm)	3D Error (mm)	Percentage
δt	Mean (StdDev)	Mean (StdDev)	Mean (StdDev)	Mean (StdDev)	Success
4	6.65 (0.98)	8.75 (0.35)	1.74 (0.58)	2.36 (0.67)	85
8	13.30 (1.96)	17.49 (0.72)	2.97 (1.56)	3.95 (1.86)	81
12	19.94 (2.95)	26.23 (1.11)	5.09 (3.15)	6.52 (3.74)	83
16	26.51 (3.84)	34.86 (1.37)	7.87 (4.21)	9.85 (4.96)	80

Table 7.15: Mean (standard deviation) projection and 3D errors for registrations of a Tricorder^{TM} surface to images (a) and (c) from figure 7.8 using $PC_{inverse}$, for $\delta t = \pm 4, 8, 12$ and 16 mm and degrees.

	Pre-Registration		Post-Registration		
	Projection Error (mm)	3D Error (mm)	Projection Error (mm)	3D Error (mm)	Percentage
δt	Mean (StdDev)	Mean (StdDev)	Mean (StdDev)	Mean (StdDev)	Success
4	6.65 (0.98)	8.75 (0.35)	2.74 (0.64)	3.31 (0.74)	88
8	13.30 (1.96)	17.49 (0.72)	4.52 (1.78)	5.50 (2.12)	98
12	19.94 (2.95)	26.23 (1.11)	6.05 (2.57)	7.42 (3.08)	88
16	26.51 (3.84)	34.86 (1.37)	8.02 (3.89)	9.78 (4.63)	75

Table 7.16: Mean (standard deviation) projection and 3D errors for registrations of a Tricorder^{TM} surface to images (b) and (d) from figure 7.8 using $PC_{inverse}$, for $\delta t = \pm 4, 8, 12$ and 16 mm and degrees.

	Registration Parameter						Percentage
δt	t_x	t_y	t_z	r_x	r_y	r_z	Success
4	0.35	0.20	0.18	0.24	0.43	0.32	100
8	0.33	0.20	0.20	0.26	0.42	0.32	100
12	0.75	2.02	0.20	0.49	0.65	0.86	91
16	1.73	4.68	0.28	1.04	0.98	1.94	88

Table 7.17: Standard deviation of recovered extrinsic parameter values for registration of an MR surface to images (a) and (b) from figure 7.8, using $\mathrm{PC}_{\mathrm{inverse}}$ for $\delta t = \pm 4, 8, 12$ and 16 mm and degrees.

	Registration Parameter						Percentage
δt	t_x	t_y	t_z	r_x	r_y	r_z	Success
4	0.34	0.26	0.18	0.30	0.34	0.58	100
8	0.72	1.06	0.67	0.44	0.64	1.28	100
12	0.36	0.20	0.20	0.31	0.40	0.59	100
16	0.38	0.23	0.21	0.37	0.41	0.69	97

Table 7.18: Standard deviation of recovered extrinsic parameter values for registration of an MR surface to images (c) and (d) from figure 7.8, using $\mathrm{PC}_{\mathrm{inverse}}$ for $\delta t = \pm 4, 8, 12$ and 16 mm and degrees.

	Registration Parameter						Percentage
δt	t_x	t_y	t_z	r_x	r_y	r_z	Success
4	0.30	0.29	0.22	0.23	0.34	0.32	100
8	0.32	0.29	0.20	0.19	0.40	0.30	100
12	0.29	0.44	0.19	0.22	0.40	0.35	98
16	1.25	1.62	0.36	0.20	1.15	0.94	81

Table 7.19:]

Standard deviation of recovered extrinsic parameter values for registration of an MR surface to images (a) and (d) from figure 7.8, using $\mathrm{PC}_{\mathrm{inverse}}$ for $\delta t = \pm 4, 8, 12$ and 16 mm and degrees.

	Registration Parameter						Percentage
δt	t_x	t_y	t_z	r_x	r_y	r_z	Success
4	0.38	0.53	0.20	0.44	0.54	0.58	100
8	0.44	0.58	0.21	0.44	0.63	0.60	100
12	0.94	0.95	0.81	0.83	0.95	1.49	98
16	1.07	2.78	0.47	0.45	0.73	1.70	87

Table 7.20: Standard deviation of recovered extrinsic parameter values for registration of an MR surface to images (b) and (c) from figure 7.8, using $\mathrm{PC}_{\mathrm{inverse}}$ for $\delta t = \pm 4, 8, 12$ and 16 mm and degrees.

	Registration Parameter						Percentage
δt	t_x	t_y	t_z	r_x	r_y	r_z	Success
4	0.21	0.21	0.23	0.22	0.22	0.24	19
8	1.03	0.25	0.45	0.40	0.83	0.65	36
12	1.00	3.87	1.49	0.45	1.04	1.18	22
16	2.45	3.42	1.87	0.43	2.16	0.78	15

Table 7.21: Standard deviation of recovered extrinsic parameter values for registration of an MR surface to images (a) and (c) from figure 7.8, using $\mathrm{PC}_{\mathrm{inverse}}$ for $\delta t = \pm 4, 8, 12$ and 16 mm and degrees.

	Registration Parameter						Percentage
δt	t_x	t_y	t_z	r_x	r_y	r_z	Success
4	-	-	-	-	-	-	0
8	0.68	1.72	0.38	0.58	0.87	1.29	11
12	1.26	2.57	0.26	0.95	0.84	2.18	11
16	1.18	8.34	2.59	0.79	1.44	1.05	9

Table 7.22: Standard deviation of recovered extrinsic parameter values for registration of an MR surface to images (b) and (d) from figure 7.8, using $\mathrm{PC}_{\mathrm{inverse}}$ for $\delta t = \pm 4, 8, 12$ and 16 mm and degrees.

7.5.4.3 Conclusions

The experiments with different pairs of images illustrate two points. First, registration performance does vary with the orientation and shape of the objects relative to the position of the video cameras. However, with a good choice of camera position, performance with two views is very good and comparable with the four view experiments. Even with misregistration sizes of $\delta t = \pm 12$ and 16 mm and degrees, the algorithm was 100 percent successful for the horizontal pairings. Diagonal pairings performed slightly worse, especially for $\delta t = \pm 16$ mm and degrees. The vertical pairings performed poorly for all misregistration sizes. The fact that performance does vary with camera setup relative to the object of interest is not surprising. If two cameras were positioned along the major axis of a perfectly Lambertian reflecting infinitely long cylinder, it would be impossible using this method to determine the rotation about the axis, and translations parallel to the axis, as with these transformations, the cylinder would produce identical image intensities. If the face is assumed to be approximately symmetrical about a vertical axis, then both horizontal and vertical pairings could be affected.

A more likely explanation is that for the Tricorder system, the horizontal cameras have a rotational disparity of 32 degrees, the diagonal pairings have a rotational disparity of 38 degrees, but the vertical pairings have a disparity of only 19 degrees. As the angular disparity between the cameras decreases, so the effective signal to noise ratio decreases. The photo-consistency measures try to detect misregistration by measuring differences in intensity. As these differences decrease, they will be more affected by the inherent video image noise.

Furthermore, it may well be the case that of the facial features, the nose and the curvature from one side of the face to the other is the most important in terms of registration. The horizontal and diagonal pairs of images capture views from both sides of the face, whereas the vertical pairs only view one side of the face. It may be the case that for the vertical pairings, there is simply not enough angular disparity between the video cameras, or not enough surface curvature to be enable accurate registration.

To quantify the lower bounds on the angular disparity of the video cameras, and answer questions such as how much surface curvature is necessary within the field of view to enable accurate registration would require a large amount of further research, and is left for future work.

7.5.5 Registration Of An MR Scan Or Tricorder^{TM} Surface Model To Two Video Images Using PC_{mutual}

7.5.5.1 Methods

In section 7.4.4, the similarity measure PC_{mutual} was described. In this section PC_{mutual} is tested with respect to its performance with two video views. The video images shown in figure 7.8(a)(b)(c) and (d) were taken, along with the Tricorder^{TM} surface from section 7.5.2. As before the gold standard registration was known. For pairs of images (a)(b), (c)(d), (a)(c), (b)(d), (a)(d) and (b)(c), and for misregistration sizes of $\delta t = \pm 4, 8, 12$ and 16 mm and degrees, the Tricorder ^{TM} surface was registered to the video images. Successful registrations were classified as before (section 7.5.2) and the mean and standard deviation projection and 3D error was calculated for each group of images. For pairs of images, PC_{mutual} evaluates a 2D joint probability distribution where image intensities are binned into a 64×64 histogram. As in section 7.5.4, this was also repeated using the MR scan from section 7.5.3. For successful registrations, the standard deviation of the recovered extrinsic parameter values was calculated.

7.5.5.2 Results

Table 7.23 shows the mean and standard deviation projection and 3D errors when registering a Tricorder^{TM} surface to each pair of video images, where the initial misregistration size was $\delta t = \pm 8$ mm and degrees. Similarly, table 7.24 shows the standard deviation of the recovered extrinsic camera parameters when registering the MR surface to each pair of video images. These two tables show a summary of the results for this experiment.

Comparing the performance of PC_{mutual}, in tables 7.23 and 7.24, with $PC_{inverse}$, in table 7.9 and 7.10, it can be seen that for horizontal and diagonal pairs of images, the performance of PC_{mutual} is slightly less successful, less accurate and less precise than $PC_{inverse}$. For vertical pairings however, PC_{mutual} does not suffer as badly in terms of accuracy and robustness, when $PC_{inverse}$ does.

Tables 7.25 to 7.30 show the mean and standard deviation projection and 3D errors when registering a Tricorder^{TM} surface to each pair of video images using PC_{mutual}, for each misregistration size of $\delta t = \pm 4, 8, 12$ and 16 mm and degrees. The tables 7.31 to 7.34 show the standard deviation of the post-registration extrinsic parameter values $t_x \ldots r_z$ for registration of an MR surface to different pairs of video images from figure 7.8, using

Image Pair	Image Grouping	Projection Error (mm) Mean (StdDev)	3D Error (mm) Mean (StdDev)	Percentage Success
pre-registration		13.30 (1.96)	17.49 (0.72)	
(a)(b)	horizontal	2.10 (1.82)	2.59 (2.04)	92
(c)(d)	horizontal	1.59 (1.22)	2.11 (1.27)	95
(a)(c)	vertical	1.34 (0.49)	1.73 (0.52)	100
(b)(d)	vertical	2.30 (1.42)	2.69 (1.61)	95
(a)(d)	diagonal	2.00 (1.44)	2.34 (1.68)	91
(b)(c)	diagonal	1.79 (1.06)	2.04 (1.23)	87

Table 7.23: Mean (standard deviation) projection and 3D errors for registrations of a Tricorder$^{\text{TM}}$ surface to pairs of images from figure 7.8, using $\text{PC}_{\text{mutual}}$, for $\delta t = \pm 8$ mm and degrees

Image Pair	Image Grouping	Registration Parameter						Percentage Success
		t_x	t_y	t_z	r_x	r_y	r_z	
(a)(b)	horizontal	0.37	0.17	0.13	0.16	0.50	0.29	100
(c)(d)	horizontal	0.73	0.81	0.69	0.51	0.68	1.03	100
(a)(c)	vertical	0.28	0.15	0.16	0.13	0.45	0.30	91
(b)(d)	vertical	0.39	0.17	0.16	0.19	0.57	0.30	83
(a)(d)	diagonal	0.39	0.29	0.14	0.18	0.48	0.44	89
(b)(c)	diagonal	0.49	0.41	0.22	0.32	0.64	0.47	92

Table 7.24: Standard deviation of recovered extrinsic parameter values for registration of an MR surface to pairs of images from figure 7.8, using $\text{PC}_{\text{mutual}}$ for $\delta t = \pm 8$ mm and degrees.

$\text{PC}_{\text{mutual}}$ and for each misregistration size of $\delta t = \pm 4, 8, 12$ and 16 mm and degrees. It can be seen that the similarity measure $\text{PC}_{\text{mutual}}$ has a smaller capture range. For the 'horizontal' pairings, (a)(b) and (c)(d), the success rate drops from 100% when the initial misregistration size $\delta t > 8$ mm and degrees, whereas for $\text{PC}_{\text{inverse}}$ this is not the case.

However, the performance for the vertical pairings (a)(c) and (b)(d) is not significantly different from the horizontal or diagonal pairs. Recall that, using $\text{PC}_{\text{inverse}}$, the vertical pairs registered poorly. This implies that $\text{PC}_{\text{mutual}}$ is working for vertical pairings where, $\text{PC}_{\text{inverse}}$ failed completely.

