Integration of Visual and Haptic Feedback for Teleoperation

Richard Lee Thompson
Trinity College

Robotics Research Group
Department of Engineering Science
University of Oxford

Trinity Term, 2001

Richard Lee Thompson
Trinity College

Doctor of Philosophy
Trinity Term, 2001

The Integration of Visual and Haptic Feedback for Teleoperation

Abstract

Teleoperation systems are an important tool for performing tasks which require the sensori-motor coordination of an operator but where it is physically impossible for an operator to undertake such tasks *in situ*. The vast majority of these devices supply the operator with both visual and haptic sensory feedback in order that the operator can perform the task at hand as naturally and fluently as possible and as though physically present at the remote site.

This thesis is concerned with overcoming the sensory limitations imposed by a fixed camera teleoperation system. The principal aim of this work is the extraction and redisplay of visual information to facilitate such a system.

The thesis augments the Oxford teleoperation system with a virtual viewing module, where the operator is able to select his or her viewpoint and viewing direction onto the workcell by first tracking the locations of known objects in the workcell using a computer vision system, and then rendering them graphically on a display in front of the operator.

This system, because the model-based object tracker is based around a Kalman filter, motivates the design of experiments to examine whether the operator's visuo-motor control loop maintains a state model of the manipulation process as in a Kalman filter. Experimental evidence is presented showing the latter to be false. A new operator model is then postulated using an adaptive gain controller, with the gain chosen to minimise the variance between desired and actual output. The experimental evidence supports this model.

These findings support the hypothesis that the required bandwidth of the tracking filter is both i) sufficient that the tracker can robustly track hand manipulated objects and ii) matched to the visual needs of the operator.

Acknowledgements

First of all, I'd like to thank my supervisors Prof. David Murray and Prof. Ron Daniel for all of their ideas, encouragement and sage advice.

I am extremely grateful to Dr Ross McAree and Dr Alberto Muñoz for sharing their knowledge of the real robotic and programming world with an Engineering graduate fresh from Oxford academia. I would also like to single out thanks to Prof. Andrew Zisserman for his vast knowledge of computer vision and inspiring lectures.

Thanks must go to the people of the Active Vision Laboratory for proofreading, dinners, frisbee and all that: Andy, Lourdes, Ben, Joss, Eric, Torfi and Walterio.

Thanks to college friends: Dougan, Colin, Sean, Adrian, Martin, Tom, Tim, Raegs, Sara and Alastair for the best of times.

I am grateful to EPSRC for funding my research.

Above all, thanks and love always to Mum, Dad, Helen and Max.

Contents

1

Introduction

1.1 Teleoperation

A first look at a modern organisation is enough to convince most observers that we live in a slick, well-automated world where machines are well on the way to replacing people.

On closer inspection, however, a somewhat different world is revealed. Businesses do not produce products for the greater good of mankind automonously and at the drop of a hat. Management structures which need to be in place to cope with the foibles of modern day life take years to fine tune. On closer inspection still, one finds that the most popular management structure is hierarchical, with the lower subordinates receiving, executing and then reporting on small-sized orders directly to their superior. On even further inspection one finds electro-mechanical devices replacing human subordinates *only* at the lower echelons, essentially providing the "grunt" to perform routine tasks. Human control is still and in the foreseeable future necessary at all higher levels. This hierarchy of man and machine is prevalent in many other systems in the modern world and Sheridan [1] coined the term *Human Supervisory Control* (HSC) to encapsulate this idea.

Today's sensor and actuator laden motor vehicles provide a number of examples of HSC, and of what happens when HSC is wrested from the driver. In the area of force-

feedback, power assisted steering and braking are universally accepted, and most drivers are happy to allow ABS and traction control to moderate their actions where necessary. Entirely acceptable too are devices which supply information, from the humble oil-pressure gauge, to route-planning devices which suggest optimal routes. But acceptability of devices going further than this is much less. Deciding when to use windscreen wipers is a example of rather menial human supervisory control, but having the process automated (as one manufacturer boasts at present) would seem highly irritating. Not only is Human Supervisory Control necessary in many cases, but we also enjoy it. Contrasting the ABS and wiper examples suggests that if the driver has the sensory information to make a decision, then he or she is happiest exercising it.

This thesis is about what sensory information should be supplied to the human controller of a device being operated at a distance. Teleoperation, the direct and continuous control of mechanical systems to extend a person's sensing and manipulation to a remote site, is a subset of human supervisory control which is an active area of research in its own right.

Goertz invented the modern master-slave teleoperation system in 1945 [2]. His device comprised two mechanical mechanisms connected by wires and tape so that when an operator moved the master mechanism the slave mechanism copied the action. This system was used for the safe manipulation of radioactive materials in a contaminated area. It was soon afterwards that electro-servomechanisms replaced the wires and tapes, and nowadays the master and slave mechanisms are independent robots and their kinematics are usually quite different. The Oxford teleoperation system, like many, uses a bilateral Stewart platform for the master and an articulated PUMA 560 robot arm for the slave.

An early problem faced by the pioneers of teleoperation was to understand which of the five senses should be sufficiently stimulated so that the operator was aware of the remote environment. Goertz's first teleoperation system, for example, allowed the operator directly to view the remote site hence providing visual feedback. Very soon afterwards force feedback was conveyed to the operator by allowing the slave to back-drive the master robot.

Modern day force-reflecting master-slave systems increase the reflected force fidelity by using a force/torque sensor mounted on the slave's wrist. High fidelity force feedback

is often referred to as haptic feedback relating the notion that it is the operator's sense of touch or tactile receptors which are stimulated. These modern systems have found a use in many hazardous environments. Space teleoperation, for example, holds the promise of assembly, servicing and repair with the minimal number of expensive manned missions. Hirzinger's [3] (1993) ROTEX project allows the earth based operator to control the remote slave, which is situated on a space shuttle orbiting the earth, through a six degree-of-freedom control ball. Undersea teleoperation using remotely operated vehicles (ROVs) are popular with the offshore oil and gas industry. Their tasks include the monitoring of pipelines and the inspection of sub-sea structures (Vadus [4], 1976). Their popularity has been driven by the high cost of human divers ($£5000$ per working hour at depths of 300m, due to support facilities) not to mention the significant risk to life. Although the risks to human life are relatively high for these two applications of teleoperation, perhaps it is with nuclear teleoperation, in which the long term risk to human life is just too great, that force-reflecting master-slave systems have become most associated. Teleoperation systems for nuclear processing work serve two basic functions. The first function is to perform routine maintenance within the cell and the second function is the decommissioning of the cell after it has served its useful life. Teleoperation is also used in toxic waste and mine clearance, where the risks of chemical toxicity and bomb detonation are again too great for human workers, even with protective clothing (Sheridan [1]).

But the use of teleoperation is not just restricted to hazardous environments. Many large and small scale tasks need to be performed which require the use of mechanical devices and the sensori-motor coordination of the human operator. In construction, for example, cranes and diggers are akin to a teleoperator, extending the operator's control and sensing to the lifting of loads of many tons. Moving from macro to micro, teleoperators are used to facilitate microsurgery. Moreover the use of teleoperators within medicine has involved a long and illustrious partnership. The first medical teleoperator was the endoscope, a fiber optic bundle with tubes for conveying fluids to or from the distal end point, wires for modifying its curvature so it can be maneuvered around corners and an end-point device for gripping which allows the operator to inspect, collect samples and perform surgery on a very small scale. 6-DOF vitroretinal teleoperators with bilateral force feedback and adjustable gain from master to slave down to a resolution of a few microns

are now commonplace for eye surgery (Sheridan [1]). More recently medical operators have been offered enhanced visual feedback with the novel use of imaging scanners such as an MRI scanner and the application of virtual reality. The medical vision laboratory at MIT (Grimson, *et al.* [5]) and other labs have developed visual tools to allow the surgeon to view a registered 3D reconstruction of internal anatomy obtained from MRI scans of the patient for preoperative surgical planning and intraoperative surgical guidance. They refer to the feedback displays as an *enhanced* reality visualisation tool.

1.2 Fundamental Challenges

Just as one finds with our introductory examples of the power assisted steering and anti-lock braking systems, teleoperators must, if they are to enable the operator to perform non-routine tasks for large amounts of time, be as transparent and as natural to use as possible. But are these systems physically realizable?

1.2.1 Physical Limitations

In a perfect world the force measured using a wrist-force/torque sensor located at the slave's end-effector should drive the motors of the master arm which would then push back on the operator's hand with the same force as the slave pushes against the environment. Unfortunately this system requires that the master and slave impose no mass, compliance, viscosity or static friction characteristics of their own. A very tall order! To overcome these restrictions the reflected force is considerably attenuated before transmission to the operator. This attenuation is usually set by trial and error to a value just below that which would make the master-slave system marginally stable. Kim [6], for example, reports (using a moderately lightweight master input device and a PUMA 560 slave manipulator) that no more than 10% of the force at the slave-environment interface can be reflected to the operator. This observation is common. At Oxford, Daniel and McAree [7] made use of a notch filter, detailed in chapter 3, to increase the fidelity of the reflected force signal.

Large transmission delays between slave and master can also cause instability. A condition for stability is that the plot of the open-loop phasor with increasing frequency must never circle the (-1) point on the complex plane (the Nyquist criterion). A pure time de-

lay, without attenuation, is represented by the transfer function $\exp -sT$ where T is the delay time. This factor adds a phase advance to the frequency response without altering the magnitude of the curve thus decreasing the phase margin of the system.

The first reported case of the effects of time delay was that of the early lunar teleoperator, Surveyor, which was controlled from an earth ground station with round trip delays of some 4s. To make the system stable the operator had to execute a sequence of move and then wait commands with tasks taking minutes rather than seconds to complete causing severe operator fatigue. Low-orbit and deep ocean teleoperators are also unstable even with a much smaller round-trip delay of around 0.3s. Deep ocean teleoperation unfortunately has to use acoustic telemetry which is subject to around one second delay for every kilometer covered.

By closing prompt control loops (loops that are not subject to time-delays) at either the master or slave environments the debilitating effects of time-delayed teleoperation can be removed. Noyes [8] first closed the master control loop with the use of a *predictor display* where performance times improved by some 50%. Here the operator views a locally prompt simulation of the remote site viewing either artificially generated graphics over the time-delayed video feed or a virtual reality display. The simulation is typically achieved by extrapolation using a model of the remote cell and the operator's current commands via a Taylor expansion. A typical framework makes use of a Kalman filter (Hirzinger *et al.* [9]).

Perhaps the most impressive example of a space teleoperation system where the slave's control loop is closed is with the robot navigator "Sojourner" [10] during NASA's Pathfinder Mission to Mars in 1997. Operators based on earth used telemetry information sent by the lander to gain a realistic impression of the terrain surrounding the robot. A course was then plotted and was sent to the robot for the robot to complete during that day using its own sensors and actuators.

But instability is not the only cause of debilitation in teleoperation.

1.2.2 Sensory Transfer Limitations

Problems can occur in teleoperation when the operator receives either insufficient or conflicting sensory feedback. Several studies carried out at MIT, and summarized in Sheridan [1], have explored the interaction between the quality of video information and op-

erator performance. Ranadive [11] and Deghuee [12] considered how teleoperator performance varies with changing frame rate, resolution and gray scale, the product of the three determining the bit rate which has as its ceiling the bit rate bandwidth of the link. By systematically varying each parameter, Ranadive found an almost direct correlation between operator performance and the bit rate of the image data displayed over the range tested, concluding that the three parameters weigh equally. Operator performance is little improved if the frame rate is increased above 15 Hz, but if the rate is taken below some 5.6 Hz teleoperation is all but impossible.

Operators can receive insufficient visual information even from fixed vision systems which are not subject to bandwidth restrictions. Objects may be obscured or the viewing angle and direction may be poor. Virtual predictive displays as used in time-delayed teleoperation provide a means to overcome these difficulties. However they have two defects — a dependence on prior geometric knowledge, and a lack of visual realism. Barth *et al.* [13] have implemented a predictive teleoperation system in which the model of the environment is photo-realistic. This photo-realism is achieved by using computer graphics techniques to map the original textures of the workcell objects onto their virtual counterparts using a stereo rig with known kinematics. But the virtual scene is only geometrically realistic for views centered around the original view pair.

Ensuring sufficient force fidelity to allow the operator to deconvolve and thereby interpret slave contact, is also important. Unfortunately to stabilize teleoperation systems, the magnitude of the reflected force must be attenuated significantly. As a consequence most force-reflecting systems have a characteristic "spongy" or "mushy" sense of feel that most operators find unsatisfactory [14]. Daniel and McAree [7] note that only by the proper stimulation of the hand and arm receptors, subject to the constraints imposed by stability considerations, will these sensory deprivation problems be overcome. Their work is described in more detail in chapter 3.

Unfortunately supplying sufficient visual and haptic information to the operator in itself is not enough. Problems can occur when force and visual information conflict. A theory of visual dominance was put forward by Posner, Nissen and Klein, who had explored the effects of poor visual information [15]. Their observations suggested that because vision is intrinsically less good as an event alarm (eg. the sound of impact is more alerting than

the sight of impact), the human perceptual system *raises* the attentional awareness given to vision. They used their findings to explain a bias towards vision and away from kinesthetic cues found earlier by Jordan [16]. They argued further that an operator can have a *lower* performance when visual data is added to another sensor modality than that achievable with that modality alone.

When combining force and visual feedback in a teleoperation system it is essential that: i) reflected force be of sufficient fidelity to allow the operator to deconvolve and thereby interpret slave contact; ii) the slave motion observed by the operator be well correlated to his own motion and iii) what the operator sees is consistent with the force he feels. Sheridan [1] coined the term *telepropriоception* to describe these basic requirements to overcome force and visual sensory conflict.

1.3 Thesis Objectives

The aims of the work conducted in this thesis are to explore, improve and in some way optimise the extraction of information from fixed cameras to facilitate teleoperation.

In the previous section it was noted that fixed cameras are restrictive in the context of teleoperation because they supply the operator with suboptimal visual information. A better viewing system has been demonstrated in the work on predictive displays where the operator views a virtual reality model of the workcell environment, with the operator able to control the viewpoint and viewing direction. By necessity, due to the instability caused by time-delay, predictive displays use a local simulation of the remote site to predict the positions of the virtual objects. But static biases may be introduced through incorrect prior geometric information, with damaging effects on operator performance. However our concern here is not with time-delayed teleoperators[1] but with prompt teleoperators, where the video stream offers prompt recoverable geometrical information. The first practical objective of this thesis is therefore

- to develop a visual pose[2] recovery system which can recover the pose of workcell objects from multiple cameras arranged around the remote workcell.

[1] in this thesis a teleoperator refers to the master and slave robots and not to the human operator
[2] position and orientation

Two important choices in the design of the visual pose recovery system were, first, whether to use data-driven structural recovery methods for polyhedra such as used in the system developed by Barth *et al.* [13] or to rely on model-based, or prior-based methods [17–24]; and second, whether to use calibrated vision to recover Euclidean structure or un- or partially-calibrated vision to recover structure modulo an unknown transformation. For the latter choice, whilst there are certainly tasks in hand-eye coordination for which uncalibrated vision is quite adequate [25,26], the need to function in a Euclidean robot work space and the a priori undefined nature of the operator's tasks suggest that the calibrated route might be more embracing, though more inflexible. For the former choice, the needs for robustness, geometric realism and real-time performance on modest hardware still point to a purely model-based approach.

Providing the operator with a geometrically realistic picture of the remote environment is all well and good, but have we satisfied the informational needs of the operator? The tracker, for instance, might introduce latency or might not be able to track at frame-rate. Hence the second objective of this thesis is to

- explore the visual informational needs of an operator performing a manipulation task, to improve the design of a pose recovery system.

A chapter-by-chapter layout of this thesis is as follows:

- Chapter 2 reviews the pose recovery and operator modeling literature. This helps to place the contributions of this thesis in context.