	Pre-Registration		Post-Registration		
	Projection Error (mm)	3D Error (mm)	Projection Error (mm)	3D Error (mm)	Percentage
δt	Mean (StdDev)	Mean (StdDev)	Mean (StdDev)	Mean (StdDev)	Success
4	6.65 (0.98)	8.75 (0.35)	1.28 (0.71)	1.62 (0.75)	100
8	13.30 (1.96)	17.49 (0.72)	1.19 (1.00)	1.59 (1.13)	100
12	19.94 (2.95)	26.23 (1.11)	4.17 (4.16)	4.90 (4.62)	48
16	26.51 (3.84)	34.86 (1.37)	10.24 (7.08)	11.91 (7.91)	9

Table 7.25: Mean (standard deviation) projection and 3D errors for registrations of a Tricorder$^{\text{TM}}$ surface to images (a) and (b) from figure 7.8, using $\text{PC}_{\text{mutual}}$, for $\delta t = \pm 4, 8, 12$ and 16 mm and degrees.

	Pre-Registration		Post-Registration		
	Projection Error (mm)	3D Error (mm)	Projection Error (mm)	3D Error (mm)	Percentage
δt	Mean (StdDev)	Mean (StdDev)	Mean (StdDev)	Mean (StdDev)	Success
4	6.65 (0.98)	8.75 (0.35)	1.11 (0.63)	1.59 (0.54)	100
8	13.30 (1.96)	17.49 (0.72)	1.59 (1.22)	2.11 (1.27)	100
12	19.94 (2.95)	26.23 (1.11)	3.26 (3.85)	3.91 (4.17)	67
16	26.51 (3.84)	34.86 (1.37)	8.12 (6.18)	9.58 (7.16)	18

Table 7.26: Mean (standard deviation) projection and 3D errors for registrations of a Tricorder$^{\text{TM}}$ surface to images (c) and (d) from figure 7.8, using $\text{PC}_{\text{mutual}}$, for $\delta t = \pm 4, 8, 12$ and 16 mm and degrees.

	Pre-Registration		Post-Registration		
	Projection Error (mm)	3D Error (mm)	Projection Error (mm)	3D Error (mm)	Percentage
δt	Mean (StdDev)	Mean (StdDev)	Mean (StdDev)	Mean (StdDev)	Success
4	6.65 (0.98)	8.75 (0.35)	1.36 (0.68)	1.58 (0.75)	100
8	13.30 (1.96)	17.49 (0.72)	2.00 (1.44)	2.34 (1.68)	91
12	19.94 (2.95)	26.23 (1.11)	3.88 (3.87)	4.47 (4.37)	48
16	26.51 (3.84)	34.86 (1.37)	5.19 (6.79)	5.68 (7.37)	6

Table 7.27: Mean (standard deviation) projection and 3D errors for registrations of a Tricorder$^{\text{TM}}$ surface to images (a) and (d) from figure 7.8, using $\text{PC}_{\text{mutual}}$, for $\delta t = \pm 4, 8, 12$ and 16 mm and degrees.

	Pre-Registration		Post-Registration		
	Projection Error (mm)	3D Error (mm)	Projection Error (mm)	3D Error (mm)	Percentage
δt	Mean (StdDev)	Mean (StdDev)	Mean (StdDev)	Mean (StdDev)	Success
4	6.65 (0.98)	8.75 (0.35)	1.24 (0.59)	1.39 (0.64)	100
8	13.30 (1.96)	17.49 (0.72)	1.79 (1.06)	2.04 (1.23)	87
12	19.94 (2.95)	26.23 (1.11)	3.93 (4.34)	4.44 (4.87)	41
16	26.51 (3.84)	34.86 (1.37)	12.91 (7.83)	14.78 (8.86)	13

Table 7.28: Mean (standard deviation) projection and 3D errors for registrations of a Tricorder^{TM} surface to images (b) and (c) from figure 7.8, using PC_{mutual}, for $\delta t = \pm 4, 8, 12$ and 16 mm and degrees.

	Pre-Registration		Post-Registration		
	Projection Error (mm)	3D Error (mm)	Projection Error (mm)	3D Error (mm)	Percentage
δt	Mean (StdDev)	Mean (StdDev)	Mean (StdDev)	Mean (StdDev)	Success
4	6.65 (0.98)	8.75 (0.35)	1.27 (0.38)	1.63 (0.40)	100
8	13.30 (1.96)	17.49 (0.72)	1.34 (0.49)	1.73 (0.52)	100
12	19.94 (2.95)	26.23 (1.11)	2.52 (2.90)	3.06 (3.23)	86
16	26.51 (3.84)	34.86 (1.37)	3.86 (5.54)	4.51 (6.15)	38

Table 7.29: Mean (standard deviation) projection and 3D errors for registrations of a Tricorder^{TM} surface to images (a) and (c) from figure 7.8, using PC_{mutual}, for $\delta t = \pm 4, 8, 12$ and 16 mm and degrees.

	Pre-Registration		Post-Registration		
	Projection Error (mm)	3D Error (mm)	Projection Error (mm)	3D Error (mm)	Percentage
δt	Mean (StdDev)	Mean (StdDev)	Mean (StdDev)	Mean (StdDev)	Success
4	6.65 (0.98)	8.75 (0.35)	1.73 (0.67)	2.06 (0.70)	100
8	13.30 (1.96)	17.49 (0.72)	2.30 (1.42)	2.69 (1.61)	95
12	19.94 (2.95)	26.23 (1.11)	3.23 (3.30)	3.77 (3.75)	66
16	26.51 (3.84)	34.86 (1.37)	4.90 (4.71)	5.74 (5.46)	39

Table 7.30: Mean (standard deviation) projection and 3D errors for registrations of a Tricorder^{TM} surface to images (b) and (d) from figure 7.8, using PC_{mutual}, for $\delta t = \pm 4, 8, 12$ and 16 mm and degrees.

	Registration Parameter						Percentage
δt	t_x	t_y	t_z	r_x	r_y	r_z	Success
4	0.42	0.15	0.13	0.15	0.58	0.36	100
8	0.37	0.17	0.13	0.16	0.50	0.29	100
12	0.96	1.71	0.25	0.66	1.25	1.89	50
16	-	-	-	-	-	-	5

Table 7.31: Standard deviation of recovered extrinsic parameter values for registration of an MR surface to images (a) and (b) from figure 7.8, using $\mathrm{PC}_{\mathrm{mutual}}$ for $\delta t = \pm 4, 8, 12$ and 16 mm and degrees.

	Registration Parameter						Percentage
δt	t_x	t_y	t_z	r_x	r_y	r_z	Success
4	0.47	0.12	0.14	0.13	0.44	0.27	100
8	0.73	0.81	0.69	0.51	0.68	1.03	100
12	2.10	2.78	1.40	1.00	1.17	2.67	77
16	1.94	6.61	3.37	0.83	4.77	3.68	16

Table 7.32: Standard deviation of recovered extrinsic parameter values for registration of an MR surface to images (c) and (d) from figure 7.8, using $\mathrm{PC}_{\mathrm{mutual}}$ for $\delta t = \pm 4, 8, 12$ and 16 mm and degrees.

	Registration Parameter						Percentage
δt	t_x	t_y	t_z	r_x	r_y	r_z	Success
4	0.30	0.20	0.13	0.15	0.42	0.40	100
8	0.39	0.29	0.14	0.18	0.48	0.44	89
12	0.55	0.19	0.12	0.13	0.63	0.52	16
16	-	-	-	-	-	-	3

Table 7.33: Standard deviation of recovered extrinsic parameter values for registration of an MR surface to images (a) and (d) from figure 7.8, using $\mathrm{PC}_{\mathrm{mutual}}$ for $\delta t = \pm 4, 8, 12$ and 16 mm and degrees.

	Registration Parameter						Percentage
δt	t_x	t_y	t_z	r_x	r_y	r_z	Success
4	0.45	0.42	0.21	0.35	0.59	0.49	100
8	0.49	0.41	0.22	0.32	0.64	0.47	92
12	3.19	3.97	2.14	2.39	3.12	3.64	19
16	-	-	-	-	-	-	2

Table 7.34: Standard deviation of recovered extrinsic parameter values for registration of an MR surface to images (b) and (c) from figure 7.8, using $\mathrm{PC}_{\mathrm{mutual}}$ for $\delta t = \pm 4, 8, 12$ and 16 mm and degrees.

	Registration Parameter						Percentage
δt	t_x	t_y	t_z	r_x	r_y	r_z	Success
4	0.28	0.14	0.14	0.15	0.43	0.29	100
8	0.28	0.15	0.16	0.13	0.45	0.30	91
12	1.57	4.08	1.45	0.22	1.83	0.96	25
16	5.47	6.99	2.67	1.98	5.06	5.17	9

Table 7.35: Standard deviation of recovered extrinsic parameter values for registration of an MR surface to images (a) and (c) from figure 7.8, using $\mathrm{PC}_{\mathrm{mutual}}$ for $\delta t = \pm 4, 8, 12$ and 16 mm and degrees.

	Registration Parameter						Percentage
δt	t_x	t_y	t_z	r_x	r_y	r_z	Success
4	0.39	0.14	0.16	0.17	0.55	0.27	100
8	0.39	0.17	0.16	0.19	0.57	0.30	83
12	2.48	3.56	1.21	0.31	3.00	0.88	35
16	5.67	6.88	1.95	0.66	5.46	4.86	13

Table 7.36: Standard deviation of recovered extrinsic parameter values for registration of an MR surface to images (b) and (d) from figure 7.8, using $\mathrm{PC}_{\mathrm{mutual}}$ for $\delta t = \pm 4, 8, 12$ and 16 mm and degrees.

7.5.5.3 Conclusions

Clearly the mutual information based measure PC_{mutual} is performing differently to $PC_{inverse}$ for vertical pairings. In section 7.5.4.3 it was suggested that this could be due to the signal to noise ratio being small as the angular disparity between video views is smallest for vertical views. $PC_{inverse}$ measures a squared error in intensities, and so will be effected by noise. PC_{mutual} however is based on probability of different intensity values and so will not be so badly affected by outliers. It provides a statistical measure of consistency between corresponding intensities, independent of the actual magnitude of the intensity difference. Again this illustrates the further research necessary to establish the lower bounds for parameters such as the angular disparity. It could be the case that using PC_{mutual} and increasing the number of histogram bins as the algorithm approaches registration would improve the sensitivity of the similarity measure and hence improve the accuracy. This may make PC_{mutual} preferable to $PC_{inverse}$ for two view registration.

These results demonstrate that PC_{mutual} is effective for two video views. In general, $PC_{inverse}$ performed quicker, more accurately, precisely and with higher success rate, than PC_{mutual} except for vertical view configurations. In the following experiments, $PC_{inverse}$ was chosen to study the robustness to added image noise, and different surface subsamplings. In addition, $PC_{inverse}$ is tested using datasets of different people. These are preliminary experiments and should also be performed for PC_{mutual}, and this will form part of the future work. PC_{mutual} could be used in situations with two or three views, where speed is not important, or where the reflectance function of the surface is unknown or difficult to model.

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

Figure 7.11: Video images with added Gaussian noise. (a) $\sigma = 0$ pixels (no noise), (b) $\sigma = 16$ pixels, (c) $\sigma = 32$ pixels, (d) $\sigma = 64$ pixels.

7.5.6 Testing The Response To Video Image Noise

7.5.6.1 Methods

Subsequently, the following test was performed to investigate how the performance of the algorithm varied with added video camera noise. The four video images shown in figure 7.8 were taken, and zero mean Gaussian noise of standard deviation 1 intensity value added. Intensity values were clipped to still lie within the range 0-255. This was repeated with noise of standard deviation $2, 4, 8, 16, 32$ and 64 pixel intensity values. For each set of four images, and for misregistration size of $\delta t = \pm 8$ mm and degrees, the algorithm was used to register the video images to the MR surface model. The similarity measure was $\mathrm{PC}_{\mathrm{inverse}}$. For successful registrations, the standard deviation of the parameter values were calculated.

7.5.6.2 Results

The results can be seen in table 7.37. It can be seen that the noise has very little effect. It can be seen in figure 7.11(c) that noise with a standard deviation of 32 or 64 grey values significantly degrades the image. With these levels of added noise tested, the success rate stays at 100% throughout and the standard deviation of the recovered parameters does not change significantly. When the standard deviation of added noise reaches 64 intensity values, the algorithm breaks down. However, the expected amount of noise in any application is much lower than this.

StdDev of	Registration Parameter						Percentage
Added Noise	t_x	t_y	t_z	r_x	r_y	r_z	Success
None	0.21	0.12	0.20	0.24	0.41	0.70	100
1	0.20	0.17	0.16	0.18	0.29	0.22	100
2	0.20	0.14	0.14	0.15	0.25	0.18	100
4	0.20	0.17	0.16	0.18	0.29	0.22	100
8	0.17	0.17	0.11	0.17	0.23	0.19	100
16	0.24	0.22	0.16	0.25	0.31	0.34	100
32	0.31	0.21	0.18	0.29	0.39	0.55	100
64	1.02	0.50	0.49	0.75	1.00	2.22	80

Table 7.37: Standard deviation of recovered extrinsic parameter values using images with noise added.

7.5.6.3 Conclusions

The registration algorithm was tested as zero mean, Gaussian additive noise with standard deviation $1, 2, 4, 8, 16, 32$ and 64 intensity values was added to each of the set of four images. With noise up to 32 intensity values, the noise had little effect as the robustness stayed at 100%, even though it was far more noise than would usually be encountered. The standard deviation of the recovered extrinsic parameters ranges from $0.11 - 0.55$ for these levels of noise.

7.5.7 Testing The Number Of Points, And Z-Buffer Requirements

7.5.7.1 Methods

Recall from section 7.4.3 that it is necessary to calculate which points project to which video image, and that this is done by checking the z-buffer. Calculating the z-buffer is computationally expensive requiring a complete rendering of the surface model, and storing the corresponding depths of the visible points. The speed of the algorithm is limited by the number of points in the surface mesh and the frequency with which the z-buffer is recalculated. The following experiment tests the effects of reducing the number of points on the surface model and the frequency of the z-buffer calculation.

Currently in the gradient ascent search strategy, the z-buffers are calculated for each evaluation of the similarity measure. Calculating the derivative of the similarity measure with finite differences requires 12 evaluations of the similarity measure and testing the

new position requires one evaluation of the similarity measure. So for every step of the gradient ascent search strategy, 13 evaluations of the similarity measure are required, and hence 13 times the number of video views z-buffer calculations.

The similarity measure was altered so that each point was given a flag to denote whether is was visible or not, according to the most recent z-buffer check. If visible, the point was used to evaluate the similarity measure. These flags were updated at either every evaluation of the similarity measure or every $1, 5, 10, 15$ or 20 steps of the gradient ascent search strategy. In addition the surface was sub-sampled by only using every $1, 2, 4, 8, 16, 32, 64$ or 128 points. The full surface has 20853 points, so the sub-sampling reduces this to 10426, 5213, 2606, 1303 , 652, 323 and 161 points for sub-sampling ratios, 2, 4, 8, 16, 32, 64 and 128 respectively.

So, using different frequency of z-buffer calculation and sub-sampling, and for misregistration sizes of $\delta t = \pm 8$ mm and degrees, the four video images from figure 7.8 and the Tricorder^{TM} surface model from section 7.5.2 were registered. The Tricorder^{TM} surface was chosen, as an accurate gold standard was available, whereas with the MR surface it is not. The similarity measure was $\mathrm{PC}_{\mathrm{inverse}}$, using a noise threshold $e^2 = 40$. For successful registrations, the mean and standard deviation of the projection and 3D errors can be calculated, and graphs of error against sub-sampling ratios and z-buffer checking can be plotted.

7.5.7.2 Results

Table 7.38 shows the success rate for the $\mathrm{PC}_{\mathrm{inverse}}$ based registration, as the amount of surface subsampling and the frequency of z-buffer redrawing is changed. It can be seen that with the surface subsampled by ≤ 32 the robustness of the algorithm does not significantly change. Above a factor of 32, (i.e. 64, 128), the robustness decreases. The graph in figure 7.12 shows the mean 3D errors as the surface is subsampled by different amounts, and the frequency of z-buffer redrawing is changed. Each line represents a different amount of z-buffer checking and is described in the key. i.e. "every_evaluation" means that surface points were checked against a z-buffer ever time the similarity measure was evaluated. "every_5_steps" means that surface points were checked against the z-buffer every 5 steps of the gradient ascent search strategy etc. Table 7.39 shows the mean time in seconds taken for the registrations. Times range from 337 seconds to 6 seconds.

Sub-sampling Factor	Redrawing Z-Buffer					
	Every evaluation	Every step	Every 5 steps	Every 10 steps	Every 15 steps	Every 20 steps
1	100	100	100	100	100	100
2	100	100	100	100	100	100
4	100	100	100	100	100	100
8	100	100	98	100	98	100
16	98	97	98	98	98	98
32	100	100	100	100	100	100
64	98	95	97	95	92	94
128	82	83	80	81	78	77

Table 7.38: Success rates for registration using $PC_{inverse}$, with different surface subsampling factors, and frequency of z buffer redrawing.

It can be seen that less z-buffer checking does not necessarily make the algorithm perform much less robustly. With less points however, the algorithm becomes gradually less accurate as projection and 3D errors increase. It is more important to consider the time taken to register, which decreases significantly. This suggests that further work should be done to develop a search strategy that starts with few points and uses progressively more points as registration is approached.

7.5.7.3 Conclusions

The evaluation of the algorithm with respect to the number of points used and the frequency of z-buffer checking was a preliminary test. It was demonstrated that with fewer points, a significant increase in registration speed could be obtained, however, the accuracy reduces. This suggests that a multi-resolution strategy could be used with fewer points at a lower resolution and more points at a higher resolution.

Sub-sampling Factor	Redrawing Z-Buffer					
	Every evaluation	Every step	Every 5 steps	Every 10 steps	Every 15 steps	Every 20 steps
1	337	157	141	139	143	138
2	265	89	79	74	74	74
4	225	53	42	41	40	39
8	200	37	25	23	23	23
16	202	27	17	14	14	13
32	177	22	11	9	9	8
64	165	17	8	8	8	7
128	157	17	7	6	6	6

Table 7.39: Mean time in seconds for photo-consistency based registration. It can be seen that sub-sampling the number of points and reducing the z-buffer redrawing drastically reduces the time to register.

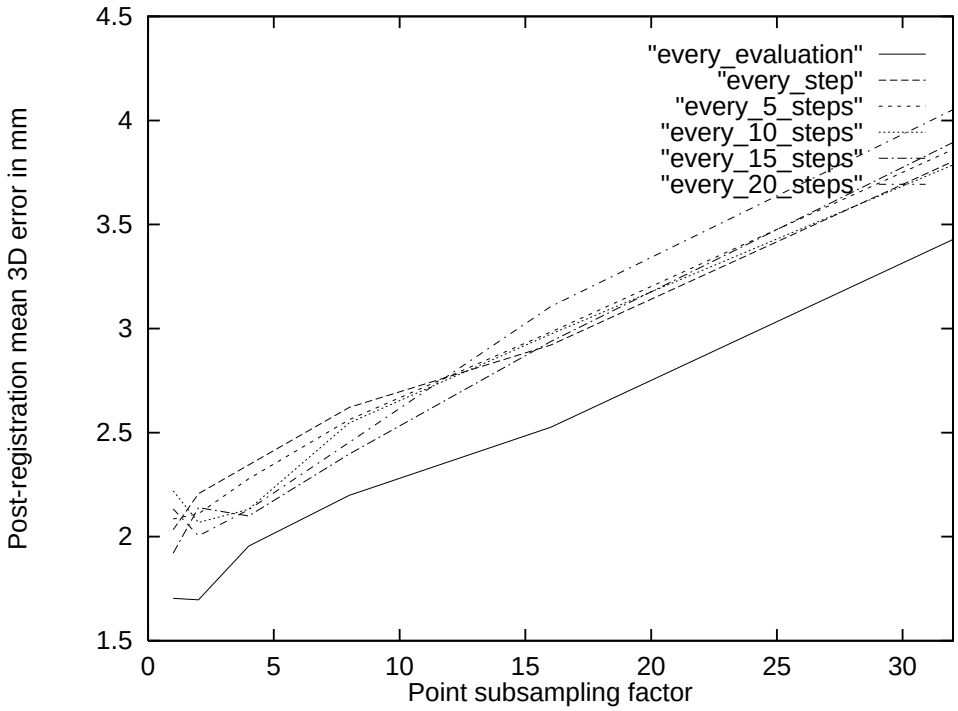


Figure 7.12: Mean 3D error in mm for different amounts of z-buffer testing and sub-sampling. See text section 7.5.7.2.

7.5.8 Registration Of Fifteen datasets

7.5.8.1 Methods

Thus far, the algorithm using $\mathrm{PC}_{\mathrm{inverse}}$ has been tested with either simulations or registering Tricorder$^{\mathrm{TM}}$ or MR surfaces to the same four video images. The following experiment tested the registration algorithm with 10 different Tricorder surfaces and 5 MR surfaces. The calibration performed by the author, for the initial Tricorder$^{\mathrm{TM}}$ based registrations in section 7.5.2 could not be performed here, as a different Tricorder$^{\mathrm{TM}}$ system was being used and the output file format was different. Thus no gold standard registration exists.

In the following experiments, 10 volunteers were imaged with the Tricorder$^{\mathrm{TM}}$ system. The Tricorder$^{\mathrm{TM}}$ calibration data provides an initial registration position that should be close to the true registration. From this starting position, the extrinsic camera parameters are misregistered by $\delta t = \pm 8$mm and degrees. As before, every possible combination (64) of adding ± 8 mm and degrees to the starting position was used. The algorithm then registered the surfaces to the video images using $\mathrm{PC}_{\mathrm{inverse}}$. In addition, five volunteers had also had an MR scan of their head. From each MR scan, a surface was extracted, where the surface represented the face of the volunteer, above the top lip and below the hairline. The MR surface was registered to the Tricorder$^{\mathrm{TM}}$ using a point based method [Arun *et al.*, 1987] to provide an approximate initial registration. From this position, the surface was misregistered by $\delta t = \pm 8$mm and degrees and the algorithm, using $\mathrm{PC}_{\mathrm{inverse}}$, registered the MR surface to the video images.

Although no gold standards were available, the mean position of the registrations for each surface, should be 'close' to that given by the calibration data for the Tricorder$^{\mathrm{TM}}$ system. Therefore, successful registrations were classified as those where none of the extrinsic parameters moved further away from the expected registration position than the size of the initial offset $\delta t = \pm 8$ mm and degrees. For the successful registrations, the standard deviation of the extrinsic parameter values were calculated. Visual inspection was used to check whether the mean registration position did correspond to a good alignment.

7.5.8.2 Results

Table 7.40 shows the results for the 10 volunteers, registering Tricorder$^{\mathrm{TM}}$ surfaces to the corresponding four video images. In all cases, the registration was accurate, robust and

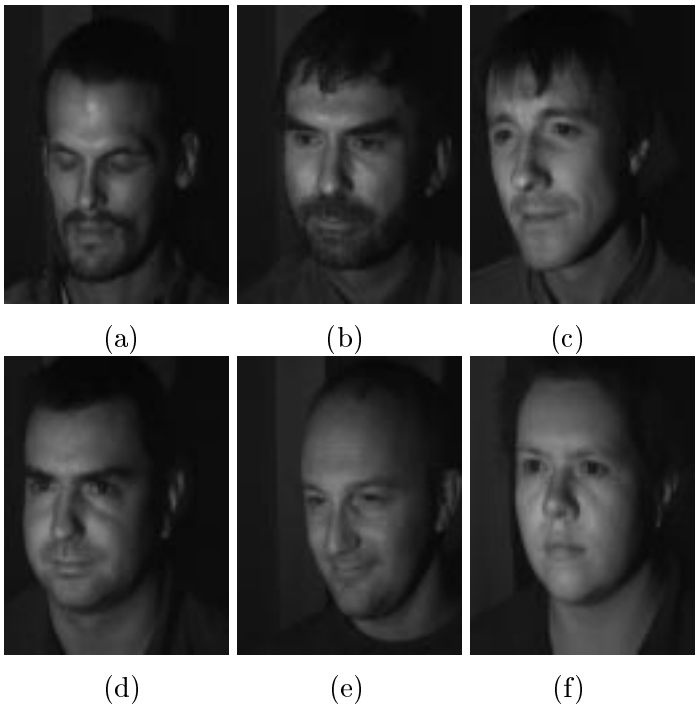


Figure 7.13: Six video images of volunteers. (a) - (e) were used for the MR surface experiments. (a) - (f), plus four others were used for the Tricorder^{TM} surface experiments.

precise. For the five volunteers, registering their MR skin surfaces to the corresponding four video images showed more variable results. The surface of volunteer 1 registered robustly and precisely. For these tests, the success rate was 100% and standard deviation of the recovered extrinsic camera parameters ranged from 0.09 to 0.34. The results for the other four volunteers was not so good. These MR scans were taken, ranging from 3 months to 1 year before the video images, so it is not surprising that the registration is less reliable as the surface could be a significantly different shape to that shown in the video images. In addition the resolution and coverage of the MR images will be different, leading to different amounts of available surface information. In general it was found better to use high resolution triangle mesh surfaces than performing any triangle decimation. The higher the resolution however, the slower the registration.

7.5.8.3 Conclusions

The results from this section suggest that with a recent scan, robust (a success rate of 89% or above), usually precise registration (a low standard deviation of extrinsic parameter values in tables 7.40 and 7.41) should be possible for images similar to those tested. It seems that with MR scans, the performance can vary significantly. Further research should develop a protocol for image acquisition and surface extraction. However, the skin is deformable, which will place a limit on the obtainable registration accuracy.