- Chapter 3 begins with an overview of teleoperation equipment and software architecture used for the work described in this thesis. The chapter continues by describing the design and build issues involved in augmenting the haptic system with a *virtual viewing system* so that the operator can perform manipulation type tasks optimally.

- Chapter 4 begins the description of the visual tools used to measure the pose of the workcell objects. In particular it is concerned with the calibration of the intrinsic and extrinsic parameters of the cameras viewing the slave robot workspace. A novel aspect of the algorithm proposed is that we use the robot to generate a calibration

pattern, without assuming that the robot can generate perfectly accurate world positions. The principal aim of this chapter is to devise a model that respects the physical attributes of the robot arm. Results are shown which validate the model.

- Chapter 5 describes the real-time pose recovery system first introduced by Harris [20] in his RAPiD tracker. This has been extended to multiple calibrated cameras and its performance improved using robust methods.

- Chapter 6 uses the system designed in this thesis to explore some characteristics of human manipulation. The purpose of this chapter is to aid in the understanding of the sensors needed to perform a dynamic manipulation task and to propose an information processing/representation model of human manipulation. Our findings falsify a commonly held model of human manipulation.

- Chapter 7 concludes the thesis with a general discussion and ideas for future work.

1.4 A different viewpoint on the objectives

The two objectives set out above can be viewed in a different way. We can view our system as a lossy data compressor able to represent the geometric information contained within the video streams compactly. The compressed geometry is then decompressed, and displayed to an operator who wishes to perform a manipulation type task. Our first objective is to design of the compressor — the pose recovery algorithm. Our second objective is to explore the visual informational needs of an operator, ensuring that this system is indeed optimal for manipulation.

2

Literature Review

This chapter reviews literature relevant to the two main objectives of this thesis:

- the design of a pose recovery system and
- the exploration of the visual informational needs of an operator performing force reflected tele-manipulation.

2.1 Visual Pose Tracking

As stated in the introductory chapter, the first practical objective of the thesis is to develop a visual pose recovery system for tele-manipulation. The problem of pose recovery may be defined as follows:

Given an object whose Euclidean structure (model) is known and the feature correspondences between a single calibrated perspective image of the object and its model.

find the pose (position and orientation) of the object relative to the camera.

The pose recovery problem is illustrated in Figure 2.1.

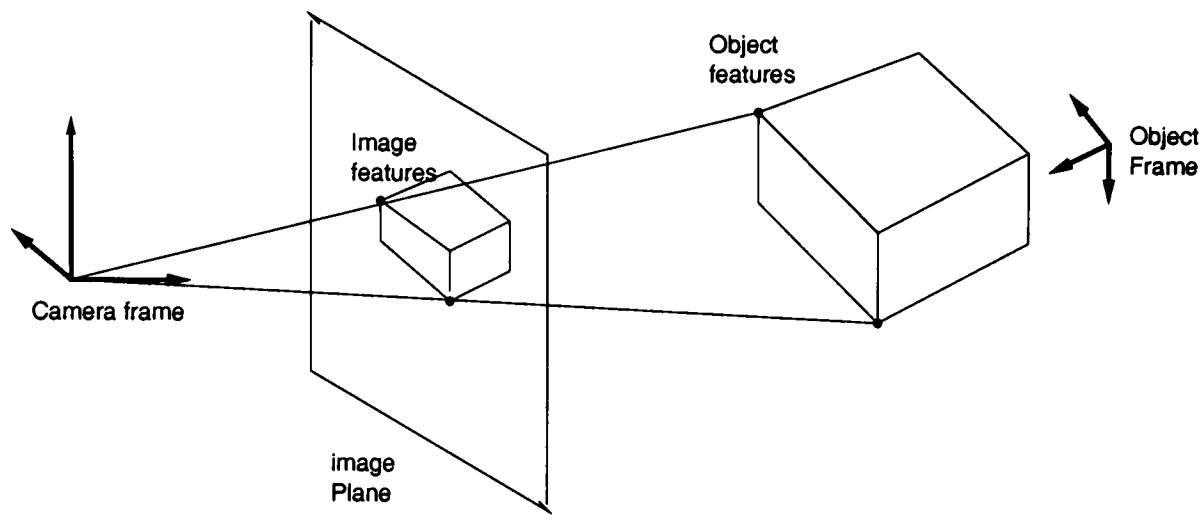


Figure 2.1: In pose recovery the image positions of the image features are used with the known 3D positions of the object features expressed in the object frame to deduce the pose of the object in the camera frame. This is the rotation and translation that maps points in the object frame to the camera frame.

The various pose recovery algorithms can be classified into either analytic or iterative methods. Although analytic or closed form methods have a number of inherent disadvantages, which we discuss, they are usually included in pose recognition algorithms. We review several seminal pose recognition papers because of their ingenious techniques to obtain pose robustness.

From a computational standpoint the iterative methods are more relevant to this thesis and are discussed in section 2.1.2.

2.1.1 Pose Recognition

Pose recognition methods aim to identify objects within images by relying on the fact that almost all the images of an object are unique to that object. A large proportion of these algorithms are split into two phases of *hypothesize* and then *test* and can be partitioned into either *pose alignment/consistency* or alternatively *pose clustering* methodologies. All these algorithms rely on analytic pose recovery methods and this will be reviewed first.

Papers detailing the analytic, or closed form pose recovery methods — also called the PnP or space resectioning problem — are numerous. Authors of note are Haralick *et al.*, Fischler and Bolles, Förstner, Horaud *et al.* [27–30] to name but a very select few. A rigid body can move in Euclidean space with six degrees-of-freedom. Each correspondence between an image and an object point (Figure 2.1) under non-degenerate conditions provides

two constraints on the motion of the object thus requiring a minimum of three points to constrain and establish the pose of the object. Horaud *et al.* [30] reviewed methods for analytic pose recovery from a variety of feature correspondences including three points, six points and three lines. More recently Quan and Lan [31] proposed a linear method of pose recovery for an arbitrary number of point correspondences. Their use in this thesis is limited by two disadvantages

- There is no clean provision for weighting the data by their positional uncertainty and estimating the resulting uncertainty in the final pose. The usual method is by perturbing the data and observing the effect in a manner similar to a Monte-Carlo approach.

- Selective constraints are used; some authors, for example, do not enforce that the recovered rotation matrix is orthogonal and whose determinant is $+1$ ie. the recovered matrix belongs to the proper orthogonal group of $\mathbb{R}^3$, SO(3). Also, the ordering of the image points changes the estimate of the pose. Haralick *et al.* [27] showed that merely changing the ordering of the features could cause the relative pose error to change by a factor of 10^4.

Pose recognition methods use analytic pose recovery methods because they only require the minimal number of feature correspondences. Perhaps the most famous pose recognition technique was proposed by Fischler and Bolles [28] in their seminal paper detailing RANSAC. This approach is also commonly referred to *pose alignment* [32]. The method of RANSAC begins with an *hypothesis* stage where the minimum number of image features needed to obtain a pose estimate are selected at random and assigned to arbitrary object correspondences. The pose is then calculated using these image and object correspondences and is used to project object features into the image. A *test* stage scores the validity of the pose estimate by examining how close the reprojected object features are to the image features. The cycle is repeated a number of times and the pose value with the highest score is chosen as the object's pose.

Another popular pose recognition technique is that of *Pose clustering* some times called the Hough transform technique. Again a putative pose is obtained by randomly picking image and object correspondences. A large number of putative poses are clustered together

and the cluster with the largest number of poses is taken as the consensus pose. A possible mechanism for clustering is to partition the pose space into bins with the bins clustering poses of similar values together.

2.1.2 Iterative Pose Recovery

Generally the equations relating to the pose of an object and the image positions of its features are non-linear due to rotation and perspective projection. Although a rotation can be represented by a rotation matrix, which is hence linear, ensuring orthonormality forms nonlinear constraint equations; the other parameterisations of rotations such as Euler angles, angle axis or unit quaternion representation also introduce the nonlinearity. Early iterative methods such as those of Harris [20] linearised these equations about an estimate of the true pose and solved the resulting system of linear equations for an adjustment to the estimate. This process is repeated until the adjustment becomes sufficiently small. More recently Dementhon and Davies [33] proposed a method which initially linearises around an estimate of a weak or orthographic camera and convergences to a perspective camera.

Although the methods of iterative pose recovery lack some of the elegance of the analytic solutions described above, the iterative approach has many advantages:

- The use of the linear framework allows a least-squares solution to be found for any number of features.
- The least-squares framework also makes possible the weighting of the image features by their uncertainty and the propagation of this uncertainty onto the final pose.

Iterative Pose Recovery : Pose refinement

Harris' [20] RAPiD tracker represents a 3D object as a set of control points which in an image lie on high contrast edges. These model points are simple to project into the image and the corresponding image edges simple to locate by searching perpendicular to the edge direction using standard edgel image processing such as a 1D Canny edgel detection (Figure 2.2). This model-based tracker makes use of a Kalman filter and the tracking cycle follows that of the filter, which is prediction, projection followed by the update of the state. The filter state consist of the object pose, which is representated as a 3-vector translation

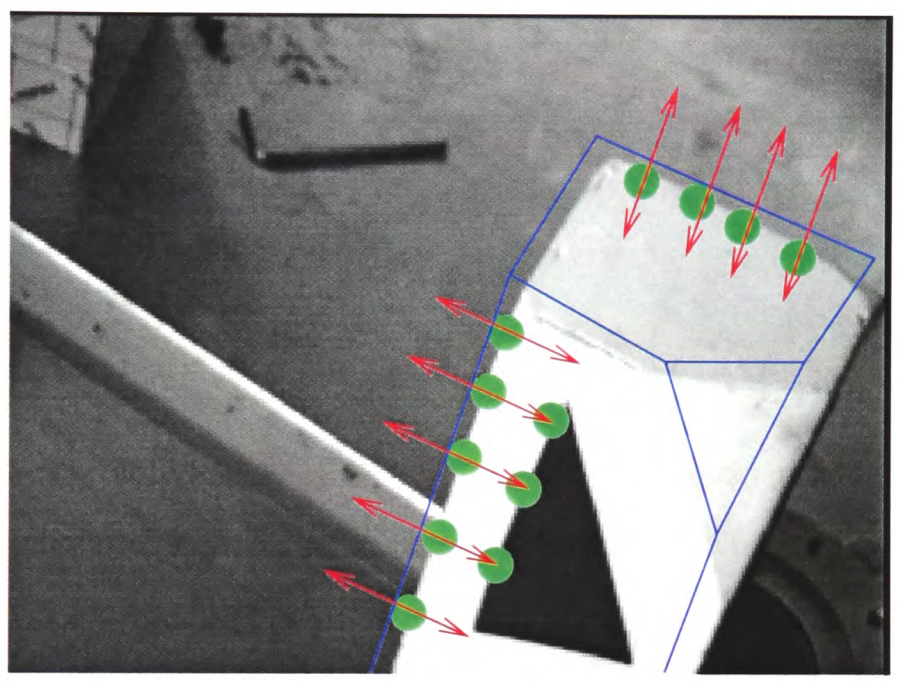
and 3-vector angle axis orientation, and object pose velocity represented as linear velocity and angular velocity (both vectors of 3 dimensions). The object's motion is assumed to evolve by a Wiener velocity process. Features adorning the model object are projected into the images and a 1D image search is then initiated to obtain the measurement innovations. Object features are control points lying along object edges and the measurement information is the perpendicular distance of the control point from its corresponding edge. Control points were chosen as object features instead of object edges because edges can break due to occlusion and their incomplete and variable termination at junctions means that only perpendicular information is of reliable use.

The Kalman filter prediction is useful when the deterministic inter-image object motions are large. The image feature 1D search regions will now be proportional in size to the projection of the stochastic element of the object motion. Providing this stochastic element is small then mismatches will not occur. When mismatches occur, however, as they will when there is substantial background clutter and/or high stochastic object motions, then other methods must be used to associate the measurement data with their true corresponding object features. Some of the other papers on pose recovery have explored this issue and will be covered in due course.

The information filter is an alternative reformulation of the Kalman filter which offers computational savings for visual pose recovery. Gennery [34] made use of such a filter within his tracking algorithm because of the advantages offered for a large measurement vector compared to the small object state vector. Gennery's method also differed from Harris' in that his image features were edges provided by a dedicated hardware edge detector (IMPEX), which preprocessed the entire image feed by applying a Sobel operator with thresholding and thinning at video rate. Gennery's tracker formed part of an object grasping system, designed for NASA, which required both pose and pose velocity measurements.

More recently Heuring and Murray [35] have implemented a tracking system similar to Harris' which makes use of an Ornstein-Uhlenbeck velocity process rather than the usual Wiener process for tracking head motions. This stochastic motion model is a better description for head dynamics. In additional to a more faithful motion model a RANSAC algorithm was used to enforce collinearity of the edge control points to remove outliers.

Armstrong and Zisserman [36] extended the layering of robust methods within a visual



1. Predict the pose of the object using the object motion model and project the object's control points into the image.
2. For each control point find the image location by searching perpendicular to the projected model edge and searching for the nearest image edgel.
3. Calculate pose correction $\Delta \mathbf{p}$.
4. Update the pose of the object using Kalman filter update.

Figure 2.2: Object features lie along high contrast image edges. The 1D search lie perpendicular to the image edge direction.

pose tracker to include an object line case deletion step (using case deletion diagnostics) at the pose level stage of their pose recovery algorithm. Although they implemented Harris' method they have not made use of a filter. While there is no question that the extra layer does offer more robustness, the paper does not make it clear what levels of background clutter and object agility the method can cope with. They extend Harris' perspective camera pose tracker to track objects using different camera models: parameters other than pose are thus recovered such as the affine camera parameters. An interesting point of note is that they found that when recovering both pose *and* the focal length of a perspective camera for imaging conditions where the perspective effects are small then the focal length and the depth of the object are highly correlated and incorrect — a disadvantage arising from not using priors.

Many other authors have chosen not to use filters. Lowe [37] for example simultaneously solves the correspondence and measurement problem, via a combination of a Levenberg-Marquardt iterative minimisation and a Bayesian decision theory approach using edges as image features which are detected by a Marr-Hildreth edge detector.

Lowe's decision function makes it possible to reason about which features to search for depending on the confidence that the feature will not be mismatched. These initial features are matched and used to calculate a better pose estimate. This estimate is used to match more ambiguous image features which in turn is repeated. It is hoped that the initial set of features are true correspondences, otherwise backtracking occurs and the process is repeated with other matches. A key part of the algorithm is the computation of the Bayesian decision value for each feature. Matched model image features will have a tight Gaussian distribution however mismatched features will have been triggered by clutter which has a uniform density dependent on the novelty of the image feature — edges being less novel than say an image patch of the corner of a high contrast labeled object feature.

The approaches detailed above rely on removing outlying features which violate the Gaussian measurement model assumption. An alternative approach is to use a better measurement model. M-estimators (also referred to as robust least-squares estimators) can cope with additional measurement models [38] where the influence of outlying data on the pose estimate is reduced rather than discarded. M-estimators unfortunately do not admit closed form solutions and progress iteratively.

Marchaud *et al.* [39] used M-estimators within their two phase 2D/3D model based tracker algorithm. In the first phase the 2D affine object image motion is estimated using an M-estimator. The second phase uses a *block matching algorithm* similar in principle to the Levenberg-Marquardt algorithm using a steadily decreasing step size to recover 3D pose.

Drummond and Cipolla [40] tracked the pose of an object using an eye-in-hand robotic system for the purpose of pose-based visual servoing using an M-estimator approach. Their robust least-squares pose recovery algorithm calculates the object pose differential from the image feature innovations using an M-estimator. A novelty of their work is the use of Lie algebra to formulate the linear image Jacobian.