Volunteer	Registration Parameter						Percentage
Number	t_x	t_y	t_z	r_x	r_y	r_z	Success
1	0.78	0.43	0.14	0.17	0.74	0.16	100
2	0.35	0.56	0.20	0.18	0.43	0.65	100
3	0.49	0.25	0.17	0.14	0.68	0.42	100
4	1.13	0.53	0.16	0.27	1.04	0.32	100
5	0.78	0.36	0.15	0.22	0.71	0.54	100
6	0.46	0.26	0.12	0.11	0.74	0.18	100
7	0.68	0.20	0.18	0.17	0.88	0.28	100
8	0.36	0.30	0.12	0.12	0.40	0.33	100
9	0.68	0.34	0.18	0.18	0.68	0.39	100
10	0.38	0.30	0.12	0.12	0.44	0.22	100

Table 7.40: Standard deviation of recovered extrinsic parameter values for the registration of 10 volunteers Tricorder^{TM} surfaces to the corresponding set of four video images.

Volunteer	Registration Parameter						Percentage
Number	t_x	t_y	t_z	r_x	r_y	r_z	Success
1	0.27	0.32	0.09	0.20	0.34	0.19	100
2	1.41	1.13	0.29	0.45	2.04	0.36	98
3	2.37	3.07	0.61	0.70	2.32	1.17	89
4	1.84	0.73	0.19	0.29	1.99	0.38	95
5	2.65	0.66	0.16	0.30	2.54	0.55	100

Table 7.41: Standard deviation of recovered extrinsic parameter values for the registration of 5 volunteers MR surfaces to the corresponding set of four video images.

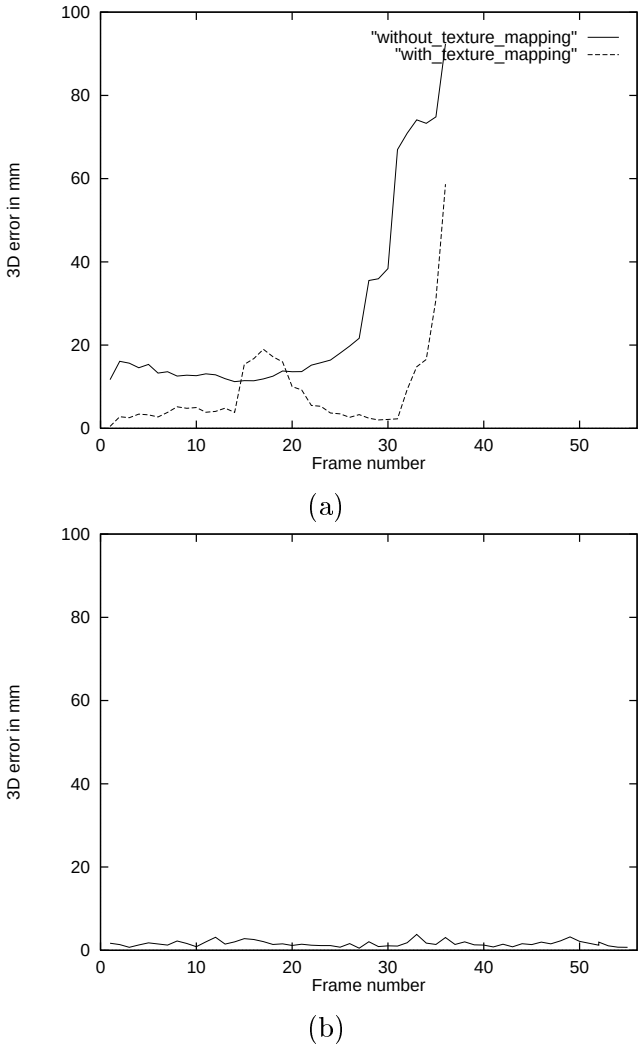
7.5.9 A Comparison With A Surface Based Registration Technique

7.5.9.1 Methods

The experiment of section 6.4.3 was repeated to compare the photo-consistency based registration with the surface based registration. To recap, a series of 55 'images' of a volunteer were taken with a Tricorder^{TM} S4m system. Each 'image' capture consists of a set of four video images, taken using four video cameras, and illuminated with pseudo-random patterned light, and a set of four video images, taken using the same four cameras, but illuminated with plain white light. The S4m system then uses the four images, taken with the patterned light to reconstruct a surface model of the viewed scene. Using this system, 56 images of a volunteer where taken while the volunteer moved his head slowly. For each frame the surface was reconstructed. The first surface was clipped, and then registered serially to the subsequent 55 surfaces using a surface based registration algorithm [Maurer Jr. *et al.*, 1998]. The same surface was also registered to the corresponding series of plainly illuminated video images using $\mathrm{PC}_{\mathrm{inverse}}$. This experiment was performed with the value e in equation (7.8) set so that $e^2 = 100$. Also a point was used to evaluate the similarity measure if it projected to at least two views, rather than if it projected to all four views. Increasing e provided a smoother search space which made the tracking more capable of tracking larger movements between frames. Using points that project to at least two views added extra robustness, as when the volunteer rotated by a large angle, all the points on one side of the face might not project to all views. Over the series of images, the 3D error, i.e. difference in registration transformation of the surface based registration and the photo-consistency based registration was calculated.

7.5.9.2 Results

The graph in figure 7.14 shows that the tracking for $\mathrm{PC}_{\mathrm{inverse}}$ was successful over the sequence of 55 images. This graph should be compared with the graph in figure 6.15. It can be seen from the graph in figure 6.15 that the original mutual information based algorithm from chapter 5 failed, the texture mapping based algorithm of chapter 6 did moderately well but fails in some areas. However this new algorithm tracks exceedingly well. This is verified in figure 7.15. Images (a) - (f) show a red wire frame representation of the surface overlaid on the corresponding video image. Images (a) - (f) correspond to frames 0 (i.e. the starting point), 11, 17, 36, 46, and 55 (i.e. the finish) respectively.



(a)

(b)

Figure 7.14: Graphs of 3D error in mm between different intensity based registration algorithms, and a surface based method [Maurer Jr. *et al.*, 1996]. Graph (a) shows the graph from figure 6.15, reproduced here for comparison. Graph (a) shows the performance of the non-texture mapping algorithm from chapter 5, and the performance of the texture mapping algorithm from chapter 6, when compared to the surface based tracking algorithm. Graph (b) shows the performance of the photo-consistency based algorithm, using $\mathrm{PC}_{\mathrm{inverse}}$.

Figure 7.15: The surface model in the registered pose is shown displayed as a red wireframe rendered surface overlayed on the video image. Images (a) - (f) show video images 0 (i.e. the starting point), 11, 17, 36, 46, and 55 (i.e. the finishing point). It can be seen that tracking is successful throughout this sequence as the red wireframe is in the correct position in each image.

7.5.9.3 Conclusions

It can be seen that this new algorithm, photo-consistency using $\mathrm{PC}_{\mathrm{inverse}}$, is more accurate relative to the surface based registration, and more robust than the algorithm of the previous chapters. As discussed in section 6.4.3.1, the surface based algorithm is not a gold standard. Nevertheless, the mean 3D error between the surface based and photo-consistency based registration estimates is 1.6mm. Both the surface based and photo-consistency based registrations appear visually accurate.

7.6 Summary

In this chapter a new method for 2D video - 3D surface model registration has been proposed. From the experiments performed, the following conclusions are made.

The method of registration is based around the concept of photo-consistency, a term used in surface reconstruction literature [Kutulakos and Seitz, 1998]. It is the idea of using photo-consistency as a method for registration that is novel. In this chapter, five similarity measures were described, $\mathrm{PC}_{\mathrm{squared}}$, $\mathrm{PC}_{\mathrm{inverse}}$, $\mathrm{PC}'_{\mathrm{squared}}$, $\mathrm{PC}'_{\mathrm{inverse}}$ and $\mathrm{PC}_{\mathrm{mutual}}$ for use in different experiments. The simulations in section 7.5.1 showed that using $\mathrm{PC}_{\mathrm{squared}}$, $\mathrm{PC}_{\mathrm{inverse}}$ and $\mathrm{PC}_{\mathrm{mutual}}$ works well with a simulation of two cameras and a light source fixed between each camera. With a different light source for each camera, the measures $\mathrm{PC}'_{\mathrm{squared}}$, $\mathrm{PC}'_{\mathrm{inverse}}$ and $\mathrm{PC}_{\mathrm{mutual}}$ did work, but not nearly as well.

The mutual information based photo-consistency measure $\mathrm{PC}_{\mathrm{mutual}}$ provided a link between this chapter and the previous. In principle, $\mathrm{PC}_{\mathrm{mutual}}$ can provide a flexible similarity measure for the case where the number of views is small or the relationship between image intensities in different views in unknown or hard to model. $\mathrm{PC}_{\mathrm{mutual}}$ registration achieved accuracies of the order of 1.73 - 2.69 mm and success rates between 87 and 100% for registration of MR and Tricorder$^{\mathrm{TM}}$ surfaces to two video views. Overall, $\mathrm{PC}_{\mathrm{inverse}}$ outperformed both $\mathrm{PC}_{\mathrm{mutual}}$ and $\mathrm{PC}_{\mathrm{squared}}$, giving registration accuracies of 1.45 mm and a good success rate, typically 100%, even with misregistration size of $\delta t = \pm 16$mm and degrees from the gold standard position. $\mathrm{PC}_{\mathrm{inverse}}$ also proves to be robust to increasing noise in the video images, and still performs well when the number of surface points is decreased significantly. It is expected that $\mathrm{PC}_{\mathrm{mutual}}$ would also be robust to the addition of noise in the video images, but the performance may degrade when the number of points in the surface model is reduced.

In a comparison of tracking algorithms, $PC_{inverse}$ performed accurately and reliably relative to a surface based registration technique, where the texture mapped tracking algorithm from chapter 6 failed after 14 frames, and the non-texture mapped tracking algorithm of chapter 5 failed to work with the volunteer data. Over the tracking sequence, $PC_{inverse}$ gave a mean 3D error relative to the surface based algorithm of 1.6mm, and the tracking results appearing visually accurate.

Comparing $PC_{inverse}$ with the other similarity measures tested in chapter 5, reveals the following: For the Tricorder^{TM} surface, and for misregistration sizes of $\delta t = \pm 8$ mm and degrees, the similarity measures mutual information, normalised mutual information, normalised cross correlation, and gradient correlation gave mean 3D errors of 2.77, 2.53, 3.13, 2.64 mm respectively. $PC_{inverse}$ however gave a mean 3D error of 1.59 mm. The new method $PC_{inverse}$ is more accurate and much quicker. Furthermore, $PC_{inverse}$ should provide improved performance as the texture of the surface increases. The rendering based methods mutual information, normalised mutual information, normalised cross correlation and gradient correlation will all degrade as the rendering looks less like the video image. In summary, the tests so far have demonstrated an effective new algorithm, that performs well for images of the face. It also leads to many more exciting paths of research for 2D-3D medical and non-medical image or image to model registration algorithms.

It has been demonstrated that a similarity measure can be developed which registers a 3D model to two or more optical images without requiring a rendered image to be calculated. Using such a similarity measure does provide more accurate and more reliable registration than the rendering based method of previous chapters.

Part III

Conclusions

Chapter 8

Conclusions

8.1 Summary Of Findings

A summary of the main findings of this thesis now follows. The summary has been subdivided according to the four main chapters of developed algorithms and experimental work.

8.1.1 Single View Registration

Chapter 4 presented the implementation and assessment of a mono-view algorithm based on currently existing ideas in the literature. Registration was performed by producing rendered images of a surface model extracted from a 3D image, and measuring the similarity of the rendered and video images using mutual information. Mutual information was maximised with respect to the pose parameters until the optimum pose was found. It was concluded that the mutual information of a single rendered and video image pair, optimised using a gradient ascent search strategy was not sufficient to register a video image with a 3D image. The algorithm was tested using images of a plastic skull phantom, and the algorithm failed to recover translations along the optical axis of the video camera. Furthermore, the algorithm exhibited unsatisfactory performance with different video images taken with changing focal lengths and fields of view.

8.1.2 Multiple View Registration

Chapter 5 presented an extension of the mono view framework. Three simple methods for combining the information from multiple rendered and video images were tested. These were called, 'adding' the information from multiple views, 'combining' the information from multiple views, or Leventon's 'alternating' method [Leventon *et al.*, 1997] It was

concluded that the adding and combining methods performed similarly and both were superior to the alternating method for these experiments. A significant finding was that with the proposed method, i.e. matching multiple video images to multiple rendered images, it was indeed necessary to know where the light sources were relative to the video cameras in order to mimic this when producing a rendering. Without knowledge of the light source position relative to the video cameras, inaccurate registration will result. The experiments, also presented in [Clarkson *et al.*, 1999a] demonstrated that with a good light source position relative to the video cameras, registration accuracy in terms of the 3D error was approximately 1 mm for a skull phantom experiment. However, even with these improvements, it was concluded that the mutual information of two or more video images and rendered image pairs, optimised using a gradient ascent search strategy was insufficient to register the video images to a 3D volume image. This was because the algorithm still did not perform well with images that had different fields of view, focal length, and performed inaccurately with images of a volunteers face.

8.1.3 Using Texture Mapping For Tracking

Chapter 6 focussed on tracking an object on the condition that an initial alignment between a surface extracted from a 3D image and video images had already been performed. In this case, pixel grey values in the first video image could be directly associated with points in the 3D model. The registration provided information describing what a 3D point in the surface model looked like in 'real life', and this information was not present in the original 3D tomographic image or surface model. Information from the initial registered video views was texture mapped onto the model and used to assist registration to subsequent video frames. With small movements, the proposed tracking algorithm performed moderately well. The algorithm will fail to track if there are large movements in between video frames, and if the object rotates through an angle of greater than 20 degrees or so. It was concluded that using texture mapping to assist a tracking algorithm does significantly improve the accuracy and robustness of mutual information based tracking when compared with the non-texture mapping algorithm of the previous two chapters.

8.1.4 Photo-Consistency, A Novel Measure Of Image Alignment

Chapter 7 described the most significant and novel research of this thesis. A new similarity measure which uses the concept of photo-consistency was developed. Several different formulations of similarity measure were studied, including a mutual information based measure. The mutual information based measure worked well when tested with two views, and when there existed a simple relationship between optical image intensities. In the general case, i.e. with many video views, mutual information will not be suitable due to the fact that in order to calculate the similarity measure for N video images, an N dimensional probability distribution will be necessary. As N increases, evaluating the similarity measure will be increasingly more computationally expensive, and increasingly unreliable due to the lack of information with which to calculate the probability distributions.

In section 5.3.1, three methods for calculating the mutual information of N pairs of video and rendered images were discussed. These were called, 'adding', 'combining' and 'alternating'. It was mentioned in section 7.4.4, that these three methods were applicable, as in the experiments demonstrated, the rendered image was made to look as close as possible (whilst using a simple lighting model) to the corresponding video image and it was known, which video image should match which rendered image. In chapter 7, a photo-consistency measure, using mutual information was tested ($\mathrm{PC}_{\mathrm{mutual}}$). Whilst the photo-consistency of many video images could be calculated using $\mathrm{PC}_{\mathrm{mutual}}$ by adapting the 'adding', 'combining' or 'alternating' methods, several problems would need resolving. Using 2D histograms to calculate the mutual information would require that a decision be made as to which pairs of images to compare. This would have to be either exhaustive, using every possible combination, or arbitrarily chosen and hence possibly biased. In addition, it was also shown that the mutual information based measure only worked well in the case of a simple relationship between the corresponding video intensities in the two views. Furthermore, the mutual information measure resulted in a smaller capture range than $\mathrm{PC}_{\mathrm{inverse}}$, but did appear to present a smooth cost function. To summarise, with two views, the mutual information based measure $\mathrm{PC}_{\mathrm{mutual}}$ is viable, but with N video views, the mutual information method is insufficient at it will be too slow, unreliable, have a small capture range, and is an inferior option when compared with $\mathrm{PC}_{\mathrm{inverse}}$.

The similarity measure $PC_{inverse}$ was found to perform best. The similarity measure was evaluated by registering MR and Tricorder^{TM} surfaces to a variety of video images. The algorithm was tested with two or four views, different levels of video image noise, different z-buffer checking, and images of different people. The measure performed well in nearly all cases. It was demonstrated that the performance of the algorithm did depend on the number of video cameras, and their relative position to the object. Future work will investigate this further.

Chapter 7 demonstrates that when performing this type of registration, if some knowledge of the relationships between the intensities in each video image is available, then it should be used. For example, in chapter 7, the surface was assumed to exhibit Lambertian reflection. Thus, utilising this knowledge and calculating $PC_{inverse}$ resulted in better registration performance than trying to use a general purpose measure PC_{mutual}, which makes no use of any lighting model.

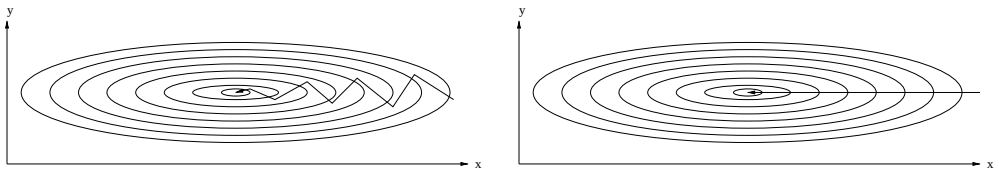
This new method, potentially, seems applicable to many areas where a model is to be related to optical image information. This may include applications in robotics, where a robot must navigate though a known environment, in computer vision where a camera system must track known objects, in telemanipulation or any robot interacting with its environment e.g. computer assisted manufacturing, or manipulation of objects in a hostile or remote environment, or computer assisted maintenance.

8.1.5 Answer To The Main Hypothesis

The main hypothesis of this thesis, originally stated in section 1.3 was:

- It is possible to develop an intensity based algorithm to register multiple video images to 3D models or images, that is sufficiently accurate, precise and robust, to be suitable for applications such as radiotherapy patient positioning and also for image guided ENT (ear, nose and throat) surgery, neurosurgery or craniofacial surgery.

In this thesis, various registration methods have been studied. The similarity measure $PC_{inverse}$, developed in chapter 7 and using the concept of photo-consistency has been demonstrated to be the most accurate, precise and robust. Accuracy for registration of Tricorder^{TM} surfaces was evaluated to have a mean 3D error of approximately 1.5 mm. The algorithm was precise for registration of Tricorder^{TM} and MR surfaces with the standard deviation of all six extrinsic camera parameters usually < 0.5 mm and degrees. The algorithm was robust as it repeatedly registered for different misregistration sizes, different numbers of video views (2 or 4 views were tested), with significant noise added to the video images, and with different amounts of surface subsampling. Although, no clinical applications were tested, it would seem feasible to expect that this algorithm will accurately verify the position of a patient's head on a radiotherapy treatment bed, accurate to 1.5 mm. In addition, for ENT, craniofacial or neurosurgery, it would also seem feasible that registration to head images would be accurate enough for providing a surgeon with overlay images such as that in figure 7.9. Thus the hypothesis stated above has been demonstrated to be true.



(a) (b)

Figure 8.1: Graphs of similarity measure (z-axis, pointing out of page) for two parameters x and y. Iso-lines show lines of constant similarity. (a) gradient ascent can take many steps, (b) a better strategy would calculate the direction of the valley and proceed straight down it.

8.2 Future Work

The work described in this thesis was carried out with clinical applications in mind. Currently however, there remains further work that needs to be carried out before clinical solutions can be realised. This will involve developing the algorithm, to apply the algorithm to the intended clinical applications, and to compare performance with other techniques to establish where this algorithm should be applied and how it compares with other pose estimation and model based recognition methods. Some possible directions for future research are now described.

8.2.1 Algorithm Improvements

8.2.2 Search Strategies

The final registration algorithm maximises $\mathrm{PC}_{\mathrm{inverse}}$, as defined in equation (7.8), with respect to the extrinsic camera parameters by using a gradient ascent search strategy. Gradient ascent is simple to implement, yet widely recognised as a relatively poor algorithm [Press *et al.*, 1992]. Consider graphs (a) and (b) in figure 8.1. Graph (a) represents a gradient ascent method. The algorithm calculates the local gradient, and heads in that direction, i.e. each segment of the zig-zag line crosses the iso-lines at a right angles. However, the valley is horizontal on this figure. A more intelligent algorithm might take a few steps to calculate the true valley direction and proceed quickly to the optima, as shown in graph (b). Conjugate gradient methods could be used to achieve this or other popular non-linear optimisation strategies like Levenberg Marquardt [Press *et al.*, 1992]. For obtaining more global convergence, methods such as simulated annealing or genetic algorithms may prove interesting research.

8.2.2.1 Segmentation Free Registration

Intensity based registration was developed in order to avoid segmentation as a pre-processing step, as any errors in segmentation would affect final registration accuracy. In this thesis, no video image segmentation for the purpose of registration, has been performed. However, the algorithms used a surface, segmented from the 3D image. The images were of a skull phantom, or the skin surface of a volunteer, so this segmentation was usually quite straight forward. However, a final goal would be to completely avoid segmentation.

Consider the difference between a surface rendering paradigm and a volume rendering paradigm. These are sometimes called object order or image order rendering respectively [Foley *et al.*, 1990]. In surface based rendering, specific, points, lines or triangles representing an object are defined in some 3D coordinate system, and these are projected onto the image plane. In volume rendering, for each pixel in the 2D image, a ray is projected back into 3D space and through a volume image. According to some opacity or gradient transfer function the voxel intensities are converted to colours and opacities and ultimately the 2D pixel is assigned a colour. The volume rendering approach is performing a segmentation, but it is a more flexible framework, which can easily be modified as the opacity of a point can be determined by the underlying voxel intensity, or by local gradient. Therefore, the registration could well be performed by using a volume rendering approach. For each voxel in the volume, a weighting function should be defined which determines how likely the point is to be on a surface. A volume rendering approach is then used to select likely surface points from a volume, and then these points are projected to each video image plane. The similarity measure $\mathrm{PC}_{\mathrm{inverse}}$ from equation (7.8) would be modified to

$$\mathrm{PC}_{\mathrm{inverse}} = \frac{1}{I} \sum_{i=1}^{I} w_i \frac{e^2}{e^2 + e_i^2} \tag{8.1}$$

where the summation for I is performed for all voxels that are visited when performing the volume rendering, and w_i is the weighting factor relating to surface strength. It may also be possible to remove the z-buffering completely. It may be the case that $\mathrm{PC}_{\mathrm{inverse}}$ would still work if every point or a subsampled set of points in the volume were projected into each video image. Methods such as these may realise the goal of segmentation free registration, using multiple surfaces, i.e. skin and bone simultaneously.

8.2.2.2 Considering Local Variations

The similarity measure $PC_{inverse}$ was evaluated by projecting points into each video image, and linearly interpolating the intensity at that point. However, each image intensity will be affected by noise. In surface reconstruction [Trucco and Verri, 1998] or intensity based tracking algorithms [Uenohara and Kanade, 1995] a small window is computed around each point of interest. When assessing similarity between images, the image intensities contained within each window are compared, as opposed to single intensities at a point. It may be the case that better registration could be performed by computing a window around each projected point, and seeing how photo-consistent the image intensities are within each window. This potentially opens up many important questions, i.e. how big a window, how many points, how computationally expensive is this? Also, if a window around each point is considered, many different similarity measures could be investigated, e.g. measure the correlation of each window, perform spectral analysis of each window and compare the coefficients.

8.2.3 Applications

In order to apply the proposed algorithms to applications, clinical or otherwise, it is necessary to study the imaging conditions and constraints of each application. A few suggested areas of potential research are described below.

8.2.3.1 Verification Of Patient Position For Radiotherapy Treatment

One of the most likely applications may be to use this registration algorithm to determine whether a patient, undergoing radiotherapy treatment is lying in the correct position on the treatment bed. To do this, the patient must be one who has had a pre-treatment CT/MR scan. This is most applicable to patients being given treatment to the head, and lying in a supine position. Current clinical protocol is to take a plaster of Paris mould of the head, and construct a plastic shell, with which to restrain the patient. Constructing the mould is uncomfortable, and for many patients, will result in a poorly fitting shell. If the shell is constructed several weeks prior to treatment, the patient may well put on or lose weight between shell construction and treatment. This will cause the shell to fit poorly leading to inaccurate patient positioning. Video cameras could be attached to the linear accelerator, and calibrated accordingly. The patient would then simply lie on the bed and the computer register a skin surface extracted from the CT/MR scan

to the video images of the patient. The patient bed could then be adjusted until the patient was in the correct place. The same algorithm could be used to stop treatment if the patient moves beyond specified tolerance limits. The algorithm seems well suited to this application. Possible areas of research would be to make sure that the equipment can be fitted and used around existing radiotherapy equipment, and that the use of the algorithm makes the patient positioning, quicker, more accurate, and less uncomfortable for the patient.

8.2.3.2 Surgical Guidance

This registration could be used for augmented reality applications such as computer assisted surgery, especially for craniofacial, neurosurgery or ENT surgery. Using displays similar to the figures in section 7.9, a surgeon could be guided towards tumours, avoiding critical structures such as blood vessels. Direct application to image guided surgery projects such as [Edwards *et al.*, 1999b] seems a possibility.

Important areas of research still to be studied are the effects on registration of specular reflection, deformation between pre-operative images and intra-operative images, the effect of drapes, other occlusions and most importantly, which surfaces are available for registration within a surgical environment. Furthermore, it is not only the choice of available surface, but also the featuredness in terms of shape and intensity variation within the field of view of the video cameras that will be of interest.