Robustness of pose can be ensured by other methods. Li and Wang [41] make use of the projective cross-ratio which is preserved under a projective transformation to reject outlying feature measurements. The cross ratio for four collinear image points a, b, c and d (Fig. 2.3(a)) is defined as

$$\times(a,b,c,d) = \frac{|ac|}{|bc|} / \frac{|ad|}{|bd|}$$

and this is equal to the cross ratio of the four corresponding scene points. Li and Wang made use of a cross-ratio to define a vertex in terms of the angles between four other coplanar vertices Fig. 2.3(b). The angles α, β, γ are the angles between the lines defined by the vertices AB to AC, AC to AD and AD to AE respectively. Assume that there exists a line in the plane, which intersects the lines AB, AC, AD and AE at four specific points B', C', D' and E'. The cross-ratio of A is defined as

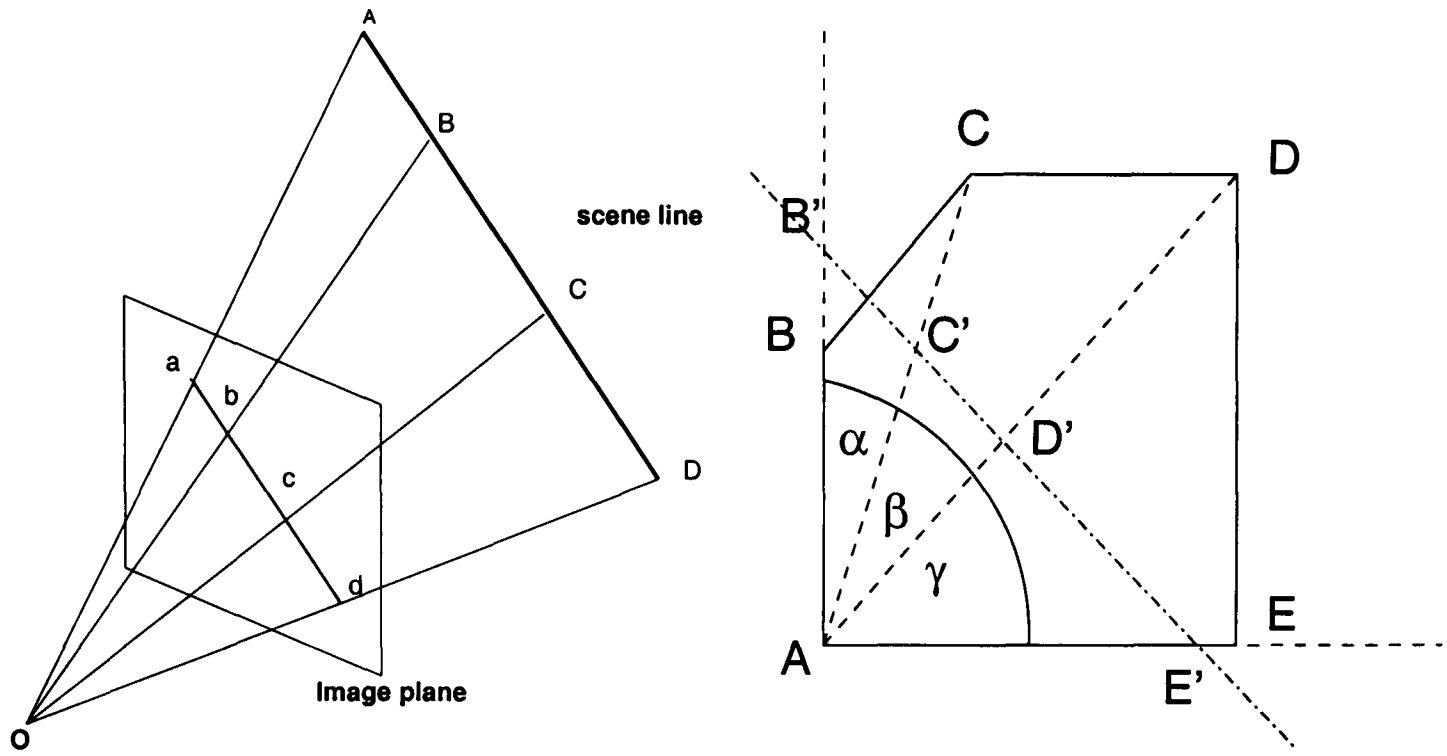
$$\times_A(B',C',D',E') = \frac{|B'D'|}{|C'D'|} / \frac{|B'E'|}{|C'E'|}$$

Let h be the distance from A perpendicular to the straight line. By using the areas of corresponding triangles, the cross-ratio of these four points is

$$\times_A(B',C',D',E') = \frac{h/2|B'D'|}{h/2|C'D'|} / \frac{h/2|B'E'|}{h/2|C'E'|}$$

and so we have

$$\times_A(\alpha,\beta,\gamma) = \frac{\sin(\alpha+\beta)\sin(\beta+\gamma)}{\sin(\beta)\sin(\alpha+\beta+\gamma)}$$



(a) (b)

Figure 2.3: (a) The cross-ratio of four collinear points. (b) The cross-ratio can be used to define a projective invariant for a point **A** with four emanating lines.

They also propose a new edge detector utilising local minimum energy, defined as the negative absolute value of the edge strength which is more robust to changes in lighting conditions.

$$e = -|\nabla(\mathbf{G}_\sigma * \mathbf{I}(w))|^2 \quad ,$$

where $\mathbf{G}_\sigma$ is a 1D Gaussian of standard deviation σ and $\mathbf{I}(w)$ is the intensity of pixels along a line segment in the direction orthogonal to the edge.

Iterative Pose Recovery : Camera refinement

Recently some authors have explored an alternative method of iterative pose recovery which does not require pose instantiation. The method introduced by Dementhon and Davis [33] first instantiates the pose linearly using with a weak perspective camera model, which converges after iteration to the pose computed with perspective camera model. Their method combines two algorithms; the first which they call POS (Pose from Orthography and Scaling) which approximates the perspective projection with a scaled orthographic projection and finds the rotation matrix and the translation vector of the object by solving a

linear system; the second algorithm, POSIT (POS with ITerations), uses in its iteration loop the approximate pose found by POS to converge in the limit to a perspective camera pose. A known fault of the algorithm is that for views of objects with large perspective effects (where the object is close to the camera and/or at some distance away from the optical axis) convergence is very slow (100 iterations rather than 5 or 10) or does not in fact converge.

Horaud *et al.* [42] extended the method of Dementhon and Davies by improving the convergence properties and enforcing the orthogonality of rotation matrices. They improved the convergence properties by initialising the second stage of the algorithm with pose linearly recovered from a first order (paraperspective) as opposed to zero order (weak perspective) camera.

While this method of pose recovery is ideal for pose instantiation from an arbitrary number of points, its use for real-time pose tracking is rather dubious. Considerable computational cost is expended in converging the pose from a paraperspective/weak-perspective camera to a projective camera every frame taking a minimum of five iterations, whereas Harris' method, for example, converges after just one iteration.

2.1.3 Tracking with Outliers

The iterative or least squares methods assume that the noise affecting the input points takes on a Gaussian distribution. The most important consequence of using this noise model is that its tails approach zero very quickly. For example, the probability of finding an error of 5σ (around five pixels for image data) occurring is roughly 10^{-7}. Since this is so low, the least-squares estimator moves the model well away from the true data points in an effort to bring outliers into line. The development of methods that are insensitive to outliers are known as robust estimation.

Two approaches have been mentioned in the pose recovery literature to deal with outliers: use a more realistic noise distribution or remove them in a preprocessing step. The first approach is the rationale behind M-estimators and the second behind the RANSAC paradigm. Both approaches assume that the pose of each object is correct and unambiguous at every time step. Recently an approach has been introduced which relaxes this rather restrictive constraint by using factored sampling.

Condensation

Isard and Blake's [43] condensation algorithm has been used to track low-dimensional (with respect to independent parameters) curves in uncorrelated clutter. It uses the same cycle framework as the Kalman filter, however it makes use of factored sampling of the state probability distribution which is represented by a number of discrete particles or samples. This extension enables a multi-modal observational model, which will be obtained when the false image features are detected, to update the state distribution. It is important however to note that the cost of this generality comes at the loss of the computationally tractable least-squares update of measurements and state prior within the Kalman filter. Forsyth and Ponce [44] explain that one can think of a particle filterer as a form of search, where the particles represent a series of estimates which are compared to the data; estimates which look as though they yielded the data are kept and the others are discarded. The difficulty with large dimensional spaces is that good hypotheses may be missed. This may occur as the likelihood function had many narrow peaks. They remark that it is extremely difficult to get good results from particle filers in spaces of dimensions much greater than about 10. Recently Sidenbladh *et al.* [45] have implemented a condensation tracker for a human motion capture system. The tracker can track walking movements however they require 15,000 particles to populate some twenty states, they report computational run times of some seven and a half minutes per frame this includes the use of a very strong walking motion prior. An alternative to using highly populated particle space is to let the search populate the space accordingly, this is done in Deutscher *et al.* [46] where an annealed particle filter essentially searches for the global extremum through a sequence of smoothed versions of the likelihood. However clutter creates peaks that can require an intractable number of particles.

2.1.4 Conclusions

This section began by reviewing the two basic methods of pose recovery; analytic and iterative. Of the two paradigms iterative methods seemed most suitable for pose recovery within tele-manipulation. Of particular note were the fusion of information from an arbitrary number of feature correspondences to form a least-squares pose estimate, which under the assumption of Gaussian noise distributions forms a maximum likelihood esti-

mate. Unfortunately outliers cause the model to move well away from the true data points in an effort to bring outliers into line. It was discussed that there were two methods to deal with outliers: a more reliable noise distribution or remove them in a preprocessing step. The first approach was the rationale behind M-estimators which are used by Drummond and Cipolla [40]. Unfortunately this approach is slow due to the need for multiple iterations. The most successful method for eliminating outliers before processing is the RANSAC approach of Fischler and Bolles [28] and has been used successfully in pose tracking by Armstrong [36] and Heuring [35] at an intermediate, instead of global level, where collinearity is ensured. Both methods make use of the framework set down by Harris [20]. Indeed Harris' method also seems attractive for tracking telemanipulated objects due to its real-time capabilities. Indeed the need to function at frame-rate (Section 2.2.1) and the already proven capabilities of the pose recovery methods which use robust methods firmly steer the choice of methodology away from condensation type methods.

All the pose recovery methods require intrinsic camera calibration for pose recovery relative to the camera frame, and extrinsic camera calibration for pose recovery relative to an arbitrary world frame. A brief review of camera calibration is given in the introduction of chapter 4 before discussing our method of camera calibration which uses the slave manipulator so that the recovered object pose is relative to the slave manipulator base frame.

2.2 Integration of visual and haptic feedback

This section is concerned with a review of the literature which has studied the visual and haptic needs of operators while performing force-reflected teleoperation. Schloerb [47] highlighted the flow of information *from* the human-machine interface *to* the operator as the most important sensor transformation within a master-slave tele-operation system. This transformation for visual force-reflected teleoperation occurs in two fundamentally different ways; the visual modality detects photons on the retina, while the haptic modality samples pressure changes on the skin and the operator's joint, muscle and tendon receptors. Both modalities are concerned with the recovery of object properties – such as judging size, shape or position — and the central nervous system fuses such information using a common representation of visual and haptic information (Ernst and Banks [48]).

The brain recreates 3D visual information from a variety of cues such as stereopsis (binocular disparity), shape-from-shading cues, knowledge of everyday objects with which to establish scale, texture gradients and motion parallax. The visual system combines each of these sources of information to yield an operationally useful 3D structure from everyday scenes. Teleoperation systems may offer suboptimal visual information to the operator and many authors have measured operator performance under varying parameters associated with visual feedback, summarised in Section 2.2.1.

Haptic cues also influence the accuracy of the recreated 3D information. Force information mediated through joint, muscle and tendon receptors can be used to infer when objects are in contact. In addition tactile information can be used to infer the texture of surfaces (Section 2.2.2). This information is conveyed by Pacinian corpuscles, pressure receptors, present under the skin and concentrated at the tips of the fingers. These two modalities are commonly referred to as either force or haptic feedback.

Problems may arise in teleoperation when either the visual or haptic information fed back to the operator is suboptimal or impoverished and also when the common representation offered by both visual and haptic information is in conflict.

2.2.1 Vision: Effects of frame rate and time delay

Most studies which have varied visual feedback parameters have been predicated (often tacitly) on teleoperation applications in space, where issues of pure video bandwidth and pure time delay dominate other considerations[1].

Several studies carried out at MIT, and summarized in Sheridan [1], have explored the interaction between the quality of video information and operator performance. Ranadive [11] and Deghuee [12] considered how teleoperator performance varies with changing frame rate, resolution and grey scale. The product of these three determines the bit rate, which has as its ceiling the bit rate bandwidth of the link. By systematically varying each parameter, Ranadive found an almost direct correlation between operator performance and the bit rate of the image data displayed, concluding that the three parameters weigh equally.

[1] An additional visual uncertainty in these types of study, particularly those designed for space implementation, is to what extent the operator's raw perception of shape and depth are being interfered with by lack of stereo information, or by unnatural lighting confounding shape-from-shading cues, or by lack of everyday objects with which to establish scale. Sheridan [1], for example, has examined the adverse effects of poor depth perception on operator performance undertaking a manipulation task.

Operator performance is little improved if the frame rate is increased above 15 Hz, but if the rate is taken below some 5.6 Hz teleoperation is all but impossible.

Considerable attention has been given to the problems caused by pure transport delay in feedback. Eliminating this delay by the use of predictive displays which copy the operator's actions onto a virtual world, and generate visual feedback from that world, has been shown to reduce operator confusion and fatigue substantially. Operators view prompt graphical output generated from a modelled scene, either overlaid as wireframes on the video (albeit time delayed), or rendered either artificially or by video warping. They become able to perform smooth ballistic motions, reducing task completion times in some cases ten-fold (Kim and Bejczy, Sheridan, Kim) [49–51]. It is worth noting that this use of scene models differs fundamentally from ours. In systems with large time delay, the interactions of slave and scene are synthesized, placing very high demands on the fidelity of geometric and dynamical modelling. In our case, because there is negligible time delay, force and visual signals are available from the real world, providing actual rather than synthesised measurements of response.

2.2.2 Force: Frequency response

Whereas work on visual feedback has been largely phenomenological, the greater degree of similarity between robotic and bio-mechanical systems has allowed a concomitantly greater degree of sophistication in the treatment of force feedback within teleoperation. For example, a description of the physics of force reflection in teleoperation was given recently by Daniel and McAree [7]. Their principal theme was how to strike a compromise between the operator's desire to retain perceptual fidelity by including force feedback at higher frequencies, while at the same time ensuring that the energy transfer at mid-range frequencies does not drive the system unstable. They argued for notch filtering the slave force signal between 0.5 and 30 Hz before reflecting it to the operator. The force information above 30 Hz is sufficient to stimulate the Pacinian corpuscles, touch-sensitive cells in the operator's hands, and by allowing appreciable energy transfer only below 0.5 Hz they were able to increase the force reflection ratio but still maintain stability under operator control.

2.2.3 Force: Proprioceptive frame

The receptors in the muscles, tendons and joints are also responsible for sensing the movement of limbs and is termed proprioception. Proprioception, "sense of self", or kinesthesis enables the operator to unconsciously monitor the position of his/her body. Problems occur for the operator during teleoperation if the various frames of reference attached to the head and body, master input device, slave arm and virtual camera — proprioceptive and visual frames — are misaligned causing the operator to lose proprioception. Indeed one must always be careful when combining force and visual information.