8.2.3.3 Endoscope Views

Conceptually it may be possible to use this algorithm to register pre-operative CT or MR to endoscope views for the purpose of image guidance within minimally invasive surgery. The small field of view of an endoscope makes it difficult to determine the orientation of the observed vessel relative to the surrounding anatomy. If a rigid endoscope was tracked, then several video images could be grabbed by moving the camera around. These video images could then be registered to the pre-operative MR/CT. It is likely that an initial estimate would have to be very close to the true registration for this to work. It may be more feasible to register using images with many features, and then continue to register as the endoscope is moved towards a target of interest, i.e. tracking. Areas of research may involve determining whether there would be enough information, both surface curvature, and video image intensity, within any of the video images, for this to be at all possible.

8.2.3.4 Computer Vision

The same registration algorithm could be used within many computer vision applications as a general purpose pose estimation technique. The 3D model could be derived from for instance a CAD design. Possible applications would include robot based production line inspection or even construction. A further use may be in computer assisted training. Consider a trainee engineer, performing routine maintenance on an aircraft engine. Two or three video cameras, mounted nearby, or even on a head mounted rig may be able to register a CAD model of the engine to the video views. Computer graphics could be overlayed on a particular video view or head up display to point towards objects of interest or instruct the engineer as to what task to perform next. This is purely speculative, but possible areas of research would be to look at the different lighting conditions, the effects of dirt and obstructions within a non-surgical environment and so on.

8.3 Conclusions

The use of photo-consistency has been shown to be a powerful paradigm for 2D optical image to 3D MR/CT image registration. This thesis concentrated on the algorithm development rather than on clinical applications. The algorithm performed accurately and robustly in the performed experiments. The development of this paradigm has the prospect of many exciting areas for novel research and applications.

Bibliography

[Abidi and Chandra, 1995] M. A. Abidi and T. Chandra. A new efficient and direct solution for pose estimation using quadrangular targets: Algorithm and evaluation. *IEEE Transactions on Pattern Analysis And Machine Intelligence*, 17(5):534–538, 1995.

[Aloimonos, 1990] J. Y. Aloimonos. Perspective approximations. *Image and Vision Computing*, 8(3):177–192, August 1990.

[Alter, 1994] T. D. Alter. 3-d pose from 3 points using weak-perspective. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(8):802–808, August 1994.

[Arun *et al.*, 1987] K. S. Arun, T. S. B Huang, and S. D. Blostein. Least-squares fitting of two 3D point sets. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 9(5):698–700, 1987.

[Ayache and Lustman, 1991] N. Ayache and F. Lustman. Trinocular stereo vision for robotics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(1):73–85, January 1991.

[Ballard and Brown, 1982] D. H. Ballard and C. M. Brown. *Computer Vision*. Number ISBN 0-13-165316-4. Prentice-Hall Inc., Englewood Cliffs, New Jersey 07632, 1982.

[Barequet and Sharir, 1997] G. Barequet and M. Sharir. Partial surface and volume matching in three dimensions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(9):929–948, 1997.

[Bascle and Deriche, 1994] B. Bascle and R. Deriche. Region tracking through image sequences. Technical Report 2439, INRIA, December 1994.

[Beardsley *et al.*, 1992] P. Beardsley, D. Murray, and A. Zisserman. Camera calibration using multiple images. In *Computer Vision – Proc. 2nd European Conference on*

Computer Vision (ECCV'92), volume 588 of *Lecture Notes in Computer Science*, pages 312–320. Springer-Verlag, 1992.

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

[Betting and Feldmar, 1995] F. Betting and J. Feldmar. 3D-2D projective registration of anatomical surfaces with their projections. In Y. Bizais, C. Barillot, and R. Di Paola, editors, *Information Processing in Medical Imaging*, pages 275–286, 1995.

[Betting *et al.*, 1995] F. Betting, J. Feldmar, N. Ayache, and F. Devernay. A new framework for fusing stereo images with volumetric medical images. In N. Ayache, editor, *Proc. International Conference on Computer Vision, Virtual Reality and Robotics in Medicine (CVRMed'95)*, pages 30–39. Springer-Verlag, 1995.

[Birchfield, 1997] S. Birchfield. An elliptical head tracker. In *31st Asilomar Conference on Signals, Systems, and Computers, November*, 1997.

[Birchfield, 1998] S. Birchfield. Elliptical head tracking using intensity gradients and color histograms. In *IEEE Conference on Computer Vision And Pattern Recognition, Santa Barbera, California*, 1998.

[Bose and Amir, 1990] C. B. Bose and I. Amir. Design of fiducials for accurate registration using machine vision. *IEEE Transactions on Pattern Analysis And Machine Intelligence*, 12(12):1196–1200, 1990.

[Brown and Boult, 1996] L. M. Brown and T. E. Boult. Registration of planar film radiographs with computed tomography. In *Proc. of the IEEE workshop on Mathematical Methods in Biomedical Image Analysis*, pages 42–51, 1996.

[Brown, 1992] L. G. Brown. A survey of image registration techniques. *ACM Computing Surveys*, 24(4):325–375, December 1992.

[Burt *et al.*, 1982] P. J. Burt, C. Yen, and Xu X. Local correlation measures for motion analysis: a comparative study. *IEEE Computer in Pattern Recognition and Image Processing*, pages 269–274, 1982.

[Catmull, 1975] E. Catmull. Computer display of curved surfaces. In *Proceedings of the Conference on Computer Graphics, Pattern Recognition and Data Structures*, pages 11–17, New York, May 1975. IEEE Computer Society.

[Champleboux *et al.*, 1992] G. Champleboux, S. Lavallee, P. Sautot, and P Cinquin. Accurate calibration of camera and range imaging sensors: The NPBS method. In *Proceedings of the 1992 IEEE International Conference On Robotics And Automation, Nice, France*, pages 1552–1557, 1992.

[Chiorboli and Vecchi, 1993] G. Chiorboli and G. P. Vecchi. Comments on "design of fiducials for accurate registration using machine vision". *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 15(12):1330–1332, 1993.

[Clarkson *et al.*, 1998] M. J. Clarkson, D. Rueckert, D. L. G. Hill, and D. J. Hawkes. Registration of multiple video images to pre-operative data for image guided surgery. In *Medical Image Understanding and Analysis*, pages 73–76, 1998.

[Clarkson *et al.*, 1999a] M. J. Clarkson, D. Rueckert, D. L. G. Hill, and D. J. Hawkes. Registration of multiple video images to pre-operative CT for image guided surgery. In K. M. Hanson, editor, *Proceedings of SPIE, Medical Imaging 1999: Image Processing*, volume 3661, pages 14–23, 22-25 February, San Diego, California 1999.

[Clarkson *et al.*, 1999b] M. J. Clarkson, D. Rueckert, King. A. P., P. J. Edwards, D. L. G. Hill, and D. J. Hawkes. Using texture mapping to register video images to tomographic images by optimising mutual information. In D. J. Hawkes, D. L. G. Hill, and R. P. Gaston, editors, *Medical Image Understanding and Analysis*, pages 29–32, July 1999.

[Clarkson *et al.*, 1999c] M. J. Clarkson, D. Rueckert, King. A. P., P. J. Edwards, D. L. G. Hill, and D. J. Hawkes. Registration of video images to tomographic images by optimising mutual information using texture mapping. In C. Taylor and A. Colchester, editors, *Medical Imaging, Computing and Computer-Assisted Intervention - MICCAI '99*, volume 1679 of *Lecture Notes In Computer Science*, pages 579–588. Springer-Verlag, September 1999.

[Colchester *et al.*, 1994] A. C. F. Colchester, J. Zhao, C. Henri, R. L. Evans, P Roberts, N. Maitland, D. J. Hawkes, D. L. G. Hill, A. J. Strong, D. G. Thomas, M. J. Gleeson, and T. C. S. Cox. Craniotomy simulation and guidance using a stereo video based tracking system (VISLAN). In R Robb, editor, *Visualization In Biomedical Computing*, pages 541–551. SPIE, 1994.

[Colchester *et al.*, 1996] A. C. F. Colchester, J. Zhao, S. K. Holton-Tainter, C. J. Henri, N. Maitland, P. T. E. Roberts, C. G. Harris, and R. J. Evans. Development and

preliminary evaluation of VISLAN, a surgical planning and guidance system using intra-operative video imaging. *Medical Image Analysis*, 1(1):73–90, 1996.

[Collignon *et al.*, 1995] A. Collignon, F. Maes, D. Delaere, D. Vandermeulen, P. Seutens, and G. Marchal. Automated multimodality image registration using information theory. In Y. Bizais, C. Barillot, and R. Di Paola, editors, *Information Processing in Medical Imaging*, pages 263–274, 1995.

[Collignon *et al.*, 95] A. Collignon, D. Vandermeulen, P. Suetens, and G. Marchal. 3D multi-modality medical image registration using feature space clustering. In N. Ayache, editor, *Computer Vision, Virtual Reality and Robotics in Medical Imaging*, volume 905, pages 195–204. Springer-Verlag, Berlin, 95.

[DeMenthon and Davis, 1992a] D. F. DeMenthon and L. S. Davis. Exact and approximate solutions of the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(11):1100–1105, November 1992.

[DeMenthon and Davis, 1992b] D. F. DeMenthon and L. S. Davis. Model-based object pose in 25 lines of code. In *Computer Vision – Proc. 2nd European Conference on Computer Vision (ECCV'92)*, volume 588 of *Lecture Notes in Computer Science*, pages 335–343. Springer-Verlag, 1992.

[Duda and Hart, 1973] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis.* John Wiley and Sons, New York, 1973.

[Edwards *et al.*, 1999a] P. J. Edwards, A. P. King, D. J. Hawkes, O. Fleig, C. R. Maurer Jr., D. L. G. Hill, M. R. Fenlon, D. A. de Cunha, R. P. Gaston, S. Chandra, J. Mannss, A . J. Strong, M. J. Gleeson, and T. C. S. Cox. Stereo augmented reality in the surgical microscope. In J. D. Westwood, H. M. Hoffman, R. A. Robb, and D. Stredney, editors, *Medicine Meets Virtual Reality (MMVR)*, pages 102–108. IOS Press, 1999.

[Edwards *et al.*, 1999b] P. J. Edwards, A. P. King, C. R. Maurer Jr., D. A. DeCunha, D. J. Hawkes, D. L. G. Hill, R. P. Gaston, M. J. Clarkson, M. R. Pike, M. R. Fenlon, S. Chandra, A. J. Strong, C. L. Chandler, and M. J. Gleeson. Design and evaluation of a system for microscope-assisted guided interventions (magi). *IEEE Transactions on Medical Imaging*, 1999.

[Edwards *et al.*, 1999c] P. J. Edwards, A. P. King, C. R. Maurer Jr., D. A. DeCunha, D. J. Hawkes, D. L. G. Hill, R. P. Gaston, M. R. Fenlon, S. Chandra, A. J.

Strong, C. L. Chandler, R. Aurelia, and M. J. Gleeson. Design and evaluation of a system for microscope-assisted guided interventions (magi). In C. Taylor and A. Colchester, editors, *Medical Imaging, Computing and Computer-Assisted Intervention - MICCAI '99*, volume 1679 of *Lecture Notes In Computer Science*, pages 842–851. Springer-Verlag, September 1999.

[Edwards *et al.*, 1999d] P. J. Edwards, A. P. King, C. R. Maurer Jr., D. A. DeCunha, M. R. Pike, D. J. Hawkes, D. L. G. Hill, R. P. Gaston, M. R. Fenlon, A. J. Strong, C. L. Chandler, and M. J. Gleeson. A locking acrylic dental stent for registration and tracking during image-guided surgery of the head. In R. P. Gaston, D. L. G. Hill, and D. J. Hawkes, editors, *Proc. Medical Image Understanding and Analysis*, 1999.

[Eggert *et al.*, 1997] D. W. Eggert, A. Lorusso, and R. B. Fisher. Estimating 3-d rigid body transformations: a comparison of four major algorithms. *Macine Vision and Applications*, 9:272–290, 1997.

[Faig, 1975] I. W. Faig. Calibration of close-range photogrammetric systems: Mathematical formulation. *Photogrammetric Engineering and Remote Sensing*, 41(12):1479–1486, December 1975.

[Faugeras and Robert, 1994] O. Faugeras and L. Robert. What can two images tell us about a third one? In Jan-Olof Eklundh, editor, *Computer Vision – Proc. 3rd European Conference on Computer Vision (ECCV'94)*, volume 800 of *Lecture Notes in Computer Science*, pages 485–492. Springer-Verlag, 1994.

[Faugeras *et al.*, 1992] O. D. Faugeras, Q. T. Long, and S. J. Maybank. Camera self-calibration: Theory and experiments. In *Computer Vision – Proc. 2nd European Conference on Computer Vision (ECCV'92)*, volume 588 of *Lecture Notes in Computer Science*, pages 321–334. Springer-Verlag, 1992.

[Faugeras, 1992] O. D. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In *Computer Vision – Proc. 2nd European Conference on Computer Vision (ECCV'92)*, volume 588 of *Lecture Notes in Computer Science*, pages 563–578. Springer-Verlag, 1992.

[Faugeras, 1993] O. Faugeras. *Three-Dimensional Computer Vision*. MIT Press, 1993.

[Feldmar *et al.*, 1994] J. Feldmar, N. Ayache, and F. Betting. 3D-2D projective registration of free-form curves and surfaces. *research report 2434, INRIA 1994. Available*

via anonymous ftp on zenon.inria.fr, file /pub/rapports/RR-2434.ps and via WWW ftp://zenon.inria.fr/pub/rapports., 1994.

[Feldmar *et al.*, 1997] J. Feldmar, N. Ayache, and F. Betting. 3D-2D projective registration of free-form curves and surfaces. *Computer Vision and Image Understanding*, 65(3):403–424, March 1997.

[Fishler and Bolles, 1981] M. A. Fishler and R. C. Bolles. Random sampling concensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[Fleig *et al.*, 1998] O. J. Fleig, P. J. Edwards, S. Chandra, H. Stuttler, and D. J. Hawkes. Automatic microscope calibration for image guided surgery. In H. U. Lemke, M. W. Vannier, K. Inamura, and A. G. Farman, editors, *Proceedings of the 12th International Symposium and Exhibition on Computer Assisted Radiology And Surgery (CAR '98)*, Excerpta Medica International Congress Series 1165, pages 747–752, 1998.

[Foley *et al.*, 1990] J. Foley, A. van Dam, S. Feiner, and J. Hughs. *Computer Graphics 2nd Edition.* Addison Wesley, 1990.

[Ganapathy, 1984] S. Ganapathy. Decomposition of transformation matrices for robot vision. *Pattern Recognition Letters*, 2:401–412, 1984.

[Gleason *et al.*, 1994] P. L. Gleason, R. Kikinis, D. Altobelli, W. Wells, E. Alexander III, P. Black, and F. Jolesz. Video registration virtual reality for nonlinkage stereotactic surgery. *Stereotactic Functional Neurosurgery*, 63:139–143, 1994.

[Gonzalez and Woods, 1992] R. C. Gonzalez and R. E. Woods. *Digital Image Processing.* Addison-Wesley Publishing Company, 1992.

[Grimson *et al.*, 1995] W. E. L. Grimson, G. J. Ettinger, S. J. White, T. Lozano-Perez, W. M. Wells, and R. Kikinis. Evaluating and validating an automated registration system for enhanced reality visualization in surgery. In N. Ayache, editor, *Computer Vision, Virtual Reality and Robotics in Medical Imaging*, volume 905, pages 3–12. Springer-Verlag, Berlin, 1995.

[Grimson *et al.*, 1996] W. E. L. Grimson, G. J. Ettinger, S. J. White, T. Lozano-Perez, W. M. Wells, and R. Kikinis. An automatic registration method for frameless stereotaxy, image guided surgery, and enhancedreality visualization. *IEEE Transactions on Medical Imaging*, 15(2):129–140, 1996.

[Grimson *et al.*, 1998] E. Grimson, M. Leventon, G. Ettinger, A. Chabrerie, C. F. Ozlen, S. Nakajima, H. Atsumi, R. Kikinis, and P. Black. Clinical experience with a high precision image-guided neurosurgery system. In W. M. Wells, A. Colchester, and S. Delp, editors, *Medical Imaging, Computing and Computer-Assisted Intervention - MICCAI '98*, volume 1496 of *Lecture Notes in Computer Science*, pages 63–73. Springer-Verlag, 1998.

[Hager and Belhumeur, 1998] G. D. Hager and P. N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(10):1025–1039, 1998.

[Haralick *et al.*, 1989] R. B. Haralick, H. Joo, C-N. Lee, X. Zhuang, V. G. Vaidya, and M. B. Kim. Pose estimation from corresponding point data. *IEEE Transactions On Systems, Man, and Cybernetics*, 19(6):1426–1445, 1989.

[Haralick *et al.*, 1994] R. M. Haralick, C-N. Lee, K. Ottenberg, and M. Nolle. Review and analysis of solutions of the three point perspective pose estimation problem. *International Journal of Computer Vision*, 13(3):331–356, 1994.

[Hartley, 1997] R. I. Hartley. Self-calibration of stationary cameras. *International Journal of Computer Vision*, 22(1):5–23, 1997.

[Hata *et al.*, 1996] N. Hata, M. Wells, W. M. Halle, S. Nakajima, P. Viola, R. Kikinis, and F. A. Jolesz. Image guided microscopic surgery system using mutual-information based registration. In K. H. Hone and R. Kikinis, editors, *Visualization in Biomedical Computing*, pages 317–326, 1996.

[Henri *et al.*, 1995] C. J. Henri, A. C. F. Colchester, J. Zhao, D. J. Hawkes, D. L. G. Hill, and R. L. Evans. Registration of 3-D surface data for intra-operative guidance and visualization in frameless stereotactic neurosurgery. In N. Ayache, editor, *Computer Vision, Virtual Reality and Robotics in Medical Imaging*, volume 905, pages 47–56. Springer-Verlag, Berlin, 1995.

[Hill *et al.*, 1994] D. L. G. Hill, C. Studholme, and D. J. Hawkes. Voxel similarity measures for automated image registration. In R. A. Robb, editor, *Visualisation in Biomedical Computing, S.P.I.E. Proc. vol. 2359*, pages 205–216, 1994.

[Hill *et al.*, 1998] D. L. G. Hill, C. R. Maurer Jr., C. Studholme, J. M. Fitzpatrick, and D. J. Hawkes. Correcting scaling errors in tomographic images using a nine degree

of freedom registration algorithm. *Journal of Computer Assisted Tomography*, 22(2):317–323, 1998.

[Ito *et al.*, 1998] K. Ito, K. Takeuchi, and Y. Suzuki. Determining pose of curved 3-d objects based on 2-d contour matching. *IEICE Transactions on Information and Systems*, E81-D(10):1087–1094, October 1998.

[Ivins and Porrill, 1998] J. Ivins and J. Porrill. Constrained active region models for fast tracking in color image sequences. *Computer Vision and Image Understanding*, 72(1):54–71, 1998.

[King *et al.*, 1999] A. P. King, P. J. Edwards, M. P. Pike, D. L. G. Hill, and D. J. Hawkes. An analysis of calibration and registration errors in an augmented reality sttem for microscope-assisted guided interventions. In R. P. Gaston, D. L. G. Hill, and D. J. Hawkes, editors, *Proc. Medical Image Understanding and Analysis*, 1999.

[Koch, 1993] R. Koch. Dynamic 3-d scene analysis through synthesis feedback control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):556–568, 1993.

[Kollnig and Nagel, 1997] H. Kollnig and H. H. Nagel. 3D pose estimation by directly matching polyhedral models to gray value gradients. *International Journal of Computer Vision*, 23(3):283–302, 1997.

[Kutulakos and Seitz, 1998] K. N. Kutulakos and S. M. Seitz. A theory of shape by space carving. Technical report, University of Rochester Computer Science Department, May 1998.

[Kutulakos and Vallino, 1998] K. N. Kutulakos and J. R. Vallino. Calibration-free augmented reality. *IEEE Transactions On Visualization and Computer Graphics*, 4(1):1–20, 1998.

[LaCascia *et al.*, 1998] M. LaCascia, J. Isidoro, and S. Sclaroff. Head tracking via robust registration in texture map images. In *IEEE Computer-Society Conference on Computer Vision and Pattern Recognition*, pages 508–514, Boston, Jun 1998.

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

[Lavallee, 1996] S. Lavallee. Registration for computer integrated surgery: Methodol-

ogy, state of the art. In *Computer-Integrated Surgery, Technology and Clinical Applications*, pages 77–97. MIT Press, Cambridge, MA, 1996.

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

[Leventon *et al.*, 1997] M. E. Leventon, W. M. Wells III, and W. E. L. Grimson. Multiple view 2D-3D mutual information registration. In *Image Understanding Workshop*, 1997.

[Li *et al.*, 1994] H. Li, A. Lundmark, and R. Forchheimer. Image sequence coding at very low bitrates: A review. *IEEE Transactions on Image Processing*, 3(5):589–609, 1994.

[Little and Hawkes, 1997] J. A. Little and D. J. Hawkes. The registration of multiple medical images aquired from a single subject: why, how, what next? *Statistical Methods in Medical Research*, 6:239–265, 1997.

[Liu *et al.*, 1990] Y. Liu, T. S. Huang, and O.D. Faugeras. Determination of camera location from 2-D to 3-D line and point correspondances. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 12(1):28–37, 1990.

[Lorensen and Cline, 1987] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface reconstruction algorithm. *Computer Graphics*, 21(3):163–169, July 1987.

[Lowe, 1987] D. Lowe. Three-dimensional object recognition from single two dimensional images. *Artificial Intelligence*, 31(3):355–395, 1987.

[Lowe, 1991] D.G. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):441–450, 1991.

[Lowe, 1992] D. G. Lowe. Robust model-based motion tracking through the integration of search and estimation. *International Journal of Computer Vision*, 8(2):113–122, 1992.

[Maciunas, 1993] R. J. Maciunas, editor. *Interactive image-guided neurosurgery.* American Association of neurological surgeons, 1993.

[Madsen, 1997] C. B. Madsen. A comparative study of the robustness of two pose estimation algorithms. *Machine Vision and Applications*, 9:291–303, 1997.

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

[Maes, 1998] F. Maes. *Segmentation and Registration of Multimodal Medical Images: From theory, implementation and validation to a useful clinical tool in practice.* PhD thesis, Katholieke Universiteit Leuven, 1998.

[Maintz, 1996] J. B. A. Maintz. *Retrospective registration of tomographic brain images.* PhD Thesis, University of Utrecht, The Netherlands, 1996.

[Maurer Jr. and Fitzpatrick, 1993] C. R. Maurer Jr. and J. M. Fitzpatrick. A review of medical image registration. In J. R. Maciunas, editor, *Interactive image-guided neurosurgery*, pages 17–44. American Association of neurological surgeons, 1993.

[Maurer Jr. *et al.*, 1996] C. R. Maurer Jr., G. B. Aboutanos, B. M. Dawant, R. J. Maciunas, and J. M. Fitzpatrick. Registration of 3-d images using weighted geometrical features. *IEEE. Transactions On Medical Imaging*, 15(6):836–849, 1996.

[Maurer Jr. *et al.*, 1998] C. R. Maurer Jr., R. J. Maciunas, and J. M. Fitzpatrick. Registration of head ct images to physical space using a weighted combination of points and surfaces. *IEEE Transactions on Medical Imaging*, 17(5):753–761, October 1998.

[McKenna and Gong, 1998] S. J. McKenna and S. Gong. Real-time face pose estimation. *Real-Time Imaging*, 4:333–347, 1998.

[Mellor, 1995] J. P. Mellor. Realtime camera calibration for enhanced reality visualization. In N. Ayache, editor, *Proceedings of the First International Conference on Computer Vision, Virtual Reality and Robotics in Medicine, Nice, France*, pages 471–475. Springer-Verlag, Berlin, 1995.

[Murase and Nayar, 1995a] H. Murase and S. K. Nayar. Three-dimensional object recognition from appearance - parametric eigenspace method. *Systems and Computers in Japan*, 26(8):45–53, 1995.

[Murase and Nayar, 1995b] H. Murase and S. K. Nayar. Visual learning and recognition of 3-d objects from appearance. *International Journal of Computer Vision*, 14:5–24, 1995.

[Murphy, 1997] M. J. Murphy. An automatic six-degree-of-freedom image registration algorithm for image-guided frameless stereotaxic rediosurgery. *Medical Physics*, 24(6):857–866, June 1997.

[Nakajima *et al.*, 1997] S. Nakajima, H. Atsumi, R. Kikinis, T. M. Moriarty, D. Metcalf, F. A. Jolesc, and P. Black. Use of cortical surface vessel registration for image-guided neurosurgery. *Neurosurgery*, 40(6):1201–1210, 1997.

[Nayar and Bolle, 1996] S. K. Nayar and R. M. Bolle. Reflectance based object recognition. *International Journal of Computer Vision*, 17(3):219–240, 1996.

[Nayar *et al.*, 1991] S. K. Nayar, K. Ikeuchi, and T. Kanade. Surface reflection: Physical and geometrical perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7):611–634, 1991.

[Nomura *et al.*, 1992] Y. Nomura, M. Sagara, N. Hiroshi, and A. Ide. Simple calibration algorithm for high-distortion-lens camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(11):1095–1099, November 1992.

[Oberkampf *et al.*, 1996] D. Oberkampf, D. F. DeMenthon, and L. S. Davis. Iterative pose estimation using coplanar feature points. *Computer Vision And Image Understanding*, 63(3):495–511, 1996.