2.2.4 Force and Vision

The psychophysical literature suggests that when both force and vision modalities are used, they should neither be examined in isolation nor combined linearly. Often it seems that vision dominates perceptual judgments in humans. Most commonly reported are the debilitating effects of gross mismatches between visual and tactile frames, when the placement of cameras causes, say, a forward-backwards motion at the input device to appear as a left-right motion at the slave. Sheridan [1] used the term tele-proprioception to describe the common-sense need for correlations between motion of the operator, motion of the slave, and what the operator sees and feels. More subtle effects are observed (though never reported during teleoperation) when the expected relationship between image position on the retina and position in the world is disturbed. An often cited example in the human visual system (see Posner [15]) is that of Gibson [52] in which subjects wore prismatic spectacles. When subjects watched their hands move along an actually straight edge, visual information showed it to be curved and, remarkably, the edge *felt* curved. Rock and Victor [53] conducted experiments where subjects grasped a square viewed through an optical device that distorted the image to make it appear rectangular. The subjects saw and felt a rectangle — Rock and Victor referred to visual dominance as *visual capture*. Ernst *et al.* [54] conducted experiments to observe the persistent effects of haptic feedback on visual perception rather than concurrent perceptual effects, they reported that haptic feedback increases the weight of a consistent surface slant signal relative to inconsis
yielded from visual cues. Indeed, Klatzy *et al.* [55] reported that even for non-ro
manipulation tasks the importance of attentive foveal visual feedback is much reduced

spatial configurations where force feedback gives reliable spatial information hence increased informational weighting. Many authors have reported on visuomotor adaptation showing that humans can adapt behaviourally to changes in the mapping between the environment and sensory signals (Rock [56], Harris [57], Welch [58], Stratton [59]). The plasticity of the visuomotor system was first explored by Helmholtz [60] by displacing the entire visual field with prismatic spectacles and studying subsequent effects on reaching and other sensorimotor behaviour. He reported that the initial reaches were extraordinarily inaccurate, but they rapidly improved. The observed adaptation could theoretically occur in visual perception, in proprioception of body parts or in the translation from visual to motor coordinates.

Flanagan and Wing [61] make clear that visuo-haptic sensory information is not only used as feedback to adjust movements that fail to attain their targets, but also, to correct inaccurate stored information about the environment and/or effector system — they refer to the slow continual update of the operator's internal sensori-motor models as plasticity. This serves to reduce the likelihood of correction being required on another occasion. Although predictive or anticipatory control may occasionally result in large movement errors when, for instance, one picks up an empty box which one thinks is full, the alternative control strategy — reactive control based on sensory feedback — cannot support the fast and accurate movements observed in most natural actions given the large time delays associated with receptor transduction, neural conduction and processing (Flanagan *et al.* [62]). A corollary of learning and maintaining the ability to generate motor commands for desired actions is the prediction of the sensory consequences of these motor commands. Flanagan and Wing [61] and Miall and Wolpert [63] postulate that the ability to predict the sensory consequences of motor commands may be based on internal forward models that mimic the behaviour of motor system and manipulated objects. Harris and Wolpert [64] take this idea one step further and propose that the variance of the motor noise is proportional to the strength of the control signal and used this single physiological assumption to predict the empirical Fitt's law. While these studies offer a qualitative discussion of the operators ability to reweigh visual and haptic information, the experiments of Massimino and Sheridan offer a more quantitative viewpoint.

Massimino and Sheridan [65] investigated how teleoperator performance is affected

under various force and vision feedback regimes. These studies certainly supported earlier ones by Hill [66], Brooks [67] and Vertut and Coiffet [68, 69] who found that providing meaningful force feedback substantially reduced task completion times; however they also found that force feedback is particularly important first at low frame-rate and, secondly, when the operator is viewing a workcell at a small subtended visual angle. (Small viewing angles tend to remove parallax and hence differential depth discrimination.)

2.2.5 Integration of visual and haptic feedback : Conclusions

This section has discussed the studies which have measured operator performance under varying parameters associated with visual and haptic feedback. In considering the degradation of operator performance due to suboptimal visual information alone, the literature noted the debilitating effects of time-delay and frame rate on operator performance. These studies were largely motivated by the design of space teleoperation systems. Their conclusions find that visual display systems should have low latency and frame-rates of at least 15Hz. Section 3.3.3 of this thesis details the design steps in order to realise a low latency frame rate (25Hz) rendering system.

The literature concerned with just force feedback has concluded that high frequency force information is important to convey a sense of touch to the operator, Oxford's teleoperation system can convey force information at some 100Hz (Section 3.2.1). When force and vision modalities are combined then in the broad the largest degradation in operator performance will occur when tele-proprioception is not maintained, this issue is addressed in Section 3.3.1. A secondary issue, provided force and visual information do not conflict, is the need to supply high quality force feedback particularly at low frame rate and at small visual angles.

3

Equipment

3.1 Introduction

The Oxford force-reflection teleoperation system has been developed over a number of years in the Department of Engineering Science by Daniel, Fischer and McAree [70, 71]. As well as providing a valuable tool for research into fundamental issues in force reflection and operator performance, it also forms the core of a commercial teleoperation product for nuclear decommissioning [7].

A principal attribute of the device is that the slave force sensor measurements reflected to the operator are partitioned into two frequency channels. The low frequency channel contains force information that the operator senses through muscles and tendon receptors and responds to using his or her motor system. Considerable energy transfer occurs at these frequencies, determining the stability of the teleoperation system as a whole. The high frequency channel is much less energetic and excites only skin receptors, contributing to the operator's sense of touch of the environment. This chapter begins by briefly describing that system.

For studies of force reflection, and indeed in its commercial use, the system was used by an operator who viewed the workcell either directly or from a TV monitor fed from a

static camera. The chapter continues by describing the design and build issues involved in augmenting the system with a *virtual viewing system*, where the operator is able to select his or her viewpoint and viewing direction onto the workcell by first tracking the pose of known objects in the workcell using a computer vision system, and then rendering them graphically on a display in front of the operator.

The force-reflection system, the vision system and renderer — combined with the sensor, perceptual and motor characteristics of the operator — form a complex interacting real-time system. The chapter will discuss how basing the system architecture on a finite state machine has been found to reduce the design and debug time.

3.2 The force-reflection system

The teleoperation system was designed specifically for the nuclear decommissioning industry, and so uses as the "slave" arm a well-proven industrial Puma 560, manufactured by Stäubli[1]. The six-jointed arm is mounted on the floor and is able to access a workspace of some 2 m^3.

The slave manipulator's position and orientation are determined by those of the "master" input device, which is a 6 degree-of-freedom parallel mechanism attached to a joystick held in the operator's hand. Designed by Daniel and coworkers, the input device has low-mass and high-bandwidth, and its kinematics are similar to those of the Stewart platform or Stewart mechanism [71, 72] where the actuated prismatic mid leg joints have been replaced with one DoF. rotational joints and the base joint with an actuated revolute joint. Its pose is determined from encoder readings input into a real-time (and remarkably efficient) algorithm to solve, with guaranteed convergence, the forward kinematics of the Stewart Platform [73]. The working volume of the input device is bounded by a cylinder of radius 50 mm and length 120 mm, with $\pm 30°$ rotation available in roll, pitch and yaw at the joystick.

When the slave arm moves to touch objects, the forces and torques acting on the end-effector are detected by an ATI force/torque sensor [74] attached to the base of the robot's wrist. Its readings are returned back to the master input device, and force reflection is

[1] Stäubli bought out the pioneering Unimation Corporation in 1989.

achieved through direct-drive actuation of limited-angle torque motors. The small motion bandwidth of the input device is 100 Hz, where small is defined as up to 1 mm in translation and 2° in rotation. It has a force threshold of 0.1 N, and an inertial load of 0.1 kg.

The various signals are mediated by a controller running on a PC under the QNX real-time operating system [75]. Its architecture is discussed in more detail later. The system normally operates under direct position-force control. That is to say the operator's position commands are fed forward to the slave manipulator for fine control and the sensed force signal is fed back to the master. Alternatively the teleoperator can act under rate-force control, where the operator's positional commands are mapped onto slave velocity commands for coarse control.

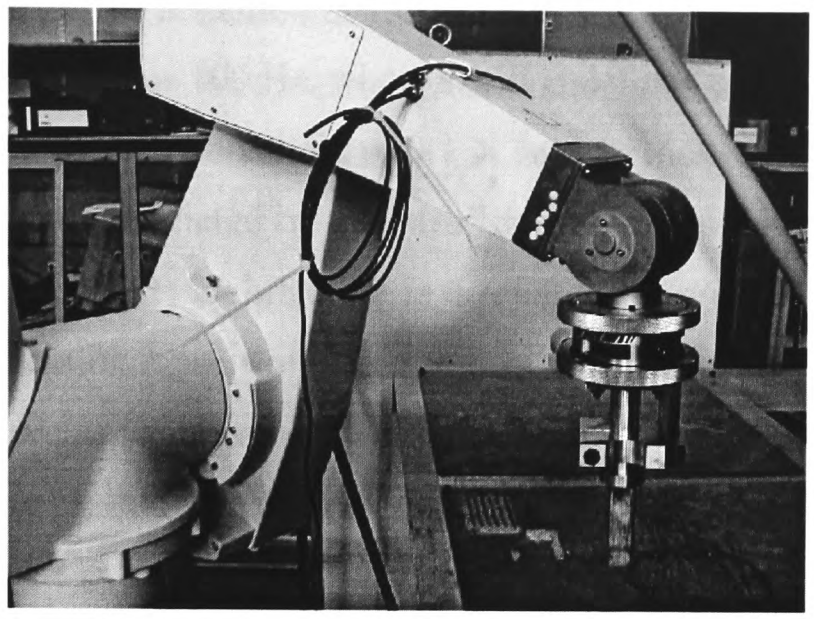
Figures 3.1(a) and (c) show general views of the slave manipulator and master input device, Figure 3.1(b) shows an annotated close up of the force sensor and Figure 3.1(d) the master's torque motors.

3.2.1 Frequency response of force reflection

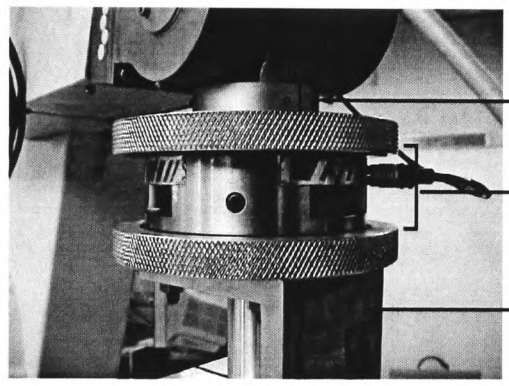
The bandwidth of the Puma slave arm is 12.5 Hz, much higher than the value of 2.5 Hz characteristic of the operator's arm motions [7]. The master device however has an output bandwidth of some 100 Hz. This asymmetry is required because the frequency range over which a person can *feel* is much higher than the frequency range over which they can *act* via the human motor system. High frequency motions/vibrations (above 30Hz) stimulate the touch-sensitive Pacinian corpuscles in the operator's hands, and increase the fidelity of the reflected force. This high frequency information carries little energy and is passed from slave to master unattenuated. At frequencies lower than 2.5 Hz, force is sensed through the operator's joint, muscle and tendon receptors, and the operator is able to respond to, and stabilize, low amplitude disturbances at these frequencies. However, to reduce the effect of high energy transfer and mismatch of the slave manipulator and operator arm masses, the low frequency information is attenuated.

Operators are unable to stabilize systems where significant energy transfer occurs at intermediate frequencies, and Daniel and coworkers use a notch filter to remove reflected force signals in the $5 - 25$ Hz band.

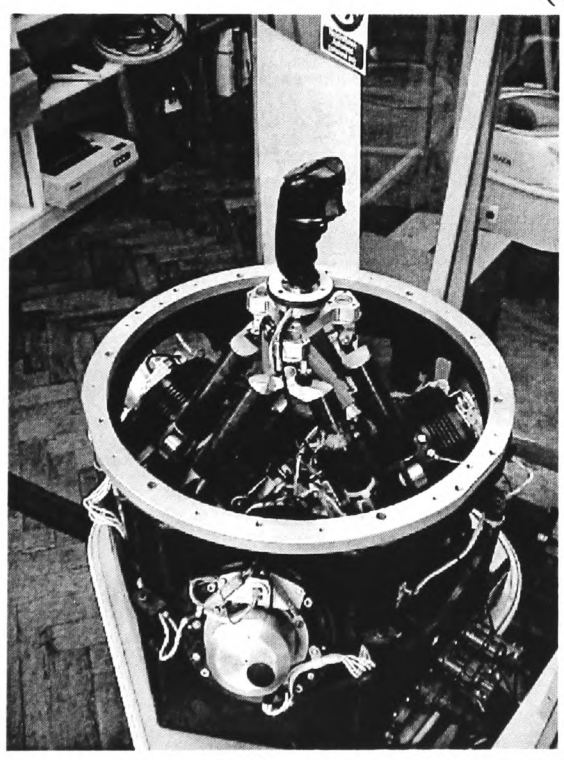
An interesting trial is to make tapping actions with the input device so that the slave



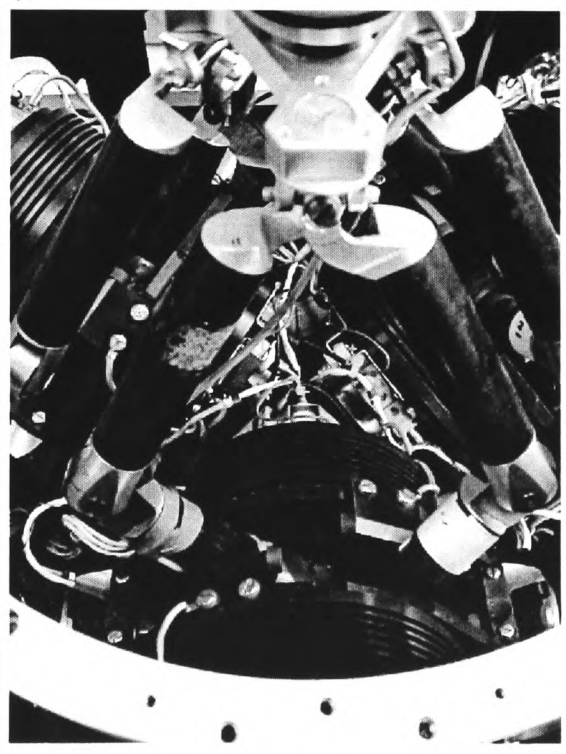
(a)



(b)



(c)



(d)

Figure 3.1: (a) The articulated slave robot arm and (b) a close up of the the force/torque sensor. (c) The master input device. Its kinematics resemble those of the Stewart platform, and the joystick moves with six degrees of freedom. The close up in (d) shows two of the six torque motors connected to two of the six platform legs.

impacts a hard surface. With the notch filter in place, the "feel" is one of tapping the surface with a wooden shafted hammer. As the notch filter is removed progressively it is as though the shaft was turning to metal and the resulting jolt becomes unpleasant, unpredictable and uncontrollable.

3.3 Design issues in the provision of virtual views

As mentioned earlier, the teleoperation system was originally used by an operator who viewed the workcell directly, or from a TV monitor fed from a static camera. We now discuss broad issues of concern when planning the provision of a virtual viewing system for the operator.

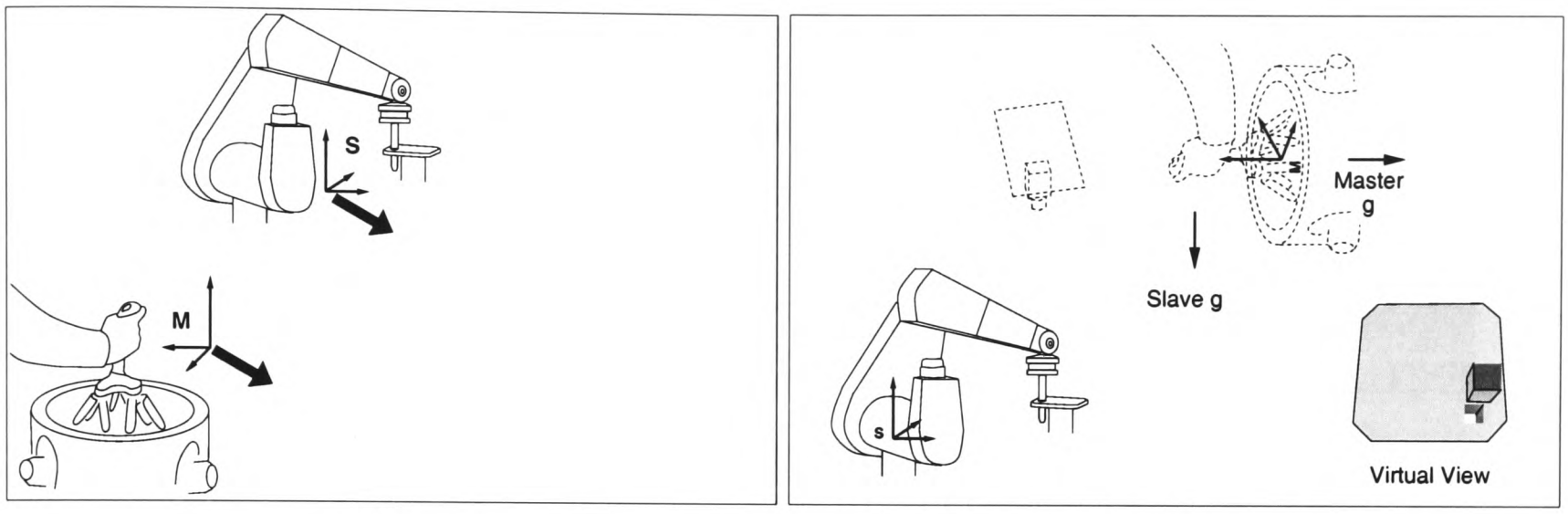
3.3.1 Issue I: Proprioceptive alignment

Problems occur for the operator during teleoperation if the various frames of reference attached to head and body, master input device, slave arm and virtual camera, are misaligned. Conflicting orientation information upsets the operator's *proprioception*[2] or "sense of self", a term used to describe the sensed movement of limbs mediated by the body's muscles and tendon receptors. Sheridan [1] noted in 1992 that achieving proprioceptive alignment was an open research question. With so many limb-centred frames involved, so much potential for underlying task dependencies, and the difficulty in making meaningful measurements of operator performance over a large number of parameters, there still remains no convincing codification today.

One of our main concerns was the alignment of the operator's visual and hand motion frames. In the laboratory, and under *direct* viewing conditions, the operator stands behind the master input device and looks towards the slave. The slave's frame is rotated about the vertical so that a rightwards movement of the joystick results in a motion of the slave that appears rightwards to the operator, as sketched in Figure 3.2(a). Of course, the angle of rotation has to be found just once, as the input device is fixed on the floor.

Now consider viewing the workspace through the virtual camera after it is rotated from this starting position by the transformation $\mathtt{R}(\phi, e, c)$ where the triple of angles is ϕ about the

[2] sometimes the Greek *kinesthesis* or "sense of motion"



(a) (b)

(c) (d)

Figure 3.2: (a) For telepropriception alignment when the operator is directly viewing the workcell, the master frame to slave frame orientation must be specified. (b) When viewing the workcell via a virtual camera, placing the virtual master manipulator directly behind the virtual camera is unnatural for arbitary rotations (ϕ, e, c). (c) The preference for consistent alignment of the gravity vector at the master and slave can be satisfied by applying only a rotation $\mathrm{R}^{-1}(\phi, 0, 0)$ to the slave frame, in effect keeping the virtual master vertical. (d) To achieve telepropriception the virtual cameras pitch and yaw frames were fixed to the operators corresponding frames. Which was chosen when the operator was viewing the monitor at the slave.

vertical axis, e in elevation, and finally cyclotorsion c about its optic axis. One possibility is always to place the virtual master device in the same relative position behind the virtual camera. In practice, it is the inverse transformation $\mathtt{R}^{-1}$ that is applied to the previously rotated slave frame, so that once more rightwards movements appear to the right in the virtual view.

While this is natural enough if e and c are zero, it feels wholly unnatural if they are not. The reason lies in our ability to sense the direction of gravity via the directional static loading in our bodies and via the vestibular system. Starting in a vertical position, the operator establishes that up-down movements of his hand are in the direction of gravity *at the master*, and that this movement is associated with up-down movements at the slave — so that the "insertion direction" of a peg in the hole coincides with the gravity vector.

Now move the virtual camera and master as shown in Figure 3.2(b). The operator still senses that gravity at the (real) master is associated with up-down movements of the hand. However the virtual view indicates that the operator must push forward on the joystick to insert the peg. So, gravity at the slave is in the forwards direction, and orthogonal to that at the master. The vestibular system is also confused, because the virtual view of the slave tells the operator that his head is horizontal looking downwards, but it is actually vertical[3].

The preference for consistent alignment of the gravity vector at master and slave can be satisfied by applying only the rotation $\mathtt{R}^{-1}(\phi, 0, 0)$ to the slave frame, in effect keeping the virtual master vertical, as shown in Figure 3.2(c), but allowing the virtual camera to move with two additional degrees of freedom in elevation and cyclotorsion with respect to the virtual master.

To achieve full proprioceptive alignment with a virtual camera able to move with these degrees of freedom e and c requires a viewing device that moves with the operator's head. This could be realized using a head mounted display (HMD), but operators report finding their bulk uncomfortable (Jang *et al.* [76]), and there are unresolved issues of quite how to control the virtual camera from the operator's head position. For example, if just the direction of gaze were copied, rotating the head downwards would cause the virtual camera

[3]This confusion can be experienced in the "rotating room" often found at fairgrounds. The victim sits in a chair in the middle of the room. The entire room is then rotated producing the sensation that the person is being turned upside down. However when the person tries to move their actual and sensed movement feel out of kilter – they feel they should be pulled to the floor, not the ceiling.

to rise above the workcell, and this might feel unnatural. We did not use an HMD in this work, preferring not to impose a device between visual output and operator that could itself introduce artifacts unrelated to the main goal of our work.

Instead, using the method of Chapter 6, we determined that for insertion tasks, an optimal elevation of the virtual camera was $e = 45°$, and the cyclotorsion $c = 0°$. A fixed monitor was installed tilted up at 45°, and set at a position just beyond the master input device where the operator gazed down at 45°, as shown in Figure 3.2(d). The operator was thus proprioceptively aligned for any value of ϕ. As the plots in Chapter 6.3.3 show operators were able to function satisfactorily at elevation angles of the virtual camera ranging from $30° < e < 60°$.

3.3.2 Issue II: Viewpoint selection

An important aspect of supplying the operator with a virtual view is the choice of a convenient and intuitive viewpoint for different phases of the teleoperation task.

Das [77] has explored whether it is useful to automate the selection of the operator's viewpoint. He measured the times to completion for a task in which a slave robot's end-effector had to be moved from a random starting point to touch a target while avoiding two obstacles. In one set of trials the viewpoint was chosen by the operator and in another set by his "best-view" algorithm. This best-view algorithm constructed three planes perpendicular to, and bisecting, three lines: that between the end-effector and target, and those between the end-effector and obstacles. The viewpoint was placed at the intersection of the planes, and the viewing direction set towards the target. In this way, none of the three lines is grossly distorted under perspective projection. He found that optimal teleoperation was achieved when the viewpoint was firmly under operator control. Unfortunately it is unclear from Das' work whether it is the whole notion of guessing the operator's desired optimal viewpoint that is inherently flawed, or whether the fault lies with the particular algorithm he used to generate it.

In our system the *operator* controls the viewpoint using the master input device. Two viewpoints are stored by the system, and one is designated the current view. There are two modes of using the views. The first mode "bookmarks" the current viewpoint, and toggles the current view to the other viewpoint. The second alters the current viewpoint and

direction. When it is engaged the master input device is temporarily released from master-slave interactions and is used as the virtual camera positioning device. The current master's pose is taken as the coordinate frame origin and all further poses are measured relative to this frame (Figure 3.2(a)). Just as with normal teleoperation, it is useful to provide both a position to position mapping for fine control and a position to velocity mapping for coarse control.

3.3.3 Issue III: Object rendering

Providing visual feedback for the purpose of assisting in the completion of a dynamic task raises some interesting questions about what visual information should be presented to the operator, and how it should be presented.

In our context, vision should supply the 3D geometric understanding of the environment which can usefully be fused with information from force sensing. Time-evolving orientation, position, proximity to contact are important strands of information: there is no requirement for a photo-realistic virtual world (Das [77]). But it is also important to realize that *exact* geometric alignment using vision is not required. As noted by Klatzky [55], force-feedback conveys much of the geometric information needed by the operator in the final contact position stages of a manipulation task.

The brain recreates 3D visual information from a variety of cues of which binocular disparity, or stereopsis, is the predominate means of gauging depth. Unfortunately by providing the operator with only a monocular view of the environment this depth cue is lost. There are other ways of providing depth information in monocular images. Partial 3D information may be recovered by the use of shadows, prior knowledge of the relative sizes of the workcell objects (eg. with a fixed size marker attached to each object), motion parallax provided by allowing the operator to smoothly change his viewpoint, and perspective effects. Early studies, conducted in 1964 by Kama and du Mars [78] into the debilitating effects of monocular views compared to stereo found little noticeable difference between task times between the two. They speculated that this may have been because of the use of force-feedback. However, another study by Chubb [79] in the same year, on the same equipment, and using the same subjects, reported a 20% decrease in task times using stereo. It was noted in Section 3.3.1 that an HMD is needed for correct proprioception alignment

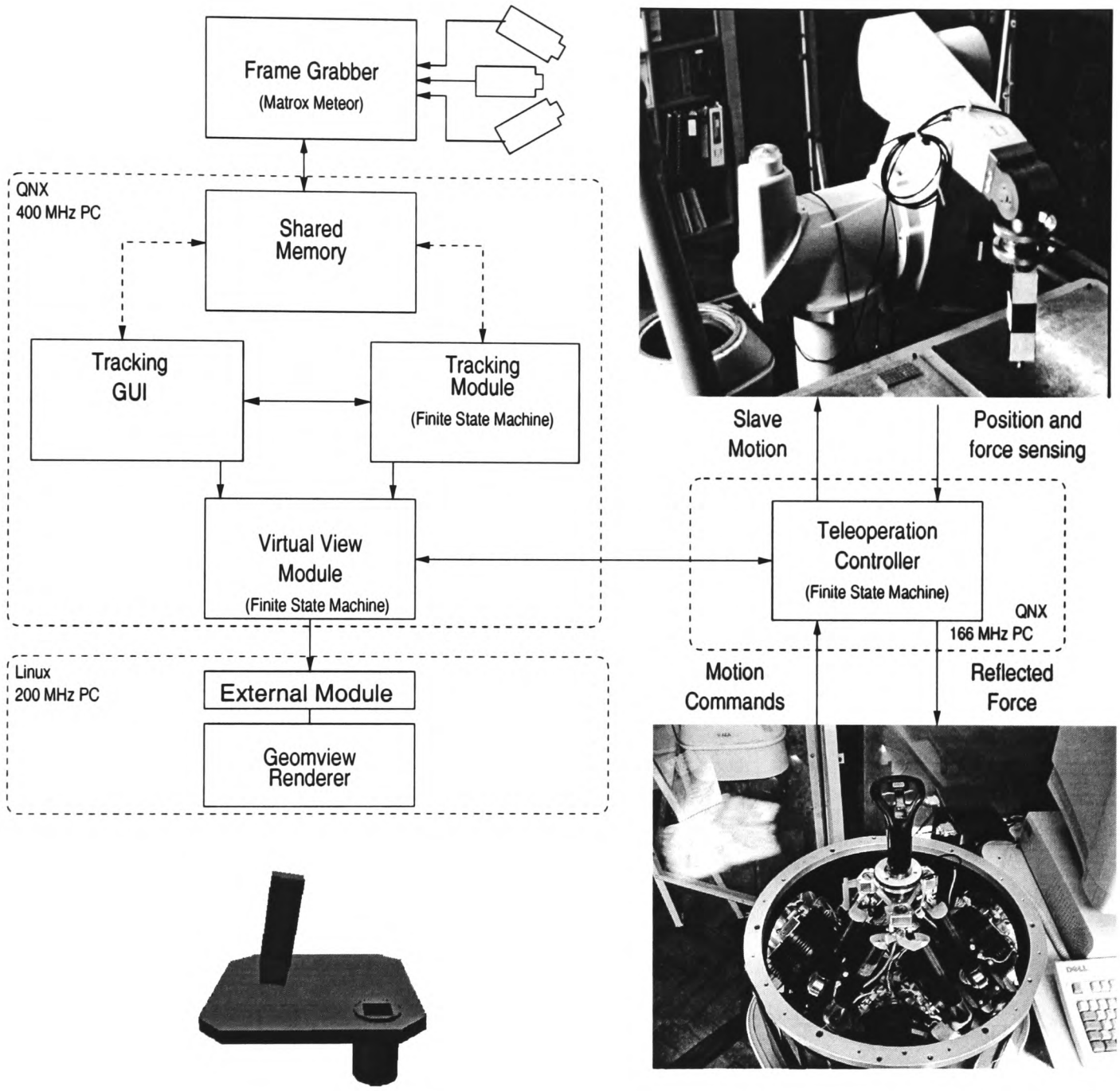


Figure 3.3: The overall layout of communication between the processes and sensors in the Oxford Teleoperation System.

when the operator wishes to view a workcell using six degrees of freedom. HMDs also provide stereoscopic views. As we noted before, we have not used one, because their disadvantage would mask their advantage. Though an HMD is a useful addition, operators have found their bulk to be uncomfortable and so we just provide monocular views. In this provision, all but one important monocular depth cue, by the use of shadows, can be provided by a hidden line wireframe object viewer. The inclusion of shadows requires the use of a somewhat more sophisticated object rendering package.

Rather than discriminate between the two viewing approaches we have designed our system to provide both wireframes and a solid object rendered view of the workcell. What we initially found was that the operators immediately preferred using the solid object rendered view, perhaps because of the obvious likeness between the virtual objects and their workcell counterparts. Later, however, they did not seem to mind which viewing system was used. Indeed we found no difference in the time taken to complete various teleoperation tasks between two views.

Although the virtual viewing system was designed so that the positions of the virtual objects could be updated by various sensors, it was clear from designing the visual pose sensor that these positions would be subject to the dynamics of the individual pose sensor. Hence we were unsure when designing the viewing system whether the positional information should be processed so as to be better matched to the operator. This issue is tackled in Chapter 6.

3.4 Overall system architecture and implementation

The overall interconnection of the system is shown in Figure 3.3. The teleoperation controller and the visual tracker run on two PCs under the QNX real-time operating system, and communicate over a dedicated Ethernet link. The third computer runs the solid object renderer software, Geomview.

3.4.1 Teleoperation control

The force-reflection part of the Oxford teleoperation system was originally designed and implemented by Fischer and Daniel in 1992, but was comprehensively re-written by McA-

ree and Daniel in 1996. It runs on a single PC under QNX.

QNX is an ideal choice of OS because it runs on PC's and so is compatible with a wide range of commercial I/O boards. It is also supported on a commercial basis and its development team are keen to write or debug drivers for new hardware. Its primary benefit however is that it is a genuine real-time OS. Of particular relevance is its priority-driven preemptive scheduling, guaranteeing completion of critical robotic processes within a fixed time frame. QNX uses a micro-kernel, and additional devices are added as communicating processes, and not compiled into a monolithic kernel as in Unix (and, unfortunately, Linux). The programmer has considerable control over the priority given to tasks that in other operating systems would be regarded as "untouchable".

Interprocess communication is via message passing, which makes it conceptually easy to split the controller and vision into separate sets of communicating concurrent processes possibly running on different processors, and each handling one well-defined part of the whole.