[Ousterhout, 1996] J. K. Ousterhout. *Tcl and the Tk Toolkit*. Addison-Wesley, 1996.

[Penna, 1991] M. A. Penna. Camera calibration: A quick and easy way to determine the scale factor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(12):1240–1245, December 1991.

[Penney *et al.*, 1998] G. P. Penney, J. Weese, J. A. Little, P. Desmedt, D. L. G. Hill, and D. J Hawkes. A comparison of similarity measures for use in 2D-3D medical image registration. *IEEE Transactions on Medical Imaging*, 17(4):586–595, 1998.

[Penney, 1999] G. P. Penney. *Registration of Tomographic Images to X-ray Projections for use in Image Guided Interventions - in preparation*. PhD thesis, University Of London, 1999.

[Phong *et al.*, 1995] T.Q. Phong, R. Horaud, and P. D. Tao. Object pose from 2D to 3D point and line correspondances. *International Journal Of Computer Vision*, 15:225–243, 1995.

[Pilu and Lorusso, 1997] M. Pilu and A. Lorusso. Uncalibrated stereo correspondance

by singular value decomposition. In A. F. Clark, editor, *Proc. 8th British Machine Vision Conference (BMVC'97)*, volume 2, pages 500–509, 1997.

[Pluim *et al.*, 1999] J. P. W. Pluim, J. B. Maintz, and M. A. Viergever. Mutual information matching and interpolation artefacts. In K. M. Hanson, editor, *Proceedings of SPIE, Medical Imaging 1999: Image Processing*, volume 3661, pages 56–65, 22-25 February, San Diego, California 1999.

[Press *et al.*, 1992] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C. The Art of Scientific Computing.* Cambridge University Press, Cambridge, 1992.

[Reza, 1961] F. M. Reza. *An Introduction To Information Theory.* McGraw Hill, 1961.

[Robert, 1996] L. Robert. Camera calibration without features extraction. *Computer Vision and Image Understanding*, 63(2):314–325, 1996.

[Rosin, 1999] P. L. Rosin. Robust pose estimation. *IEEE Transactions On Systems, Man, and Cybernetics - Part B*, 29(2):297–303, April 1999.

[Rougee *et al.*, 1993] A. Rougee, C. Picard, C. Ponchut, and Yves Trousset. Geometrical calibration of x-ray imaging chains for three-dimensional reconstruction. *Computerized Medical Imaging and Graphics*, 17(4/5):295–300, 1993.

[Schroeder *et al.*, 1992] W. J. Schroeder, J. A. Zarge, and W. E. Lorensen. Decimation of triangle meshes. In *SIGGRAPH 92: Conference Proceedings*, volume 2, pages 65–70, 1992.

[Schroeder *et al.*, 1997] W. Schroeder, Martin K., B. Lorensen, L. Avila, R. Avila, and C. Law. *The Visualization Toolkit An Object-Oriented Approach to 3D Graphics.* Prentice-Hall, ISBN: 0-13-954694-4, 1997.

[Shekhar *et al.*, 1999] C. Shekhar, V. Govindu, and Chellappa. Multisensor image registration by feature concensus. *Pattern Recognition*, 32:39–52, 1999.

[Steinbach *et al.*, 1998] E. Steinbach, P. Eisert, and B. Girod. Motion-based analysis and segmentation of image sequences using 3D scene models. *Signal Processing*, 66(2):233–247, 1998.

[Strat, 1984] T. M. Strat. Recovering the camera parameters from a transformation matrix. In *DARPA Image Understanding Workshop*, pages 264–271, Oct 1984.

[Studholme *et al.*, 1995] C. Studholme, D. L. G. Hill, and D. J. Hawkes. Multi resolution voxel similarity measures for MR-PET registration. In Y. Bizais, C. Barillot, and

R. di Paola, editors, *Information Processing in Medical Imaging*, pages 287–298, Dordrecht, The Netherlands, in press, 1995. Kluwer Academic Publishers.

[Studholme *et al.*, 1999] C. Studholme, D. L. G. Hill, and D. J. Hawkes. An overlap invariant entropy measure of 3D medical image alignment. *Pattern Recognition*, 32(1):71–86, Jan 1999.

[Studholme, 1997] C. Studholme. *Measures Of 3D Medical Image Alignment*. PhD Thesis, University of London, UK, 1997.

[Tomasi and Kanade, 1992] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992.

[Trucco and Verri, 1998] E. Trucco and A. Verri. *Introductory Techniques for 3-D Computer Vision*. Prentice Hall, Inc., Upper Saddle River, New Jersey 07458, 1998.

[Tsai, 1987] R.Y. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal Of Robotics and Automation*, RA-3(4):323–344, 1987.

[Uenohara and Kanade, 1995] M. Uenohara and T. Kanade. Vision based object registration for real time image overlay. *Journal of Computers In Biology and Medicine*, 25(2):249–260, 1995.

[Ullman and Basri, 1991] S. Ullman and R. Basri. Recognition by linear combinations of models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(10):992–1006, October 1991.

[van den Elsen *et al.*, 1993] 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*, pages 26–39, March 1993.

[van den Elsen *et al.*, 1994] P. A. van den Elsen, E-J. D. Pol, T. S. Sumanaweera, P. F. Hemler, S. Napel, and J. R. Adler. Grey value correlation techniques used for automatic matching of CT and MR brain and spine images. In R. A. Robb, editor, *Visualization in Biomedical Computing, S.P.I.E. proceedings vol. 2359*, pages 227–237, 1994.

[Viola and Wells III, 1997] P. Viola and W. M. Wells III. Alignment by maximization of mutual information. *International Journal of Computer Vision*, 24(2):137–154, 1997.

[Viola and Wells, 1995] P. Viola and W. M. Wells. Alignment by maximization of mutual information. In *Proceedings of the 5'th International Conference on Computer Vision*, pages 16–23, 1995.

[Viola, 1995] P. A. Viola. *Alignment By Maximization of Mutual Information.* PhD Thesis, Massechusetts Instituate of Technilogy. A.I. Technical Report No. 1548. Available by anonymous ftp to publications.ai.mit.edu, 1995.

[Wang and Tsai, 1991] L. L. Wang and W-H Tsai. Camera calibration by vanishing lines for 3-d computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):370–376, April 1991.

[Wang *et al.*, 1997] M. Y. Wang, C. R. Maurer Jr., and J. M. Fitzpatrick. Partial volume effect on marker localization in medical density images. In *SPIE*, volume 3034, pages 580–591, 1997.

[Weese *et al.*, 1997a] J. Weese, T. M. Buzug, C. Lorenz, and C. Fassnacht. An approach to 2D/3D registration of a vertebra in 2Dx-ray flouroscopies with 3D ct images. In J. Troccaz, E. Grimson, and R. Mosges, editors, *CVRMed/MRCAS*, volume 1205 of *Lecture Notes in Computer Science*, pages 119–128. Springer-Verlag, 1997.

[Weese *et al.*, 1997b] J. Weese, G. P. Penney, P. Desmedt, T. M. Buzug, D. L. G. Hill, and D. J. Hawkes. Voxel-based 2-d/3-d registration of flouroscopy image and ct scans for image-guided surgery. *IEEE Transactions on Biomedical Engineering*, 1(4):284–293, December 1997.

[Weese *et al.*, 1999] J. Weese, R. Gocke, G. P. Penney, P. Desmedt, T. M. Buzug, and H. Schumann. Fast voxel-based 2D/3D registration using a volume rendering method based on the shear-warp factorization. In K. M. Hanson, editor, *Proceedings of SPIE, Medical Imaging 1999: Image Processing*, volume 3661, pages 802–810, 22-25 February, San Diego, California 1999.

[Weng *et al.*, 1992] J. Weng, P. Cohen, and M. Herniou. Camera calibration with distortion models and accuracy evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(10):965–980, October 1992.

[West *et al.*, 1997] J. West, J. M. Fitzpatrick, M. Y. Wang, B. M. Dawant, C. R. Maurer Jr., R. M. Kessler, R. J. Maciunas, C. Barillot, D. Lemoine, A. Collignon, F. Maes, P. Suetens, D. Vandermeulen, P. A. van den Elsen, S. Napel, T. S. Sumanaweera, B. Harkness, P. F. Hemler, D. L. G. Hill, D. J. Hawkes,

C. Studholme, J. B. A. Maintz, M. A. Viergever, G. Malandain, X. Pennec, M. E. Noz, G. Q. Maguire, M. Pollack, C. A. Pelizzari, R. A. Robb, D. Hanson, and R. P. Woods. Comparison and evaluation of retrospective intermodality brain image registration techniques. *Journal Of Computer Assisted Tomography*, 21(4):554–566, 1997.

[West *et al.*, 1999] J. West, J. M. Fitzpatrick, M. Y. Wang, B. M. Dawant, C. R. Maurer Jr., R. M. Kessler, and R. J. Maciunas. Retrospective intermodality registration techniques for images of the head: Surface-based versus volume-based. *IEEE Transactions on Medical Imaging*, 18(2):144–149, February 1999.

[Wilson, 1994] R. G. Wilson. *Modelling and Calibratio of Automated Zoom Lenses*. PhD thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, January 1994. http://www.cs.cmu.edu/ rgw/TsaiCode.html.

[Wolfe *et al.*, 1991] W. J. Wolfe, D. Mathis, C. W. Sklair, and M. Magee. The perspective view of three points. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 13(1):66–73, 1991.

[Woods *et al.*, 1992] R. P. Woods, S. R. Cherry, and J. C. Mazziotta. Rapid automated algorithm for aligning and reslicing PET images. *Journal of Computer Assisted Tomography*, 16(4):620–633, 1992.

[Woods *et al.*, 1993] 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, 1993.

[Wu *et al.*, 1994] Y. Wu, S. S. Iyengar, R. Jain, and B. Santanu. A new generalized computational framework for finding object orientation using perspective trihedral angle constraint. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(10):961–974, October 1994.

[Yuan, 1989] J. S.-C. Yuan. A general photogrammetric method for determining object position and orientation. *IEEE Transactions On Robotics And Automation*, 5(2):129–142, 1989.

Publications

Articles in Journals

A. P. King, P. J. Edwards, C. R. Maurer Jr., D. A. DeCunha, R. P. Gaston, M. J. Clarkson, D. L. G. Hill, D. J. Hawkes, M. R. Fenlon, A. J. Strong, T. C. S. Cox, and M. J. Gleeson. Stereo augmented reality in the surgical microscope. *accepted for publication in Presence: Teleoperators and Virtual Environments*, 1999.

P. J. Edwards, A. P. King, C. R. Maurer Jr., D. A. DeCunha, D. J. Hawkes, D. L. G. Hill, R. P. Gaston, M. J. Clarkson, M. R. Pike, M. R. Fenlon, S. Chandra, A. J. Strong, C. L. Chandler, and M. J. Gleeson. Design and evaluation of a system for microscope-assisted guided interventions (MAGI). *submitted to IEEE Transactions on Medical Imaging*, 1999.

Articles in Conference Proceedings

M. J. Clarkson, D. Rueckert, D. L. G. Hill, and D. J. Hawkes. A multiple 2D-3D medical image registration algorithm. In *accepted for publication at SPIE 2000*, 2000.

D. Rueckert, M. J. Clarkson, D. L. G. Hill, and D. J. Hawkes. Non-rigid registration using higher-order mutual information. In *accepted for publication at SPIE 2000*, 2000.

M. J. Clarkson, D. Rueckert, A. P. King, P. J. Edwards, D. L. G. Hill, and D. J. Hawkes. Registration of video images to tomographic images by optimising mutual information using texture mapping. In C. Taylor and A. Colchester, editors, *Medical Imaging, Computing and Computer-Assisted Intervention - MICCAI '99*, volume 1679 of *Lecture Notes In Computer Science*, pages 579–588. Springer-Verlag, September 1999.

M. J. Clarkson, D. Rueckert, A. P. King, P. J. Edwards, D. L. G. Hill, and D. J. Hawkes. Using texture mapping to register video images to tomographic images by optimising mutual information. In D. J. Hawkes, D. L. G. Hill, and R. P. Gaston, editors, *Medical Image Understanding and Analysis*, pages 29–32, July 1999.

M. J. Clarkson, D. Rueckert, D. L. G. Hill, and D. J. Hawkes. Registration of multiple video images to pre-operative CT for image guided surgery. In K. M. Hanson, editor, *Proceedings of SPIE, Medical Imaging 1999: Image Processing*, volume 3661, pages 14–23, 22-25 February, San Diego, California 1999.

M. J. Clarkson, D. Rueckert, D. L. G. Hill, and D. J. Hawkes. Registration of multiple video images to pre-operative data for image guided surgery. In E. Berry, D. C. Hogg, K. V. Mardia, M. A. Smith editors *Medical Image Understanding and Analysis*, pages 73–76, 1998.