Message passing is also a means of synchronising the execution of several processes. This type of message is labelled an *event*. The typical life-cycle of a process is to execute, to issue an event-message and then to block, awaiting a reply. Blocked and un-blocked processes are continually being swapped out and in by the process manager, fast context switching allowing this continual swapping not to burden the system.

For any robotic processes which need to share large amounts of data between other processes a shared memory segment is initialised and paged into the process' local address space. The process addresses the shared memory segment as though it belongs to the process, semaphores are set up as guards to the data to synchronise atomic updates.

The teleoperation controller coordinates the control of two subordinate processes, one controlling the master and the other the slave manipulator, under force-position control, where the slave arm is driven under position control to follow the motions of the master, while contact forces sensed at the interface between the slave and the environment are used as inputs to the force-controlled master.

The coordinating teleoperator controller pieces together the input and outputs of the two subordinate processes to implement force-reflection teleoperation. The controller executes in a loop interrogating the master for its current pose, which it filters and transforms

into slave space, before reflecting to the slave. The slave is interrogated for the current force/torque measurements, which the controller filters using a notch filter before reflecting to the master.

The code libraries for the force-reflection system of McAree and Daniel are written using C, and comprise some 100,000 lines of code. The high level logic of the controllers is implemented in three finite state machines, one for each controller.

3.4.2 Visual Tracking

QNX turns out to be an ideal OS under which to run the tracking module as well, chiefly because of the need to communicate to the other teleoperation processes on the network using message passing, but also because it allows maximum use to made of modest processing power by reducing OS overheads. QNX's real-time features reduce the task complexity, ensuring that the tracker runs in a fixed time-slice. Under Linux, the online tracker would require an extra layer of checking to check for dropped frames as other processes of universal priority are paged in.

Three monochrome video streams are synchronized and captured into the RGB planes of a PCI-bus colour framegrabber (Matrox Meteor). The device driver was ported from the Linux driver to work under the QNX operating system.

As more fully described in Chapter 5, once the Meteor has notified the tracker that a new frame has been received the tracker projects a wireframe model of the tracked workcell objects at the object's predicted location in each of the images. A search is initiated along normals to the wireframe edges and the actual edge is localised. This measurement is fused with the predicted object pose to form the filtered object pose.

The vision libraries have been written in C++, chosen because of the need both to control and access low-level device drivers such as the frame-grabber and to handle higher-level geometrical entities with a degree of abstraction. The supervisory control for the vision system follows the lead of the teleoperation controller and is implemented as a finite state machine.

3.4.3 Virtual View Controller

The virtual view controller is responsible for setting up proxies to other processes to interrogate the geometric state of the system for subsequent display to the user. Its state is updated by accessing the various processes so that i) the poses of the objects are updated as soon as they become available and ii) the operator can alter the virtual cameras view point and direction. The operator has the choice of viewing either the controller's GUI which displays hidden line wireframe graphics of the workcell objects set in the current virtual camera frame or alternatively a solid object renderer (see next section) with which the controller establishes communication via a TCP/IP socket.

The virtual view controller usually uses the object poses as measured by the visual tracker. The camera's view and direction is altered by setting up proxies with the teleoperation controller which trigger when the operator wishes to alter the camera position. When a switch camera event is received the current camera pose and the stored camera pose are swapped and the updated pose is sent to the external viewing system. When a camera move event is received then i) the teleoperator is released from master-slave interactions, ii) the master's current pose is reset to the master origin, and iii) the pose of the master is sampled at 25Hz. This pose is used to compute the virtual camera pose, which is transmitted to the external viewer process.

3.4.4 Renderer

The viewing system uses the Geomview object modeller to render the objects being handled in the robot's workcell. Geomview [80] was chosen because it interfaces easily with any external process supplying pose data. The modeller runs on a separate machine, so in our case this external process doubles as a communication buffer. The process receives pose data from the virtual view controller using the angle-axis and translation representation via a TCP/IP socket, and translates it into Geomview's internal pose format before passing it on to Geomview itself.

The transmission of the pose data from the tracking process computer to the renderer incurs a round trip delay of 2 ms, which has a negligible effect on the operator's ability to function [49].

Figure 3.4: An example of the user interface : The operator can use the joysticks buttons to change teleoperation mode. When the trigger button is depressed a camera switch event is emitted. When both the re-index and trigger buttons are depressed a camera move event is emitted and the master input device can be used as a camera positioning tool.

3.4.5 Additional aspects of the operator interface

Graphical User Interface. The operator can interact with the system using lightweight GUIs designed in QNX's Photon Application Builder.

Joystick Buttons. Another means of generating operator events is by pressing various button combinations on the master's joystick. In Figure 3.4 the switch camera event is issued when the deadman and trigger buttons are depressed and the move camera event is issued when the reindex button is depressed in addition.

3.5 Details of the system architecture: the use of a finite state machine

Each of the various control and tracker processes is built as a finite state machine (FSM) which allows the supervisory control structure to be divorced from the low level code. We explain FSM design by starting from the design of a conventional event-driven graphical user interface.

3.5.1 Some background

A GUI is composed of various graphical widgets, each of which places a labelled event onto an FIFO event queue when the user interacts with it. The programmer associates a callback function (or event handler) with each widget event which is executed when the event reaches the head of the queue. Each callback function could be said to perform an action, or string of actions.

Suppose that we have two buttons which when pressed emit events E0 and E1, which result in callback actions A0 and A1 being executed respectively. A state diagram with just one state is given Figure 3.5(a). Now suppose that we required action A1 the first time button event E1 occured and a different action A2 the second time it occured, and so on alternately. The programmer used to callbacks would probably suppose that there was still one state, as shown in Figure 3.5(b), and simply amend the callback code to include a conditional, and introduce a flag called "pressed" which is initialized at 0.

While this is straightforward enough for a single flag, coding in this way as further flags and conditions are added is error prone. Already callback action A0 holds the seeds of later confusion because it fails to mention "pressed" at all.

The finite state programmer requires actions to be unconditional, and augments the diagram with a second state, S1, as shown in Figure 3.5(c). Which state transition occurs, and hence which string of actions occurs, depends on i) the current state and ii) which event has been received.

Now the *actual* code must look more like that in Figure 3.5(b). The reason is that in the finite state machine of Figure 3.5(c) there are two states which wait for events, but in practice in a single process there must be single point in the code where event notification occurs. We use a software package, Libero, to translate the specification of the state diagram, events and actions as a finite state machine into lower level code in C (for the force-reflection FSM) and C++ (for the vision FSM).

The Libero program source for the above example is given below, where the states are delimited by `:`, events are denoted by `(--)` and the state transition by `->` and `+` list the actions

```
S0:
```

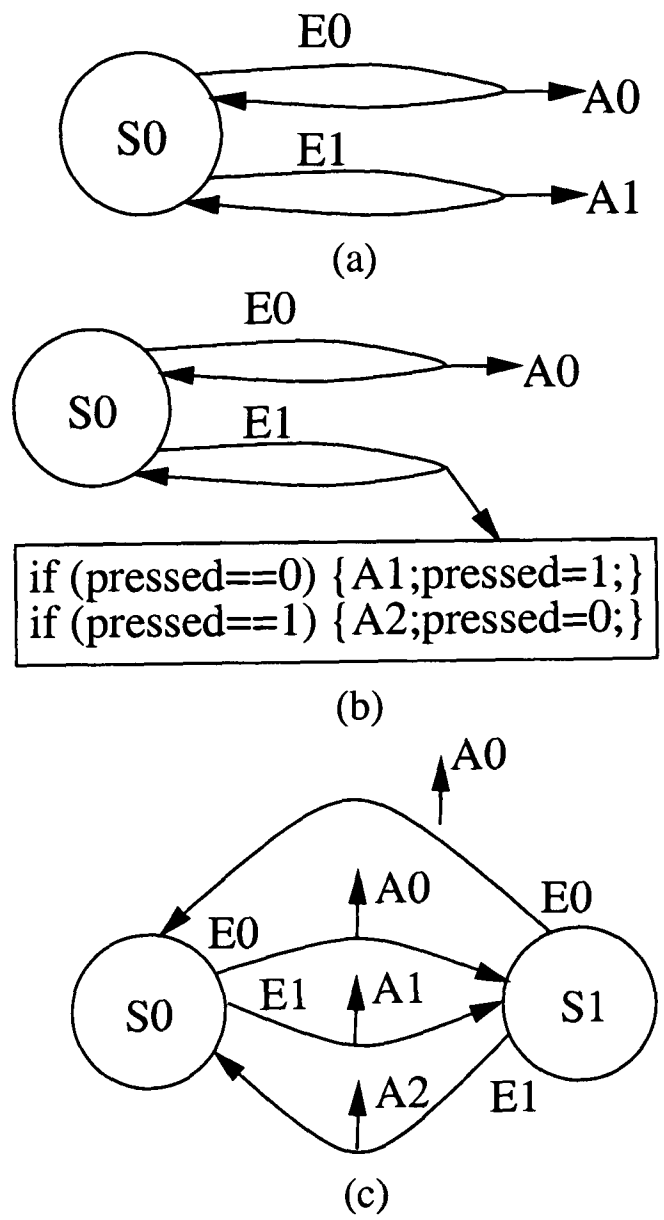


Figure 3.5: (a) A single state program with callback actions. (b) A two state system, but obfuscated by using conditional callback actions. (c) A proper finite state machine representing (b).

```
    (--) E0                    -> S1
         + A0

    (--) E1                    -> S1
         + A1

S1:
    (--) E0                    -> S0
         + A0

    (--) E1                    -> S0
         + A2
```

3.5.2 Example of applying a FSM to our task

The design steps in constructing a FSM for a real-time process are then:

1. Identify the key stimuli (events) of the system — e.g. a frame-grabber signal notifying the receipt of a new image.

2. Identify the key states — e.g. Meteor-capturing state, Meteor-not-present state, and so on.

3. Sketch in the action or string of actions associated with each transition between states. The aim should be to have relatively few "atomic" actions within a string. (An example of an atomic action is start-meteor-capturing.)

4. Describe the state diagram in a text file using Libero's syntax [81].

5. Use the Libero translator to compile the text into a C++ state machine class, generating the actions as static member functions.

6. Fill in the procedural code required for actual actions.

We illustrate this technique by designing a small subsection of the tracker's state machine (Chapter 5). Although the system can receive many events indicating or requesting

various information, the key event is generated by the frame-grabber notifying the receipt of a new frame, *Meteor-Event*. The states of interest are the *Tracking* state indicating the process is currently tracking, and the *Tracker-initialised* state indicating the tracker is initialised. The state machine expressed in Libero's syntax is shown in Figure 3.6.

```
Tracking-State:
  (--) Meteor-Event                    -> Tracking-State
       + Execute-Tracking-Cycle
       + Receive-Event

  (--) Turn-Tracker-On-Event           -> Tracking-State
       + Reply-ERROR
       + Receive-Event

  (--) Turn-Tracker-Off-Event          -> Tracker-Initialised-State
       + Reply-OK
       + Receive-Event

Tracker-Initialised-State:
  (--) Meteor-Event                    -> Tracking-Initialised-State
       + Receive-Event

  (--) Turn-Tracker-On-Event           -> Tracking-State
       + Reply-OK
       + Receive-Event

  (--) Turn-Tracker-Off-Event          -> Tracking-Initialised-State
       + Reply-ERROR
       + Receive-Event
```

Figure 3.6: The state machine expressed in Libero's syntax.

Events which are not generated by timing signals are acknowledged. If the event is invalid or not defined in a particular state such as the *Turn-Tracker-Off-Event* in the *Tracker-Initialised* state the calling process is notified by executing a *Reply-ERROR* action. If the event is valid however then a *Reply-OK* action is executed. The final action in each string of actions is the *Receive-Event* action. This places the process, if no new events are awaiting transaction, in a receive-blocked frame. This effectively removes the process from the OS. manager's list of current processes until an event is generated specifically for the process.

Running Libero produces templates for that action subroutines, which must then be filled in. For example, the C++ code required to define the *Execute-Tracking-Cycle* would look like

```
FSM::execute_tracking_cycle() {

    tracker.predict();
    tracker.gen_measurements();
    tracker.measure(meteor.current_images());
    tracker.update();

}
```

It should be noted that Libero's FSM syntax is rather simple. It is, for example, unable to implement FSM nesting which is useful where sub-blocks of an FSM involve little external interaction and use an exclusive set of events and actions. Without this facility, and without other FSM language constructs available in a commercial product like Stateflow [82], Libero's FSMs realizations tend to be large and monolithic.

3.6 Future Work

We have mentioned in the chapter that an HMD would provide a way of providing proprioceptive alignment over the elevation and cyclotorsion angles, e and c respectively. Some questions remain however about how exactly to control the viewing camera. As path A in Figure 3.7 shows, if we were to copy the operator's head orientation onto elevation e while maintaining the end-effector in view, the operator would feel that lowering his head

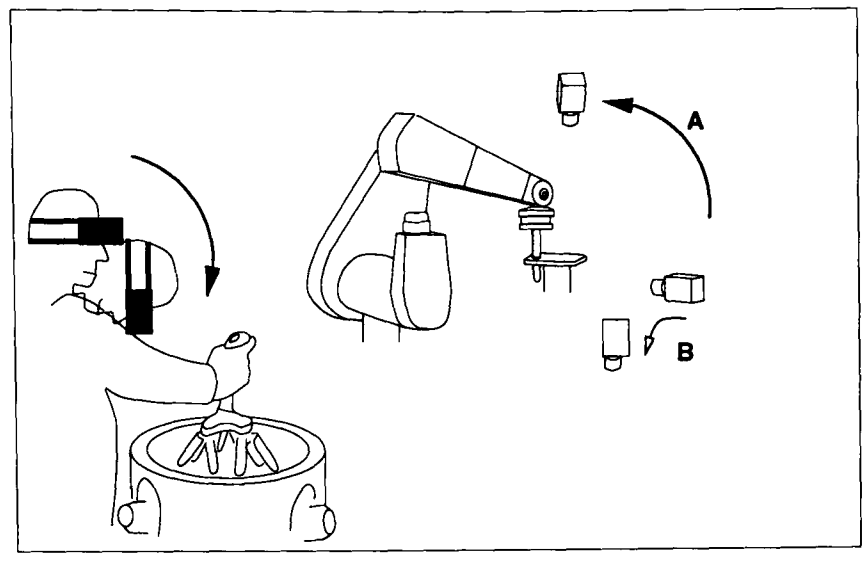


Figure 3.7: If in response to the operator's head motion the virtual camera takes path A, the operator will feel that lowering the head is causing the camera to rise. If path B is taken, then some other operator motion will have to be exploited to indicate "looking over".

introduces an upwards translation. However, as path B indicates, if the operator's radius of rotation is copied, some other operator movement would have to provide a signal for translation.

A further question is whether a stereoscopic presentation of data would be of value (note that this only involves stereo graphics, not stereo vision).

The software issues are two-fold. First is that mentioned already. It would be valuable to use a more sophistical FSM design. The second concerns the lack of tools available to assist in splitting the Graphical User Interface from the processing, particularly in programmes where considerable volumes of data (images, graphics) need exchanging.

The traditional architecture for computer vision programs with GUIs is to build a single process unit linking the computer vision program with the GUI libraries to provide operator feedback such as graphical image overlays with the minimum of fuss. Indeed by hardwiring GUI functions into the computer vision program this amounts to about 40% of the actual program code. Our implementation by providing a separate GUI process has had to partition the tracker and GUI into two. This, however, is not ideal because so much visual information needs to be communicated to the GUI particularly when debugging a program. We have thus developed a virtual GUI C++ class which the tracker uses to create an object to store quite primitive graphical overlays such as points, lines and ellipses. The GUI then accesses this virtual GUI object through a shared memory segment and uses its specific proprietary function calls to display the virtual overlay information. While this

goes some way to easing the implementation of coding a separate GUI and tracking process we would rather have used existing program code as a template for richer objects. The separation of GUI from the computer vision program would solve at least three problems. i) GUI libraries are in a state of flux at the minute and when the computer vision code is tied to a particular library which has stopped being developed it quickly looks dated and can not keep pace with current technology — Target Junior [83] being a prime example because their GUI of choice InterViews has stopped being supported. ii) The code would be more portable. It is generally the input and output of a program which is hardest to port, the input generally being the device drivers and the output being the GUI. By separating the design of the GUI away from the core program, porting would then consist of rewriting the device drivers if they were not supported by the operating system, and rewriting the GUI leaving the all important computer vision program intact. iii) Commercial GUIs libraries generally have better support and are quicker to design than free GUI libraries. However it would be very easy to substitute a commercially restrictive GUI for a free licensing GUI when the product ships.

3.7 Conclusions

This chapter has discussed the overall architectural and systems issues involved in the addition of a virtual viewing system to the Oxford force-reflection teleoperation system.

The chapter began by giving a brief overview of the salient features of the existing force-reflecting system of Daniel, Fischer and McAree, with the aim of providing context for the present work. It continued by discussing the notion of proprioceptive alignment, particularly of visual and tactile frames centred on the head and hand. No method of visual presentation appears ideally suited, and we chose a compromise that provides reasonable alignment but without encumbering the operator.

The chapter outlined the overall system architecture, and described how the vision system could readily adopt and adapt the successful features incorporated into the force-reflection system, namely the use of multiple communicating processes running on a genuine real-time operating system, and the design and coding of the system as a finite state machine.

4

Calibration

Calibration of the extrinsic and intrinsic parameters of the cameras viewing the workspace of a robot arm for use in teleoperation is a necessary step in the recovery from their images of metric information about the robot and its surroundings needed in the rest of this thesis.

In this chapter we develop an algorithm that uses the robot slave manipulator to generate the calibration pattern of points in three dimensions which define the world frame. Using the robot in this way has two primary advantages. First, the robot frame and world frame are united, and there is no need separately to discover the transformation between them. Secondly, the method is autonomous, and can be carried out as a background task without the need to introduce special calibration equipment.

However, it must not be assumed that the robot can generate perfectly accurate world positions. The principal aim of this chapter is to devise a model that respects the physical attributes of the robot arm and its linkages and hence accounts for the systematic progressive deviations from the ideal. 3D positions returned by the robot (by recording joint angles and transforming via the forward kinematics) are used, along with their associated image positions, as measurements in a maximum likelihood estimation of the calibration parameters. The 3D position measurements have a covariance that is a function of position and encodes a coherence length over which relative position measurements are accurate.

Experiments are presented which compare the calibration results with the camera parameters obtained using a conventional calibration grid.

4.1 Introduction

While there are tasks in autonomous hand-eye coordination for which uncalibrated visual techniques are quite adequate (eg. [25, 84]), vision processing for a teleoperation system has a rather different aim — that of presenting results to a human operator who is attuned to looking at a Euclidean world. To achieve Euclidean reconstruction, the cameras must first be calibrated.

The two major classes of camera calibration in the literature appear to lie on either side of that required when a robot arm is available to point to the 3D positions in the world frame.

To one side are the "test-field" calibration methods, of which the work of Tsai is an early representative [85], and in which a geometric pattern of perfectly known 3D points is placed in front of the camera and defines the Euclidean world frame. From measurements of the associated image projections, the projection matrix can be recovered, which in turn can be decomposed to recover the rotation and translation between camera and world frames (the extrinsic parameters) and the upper triangular matrix containing focal lengths, principal point and skew (the intrinsic parameters). Unfortunately, a robot arm will typically not be able to generate perfect positions, either absolutely or relatively, particularly when its movements span a large volume.

On the other side are the methods which avoid special calibration objects, and assume that the 3D scene is unknown. Such auto- or self-calibration techniques [86] can be used to infer both the scene positions and the camera calibration parameters from image correspondences. However, the most general methods have multiple solutions and are the least numerically stable [87–90]. A number of authors have considered self-calibration under constraints of special motion or special scenes [91–95] and, considering the comparative accuracy of calibration methods, all urge the inclusion of prior information whenever it is available [96].

The approach developed here is neither to include the 3D positions as absolute known

quantities, nor to attempt to recover them *ab initio*. Rather we include the values recovered from the robot's forward kinematics as measurements along with their corresponding image positions in each camera, and then allow their values to be optimized in a bundle adjustment.

Lavest *et al* [97] take a broadly similar optimization approach, but they did not use a robot and were able to assume that *all* the measurements are corrupted by random noise. While the positions of the 2D image points can be taken as independent Gaussian distributions, measurements of the 3D positions of the robot's tip derived from the robot's joint angles and the forward kinematic transformation have a more complicated error model. As discussed later, the nature of kinematic calibration of robot arms means that they suffer from spatially correlated bias. Here we develop and apply a model of the error in the 3D points as a zero-mean Gaussian distribution with covariance which depends on the proximity of points. If two points are close spatially, their errors are expected to be very similar, and, if far apart, to be un-correlated.

In the absence of a prior distribution for the camera parameters, their optimal value is the Maximum Likelihood Estimate, conditioned on error models just described, and here the MLE is found using the Levenberg-Marquardt algorithm.

Below, Section 4.3 details the noise models used for image and world data en route to a maximum likelihood estimation of the camera parameters. Section 4.4 describes the implementation using the Levenberg-Marquardt algorithm. Then, in Section 4.5, we describe implementation and experimental work with a calibration tool attached to the end-effector of the Oxford teleoperation arm. We compare the results with camera parameters obtained using a conventional calibration grid. But first, Section 4.2 defines the problem domain and introduces the nomenclature and notation used in this chapter.

4.2 Calibration with scene point adjustment

Without imposing priors, we wish to estimate the set of camera parameters $\{\mathbf{c}\} = \{\mathbf{c}_1, \mathbf{c}_2, \dots\}$ for several projective cameras arranged around the robot workspace using measurements of world points $\mathbf{X}$ and their projected positions in the camera images, $\mathbf{x} = \mathtt{P}_j\mathbf{X}$, where $\mathtt{P}_j$ is a function of the $\mathbf{c}_j$ for the jth camera.

Specifying the set of scene points $\{\mathbf{X}\}$ in a Euclidean frame allows the projection matrix for each camera to be separated as $\mathtt{P} = \mathtt{K}[\mathtt{R}|\mathtt{t}]$, where the matrix of intrinsic parameters

$$\mathtt{K} = \begin{bmatrix} f_x & s & u_o \\ 0 & f_y & v_o \\ 0 & 0 & 1 \end{bmatrix}$$

depends on the focal lengths f in the x and y directions, skew s, and principal point (u_o, v_o); and where $\mathtt{R}$ and $\mathtt{t}$ are the extrinsic rotation matrix and translation between world and camera frames. Here, we represent the rotation in terms of an angle-axis 3-vector, and assume that the skew is zero. The vector of ten parameters for camera i is

$$\mathbf{c}_j = [f_x \; f_y \; u_0 \; v_0 \; a_x \; a_y \; a_z \; t_x \; t_y \; t_z]_j^\top \; .$$

Because we use multiple cameras, we have sufficient constraint to allow the optimization procedure to estimate the set of world points $\{\hat{\mathbf{X}}\}$, and so for I points and J cameras and there are $3I + 10J$ parameters to be found.

A compact and reasonable description for image and world measurements is that of a multi-variate Gaussian distribution with mean centered on the measurement set $\{\mathbf{x}, \mathbf{X}\}$ and with covariance matrix $\Sigma_{\{\mathbf{x},\mathbf{X}\}}$. The parameter vector is $\{\mathbf{c}, \hat{\mathbf{X}}\}$, and the estimated measurement vector is $\{\hat{\mathbf{x}}, \hat{\mathbf{X}}\}$. The vector of functions mapping $\{\mathbf{c}, \hat{\mathbf{X}}\}$ to $\{\hat{\mathbf{x}}, \hat{\mathbf{X}}\}$ is

$$\mathbf{f} : \{\mathbf{c}, \hat{\mathbf{X}}\} \mapsto \{\hat{\mathbf{x}}, \hat{\mathbf{X}}\} \; .$$

The maximum likelihood estimate of the parameter set is found as the maximum of the joint probability of the data given their interpretation:

$$\arg\max_{\{\mathbf{c},\hat{\mathbf{X}}\}} \mathrm{p}(\{\mathbf{x}, \mathbf{X}\}|\{\mathbf{c}, \hat{\mathbf{X}}\}) \quad .$$

Because we have assumed Gaussian noise distributions, the joint probability takes on the usual d-dimensional multivariate form

$$\mathrm{p}(\{\mathbf{x}, \mathbf{X}\}|\{\mathbf{c}, \hat{\mathbf{X}}\}) = \frac{1}{(2\pi)^{d/2}|\Sigma_{\{\mathbf{x},\mathbf{X}\}}|^{1/2}} \; \exp\left(-\frac{\epsilon^\top \Sigma_{\{\mathbf{x},\mathbf{X}\}}^{-1} \epsilon}{2}\right)$$

where

$$\epsilon = \{\mathbf{x}, \mathbf{X}\} - \mathbf{f}(\{\mathbf{c}, \hat{\mathbf{X}}\}) = \{\mathbf{x}, \mathbf{X}\} - \{\hat{\mathbf{x}}, \hat{\mathbf{X}}\}$$

is the difference between the measurements and the predicted measurements. It is conventional and convenient to minimize the negative logarithm of the joint probability

$$\arg\min_{\{\mathbf{c},\hat{\mathbf{X}}\}} -\ln \mathrm{p}\left(\{\mathbf{x},\mathbf{X}\}|\{\mathbf{c},\hat{\mathbf{X}}\}\right) .$$

which, given that the normalizing denominator is a constant, is equivalent to minimizing the squared Mahalanobis distance (Bishop [98])

$$\arg\min_{\{\mathbf{c},\hat{\mathbf{x}}\}} \boldsymbol{\epsilon}^{\top}\Sigma^{-1}_{\{\mathbf{x},\mathbf{X}\}}\boldsymbol{\epsilon} . \tag{4.1}$$

The specification of the functions $\mathbf{f}$ is quite straightforward, and the principal issue of interest now is how to describe the measurement covariance matrix $\Sigma_{\{\mathbf{x},\mathbf{X}\}}$.

4.3 Models of measurement error

Recall that we use as measurements *both* the image positions $\mathbf{x}$ of the 3D points $\mathbf{X}$, *and* the 3D positions $\mathbf{X}$ themselves. The latter are derived from the robot arm's joint angles and its forward kinematics. Both sets of measurements are assumed to be Gaussian distribution and are thus fully specified by mean vectors and covariance matrices. The errors in the image and world measurements are uncorrelated, so that the overall covariance matrix is block diagonal

$$\Sigma_{\{\mathbf{x},\mathbf{X}\}} = \begin{bmatrix} \Sigma_{\mathbf{x}} & 0 \\ 0 & \Sigma_{\mathbf{X}} \end{bmatrix} .$$

4.3.1 Image point measurement model

Amongst themselves, the image measurements are independent and hence uncorrelated, and so the covariance $\Sigma_{\mathbf{x}}$ is simply a diagonal matrix with elements equal to σ_x^2 and σ_y^2 — the variance of the noise in the image x and y directions. Later experiments on locating the robot end effector in images show that $\sigma_x \sim \sigma_y \sim 0.1$ pixel.

4.3.2 World point measurement model

Deriving the covariance matrix $\Sigma_{\mathbf{X}}$ for world point measurements is more involved.

The position and orientation in the world frame of the end-effector of the robot arm is determined from measurement of the six joint angles $\boldsymbol{\theta}$ and prior knowledge of the robot's

forward kinematics. This geometrical information is encoded in a standard way using Denavit-Hartenberg parameters — in our case the 6-vector $\mathbf{a}$ of distances between neighbouring joint axes measured along their mutual normal, the 6-vector $\boldsymbol{\alpha}$ of skew angles between neighbouring joint axes, and the 6-vector $\mathbf{s}$ of offsets between the joint and the where the mutual normal intersects a joint axis. The parameters are measured for a particular arm during kinematic calibration which typically consists of measuring the pose of the arm's end-effector at a number of distinct configurational points for a set of known joint angles. A nonlinear iterative least-squares procedure is then used to estimate the parameters over the entire range of arm motions. A survey is given by Hollerbach and Wampler in [99].

A problem with the resulting calibration is that it has been obtained using many sources of error. Figure 4.1 shows a taxonomy of possible error sources. For example the Puma 500 Mark 3 equipment manual [100] reports an angular tolerance of $\pm 0.000375°$ and a machining length tolerance of ± 0.127mm which correspond to the error sources highlighted in red in Figure 4.1. These kinematic sources of error can be propagated into an end-effector accuracy of 0.3mm (Conrad *et al.* [101]).

Other sources of error, in particular those due to dynamic and structural uncertainties, are often not modelled. Dynamic errors may be included in the control law to achieve faster dynamic robot response and structural uncertainties may be included in both dynamic and kinematic manipulator models. The structural error sources can be split into static friction present in the bearings and gears (gear eccentricity, cogging, stick-shift) and warping of the manipulator due to load and temperature variations. These error sources are local properties and are poorly identifiable over the entire range of arm motions.

An alternative approach then might seem to be to calibrate piecewise by computing parameters in sub-volumes of the workcell. However, the ill-conditioned nature of the overall calibration problem means that the principal geometric parameters would then be poorly recovered.

Given that practical calibration must excite the parameters over as large portion of the robot workspace as possible, and does not typically include extra "subtle" model parameters, the errors introduced into the end effector positions are modelling errors which are effectively spatially correlated biases rather than noise.

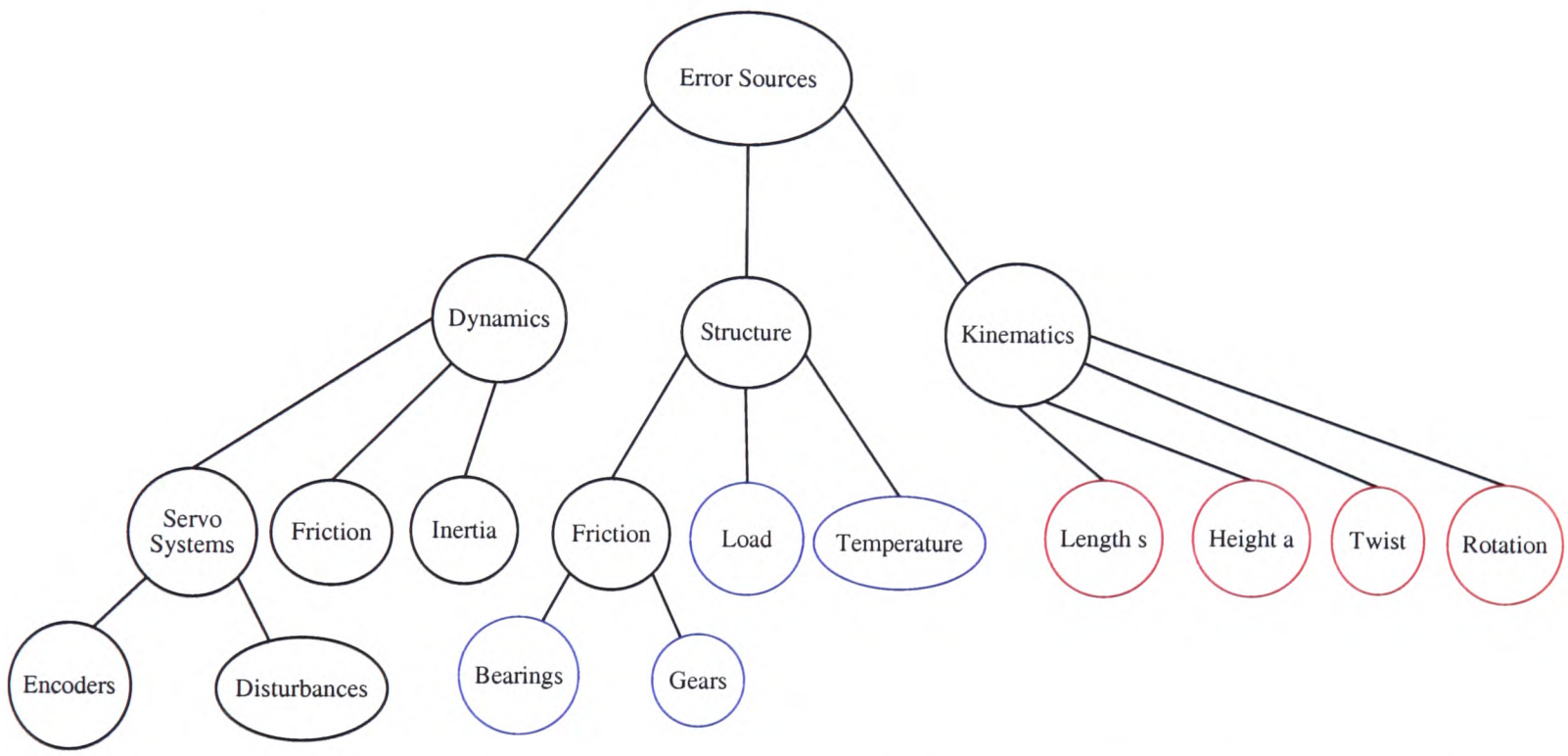


Figure 4.1: An end-effector error source tree for robot manipulators. The red leaves represent static kinematic error sources, the Puma 500 Mark 3 equipment manual [100] reports an angular tolerance of $\pm 0.000375°$ and a machining length tolerance of ± 0.127mm. The blue leaves represent structural errors which are local properties and are often unmodelled. The dynamic error sources should be accounted for in the control law to achieve faster dynamic robot response, however in this chapter measurements are taken once the robot is stationary thus dynamic errors are ignored.

Some feel for the size of these effects can be obtained by looking at the difference between the joint angles determined geometrically to place the end effector at some particular position, and the actual joint angles required to achieve that position. There are three terms of note (Daniels [102]):

$$\Delta\theta_i \sim \Delta\theta_i^{\mathrm{O}} + \Delta\theta_i^{\mathrm{G}} + \Delta\theta_i^{\mathrm{D}} \quad .$$

$\Delta\theta_i^{\mathrm{O}}$ accounts for the offset of the zero encoder position from the absolute zero position. This can be as large as 10°, but most robots remove this offset when initializing and it can be neglected. The second term, $\Delta\theta_i^{\mathrm{G}} \sim 0.01°$, arises due to structural error sources and is the bias present because gravity loading of the end-effector causes bending of the rigid links and stiffness in the bearings and is position dependent. The next most significant source of bias arises from manufacturing defects giving, for example, non-parallel joints. This bias is stated in the Puma 500 Mark 3 equipment manual as $\Delta\theta_i^{\mathrm{D}} \sim 0.000375°$.

Coherence length

We introduce a *coherence length* λ to describe the distance the end-effector needs to travel before the local biases become substantially de-correlated, and assume that beyond 2λ they can be treated as stochastic. An empirical coherence length of 0.5m is chosen, which is the length of path the robot needs to achieve, because of the singularities and gear ratios etc., so that the robot's starting position and end position become de-correlated. We now consider the derivation of the covariance of the end-effector position separately for the two regimes.

Well-separated points: $\ell > 2\lambda$

Given the joint angles and DH forward kinematic parameters, any point $\mathbf{X}^E$ defined in the end-effector frame is transformed into the world frame as

$$\begin{aligned} \mathbf{X} &= \mathbf{F}(\boldsymbol{\theta}, \boldsymbol{\alpha}, \mathbf{a}, \mathbf{s}, \mathbf{X}^E) \\ &= \mathbf{F}(\Theta, \mathbf{X}^E) \end{aligned}$$

where $\mathbf{F}$ is a vector function and Θ is the complete set of 24 joint angles and DH parameters.

For points separated by more than the twice the coherence length, errors resulting from using a single overall kinematic configuration are random. Assuming that the errors in the DH parameters are independent, and independent from those in the joint angles, the covariance of Θ is

$$\Sigma_\Theta = \begin{bmatrix} \Sigma_{\boldsymbol{\theta}} & 0 & 0 & 0 \\ 0 & \Sigma_{\boldsymbol{\alpha}} & 0 & 0 \\ 0 & 0 & \Sigma_{\mathbf{a}} & 0 \\ 0 & 0 & 0 & \Sigma_{\mathbf{s}} \end{bmatrix}.$$

where typical values for the elements of these diagonal matrices are

$$\Sigma_{\boldsymbol{\theta}_{ii}} = (0.01°)^2 \qquad \Sigma_{\boldsymbol{\alpha}_{ii}} = (0.01°)^2$$
$$\Sigma_{\mathbf{a}_{ii}} = (0.157\text{mm})^2 \quad \Sigma_{\mathbf{s}_{ii}} = (0.157\text{mm})^2$$

Consequently the covariance of the end effector point positions is

$$\Sigma_{\mathbf{X}} = \begin{bmatrix} \Sigma_{\mathbf{X}_{11}} & 0 & \cdots \\ 0 & \Sigma_{\mathbf{X}_{22}} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

where

$$\Sigma_{\mathbf{X}_{ii}} = \nabla\mathbf{F}\Sigma_\Theta\nabla\mathbf{F}^\top$$

and where $\nabla\mathbf{F} = [\partial\mathbf{F}/\partial\Theta]$ is the Jacobian matrix.

Proximate points: $\ell < 2\lambda$

Accurate camera calibration requires an order of magnitude more measurement data than does kinematic calibration, and is thus very likely to involve proximate points.

The covariance of the end-effector point position

$$\Sigma_{\mathbf{X}} = \begin{bmatrix} \Sigma_{\mathbf{x}_{11}} & \mathrm{R}_{\mathbf{x}_1\mathbf{x}_2} & \cdots \\ \mathrm{R}_{\mathbf{x}_2\mathbf{x}_1} & \Sigma_{\mathbf{x}_{22}} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

has finite on-diagonal blocks as before, but also has finite off-diagonal entries. We propose an exponential correlation function, typical of a Gauss-Markov process

$$R(\ell) = k\exp(-|\ell|/\lambda)$$

(Fig. 4.2). The value of the coherence length is set at $\lambda = 0.5$ m, and the constant k is determined by equating the uncorrelated noise power with correlated noise power.

$$\int_{-\infty}^{\infty} \delta(\ell)\mathrm{d}\ell = \int_{-\infty}^{\infty} k \exp(-|\ell|/\lambda)\mathrm{d}\ell$$

giving

$$1 = 2k\lambda \;\Rightarrow\; k = \frac{1}{2\lambda} \quad \text{and} \quad R(\ell) = \frac{1}{2\lambda}\exp(-|\ell|/\lambda)$$

The cross-covariance blocks of the covariance matrix are thus

$$\mathbf{R}_{\mathbf{x}_i\mathbf{x}_j} = \begin{bmatrix} R(\ell)\sqrt{\Sigma_{\mathbf{x}_{ii00}}\Sigma_{\mathbf{x}_{jj00}}} & R(\ell)\sqrt{\Sigma_{\mathbf{x}_{ii01}}\Sigma_{\mathbf{x}_{jj01}}} & R(\ell)\sqrt{\Sigma_{\mathbf{x}_{ii02}}\Sigma_{\mathbf{x}_{jj02}}} \\ R(\ell)\sqrt{\Sigma_{\mathbf{x}_{ii10}}\Sigma_{\mathbf{x}_{jj10}}} & R(\ell)\sqrt{\Sigma_{\mathbf{x}_{ii11}}\Sigma_{\mathbf{x}_{jj11}}} & R(\ell)\sqrt{\Sigma_{\mathbf{x}_{ii12}}\Sigma_{\mathbf{x}_{jj12}}} \\ R(\ell)\sqrt{\Sigma_{\mathbf{x}_{ii20}}\Sigma_{\mathbf{x}_{jj20}}} & R(\ell)\sqrt{\Sigma_{\mathbf{x}_{ii21}}\Sigma_{\mathbf{x}_{jj21}}} & R(\ell)\sqrt{\Sigma_{\mathbf{x}_{ii22}}\Sigma_{\mathbf{x}_{jj22}}} \end{bmatrix}$$

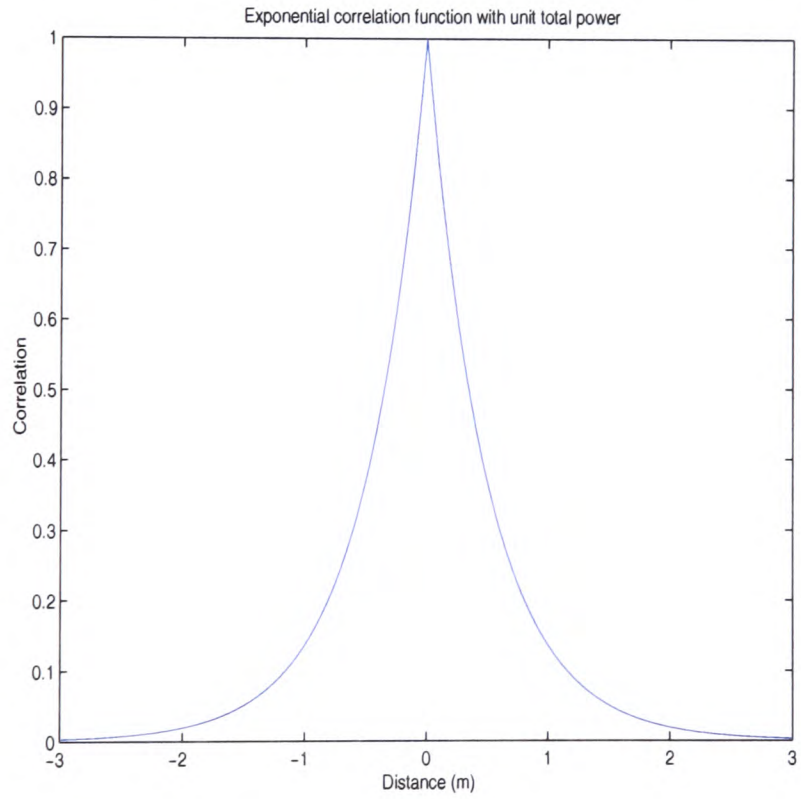


Figure 4.2: The proposed exponential correlation function is shown $R(\ell) = k\exp(-|\ell|/\lambda)$ with coherence length λ equal to 0.5 m.

4.4 Optimization procedure

To minimize the Mahalanobis distance in equation (4.1) we use a Levenberg-Marquardt (LM) method [103, 104], a restricted step optimization method which circumvents possible difficulties caused by non positive definite Hessian matrices in Newton's method. To ensure

rapid and successful convergence to the global minimum the initial parameter estimates should lie close to their final positions. The initial estimate of the world positions $\hat{\mathbf{X}}_o$ is chosen to be the measurements themselves, and the initial camera parameters are found using linear calibration of each camera separately using the measured 3D points as the correct values in a test field calibration.

4.4.1 Normalising measurement data

Both the linear calibration initialisation and the LM minimization benefit from normalizing or "centring" the data by making the mean zero and the variance equal to 2 for the 2D image data, and 3 for the 3D data. For linear calibration centring is essential for numerical accuracy, and for LM it is desirable for faster convergence [105].

Denoting the mean and standard deviation of x by $\bar{x}$ and σ_x and similarly for the other measurements the scaled image and world measurements are

$$\mathbf{x}' = \mathtt{S}_2\mathbf{x} = \begin{bmatrix} \sqrt{2}/\sigma_x & 0 & -\sqrt{2}/\sigma_x\bar{x} \\ 0 & \sqrt{2}/\sigma_y & -\sqrt{2}/\sigma_y\bar{y} \\ 0 & 0 & 1 \end{bmatrix} \mathbf{x}$$

and

$$\mathbf{X}' = \mathtt{S}_3\mathbf{X} = \begin{bmatrix} \sqrt{3}/\sigma_X & 0 & 0 & -\sqrt{3}/\sigma_X\bar{X} \\ 0 & \sqrt{3}/\sigma_Y & 0 & -\sqrt{3}/\sigma_Y\bar{Y} \\ 0 & 0 & \sqrt{3}/\sigma_Z & -\sqrt{3}/\sigma_Z\bar{Z} \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{X} .$$

The measurement covariances are transformed by taking the upper-left 2×2 submatrix of $\mathtt{S}_2$ to form $\mathtt{U}_{S2}$ and the upper-left 3×3 submatrix of $\mathtt{S}_3$ to form $\mathtt{U}_{S3}$. Transforming the measurement covariances

$$\Sigma_{\mathbf{x}'} = \mathtt{U}_{S2}\Sigma_{\mathbf{x}}\mathtt{U}_{S2}^{\top} ; \qquad \Sigma_{\mathbf{X}'} = \mathtt{U}_{S3}\Sigma_{\mathbf{X}}\mathtt{U}_{S3}^{\top} .$$

With normalized data, the recovered camera parameters are associated with the projection matrix

$$\mathtt{P}' = \mathtt{S}_2\mathtt{P}\mathtt{S}_3^{-1} ,$$

but we defer discussion of denormalising the parameters until Section 4.4.4.

Below we will assume normalised data, but to keep the notation uncluttered will not append primes to quantities.

4.4.2 Camera parameter initialisation

We obtain an estimate for the projection matrix $\mathtt{P}$ of each camera separately, given measurements of the image points and assuming absolute knowledge of the 3D points. We are not concerned about the nature of the measurement noise.

The projection equation

$$\gamma_i \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \mathtt{P}\mathbf{X}_i$$

can be rearranged as

$$\mathtt{M}_i\mathbf{p} = \begin{bmatrix} \mathbf{X}_i^\top & \mathbf{0}^\top & -x_i\mathbf{X}_i^\top \\ \mathbf{0}^\top & \mathbf{X}_i^\top & -y_i\mathbf{X}_i^\top \end{bmatrix} \mathbf{p} = \mathbf{0}\,, \tag{4.2}$$

where $\mathbf{p} = [P_{11}, P_{12}, P_{13}, P_{14}, P_{21}, \ldots, P_{34}]^\top$ Stacking n measurement equations to form an over-constrained equation set (with $n \geq 5\frac{1}{2}$ image/world correspondances)

$$\begin{bmatrix} \mathtt{M}_1 \\ \mathtt{M}_2 \\ \vdots \\ \mathtt{M}_n \end{bmatrix} \mathbf{p} = \mathbf{0} \quad .$$

$\mathtt{M}_i$ must be rank 11 and the solution for $\mathbf{p}$ is found, subject to the constraint that $|\mathbf{p}| = 1$, as the unit singular vector corresponding to the smallest singular value in the SVD of $\mathtt{M}$ (eg [105]). (ie., $\mathbf{p}$ is the right null-space of $\mathtt{M}_i$.)

The inverse of the left 3×3 submatrix of $\mathtt{P}$ has the form $\mathtt{R}^{-1}\mathtt{K}^{-1}$, the product of a rotation matrix and an upper-triangular matrix, and hence recoverable by QR decomposition. (In QR decomposition, R is the upper triangular matrix — so $\mathtt{K} \leftarrow \mathrm{R}^{-1}$ and $\mathtt{R} \leftarrow \mathrm{Q}^{-1}$.) Because $\mathbf{p}$ is only recovered up to scale, $\mathtt{K}$ requires rescaling so that $K_{33} = 1$. The translation $\mathbf{t} = \mathtt{K}^{-1}\mathbf{p}$, using $\mathtt{K}^{-1}$ before rescaling (or R directly from the QR decomposition).

Finally the rotation matrix is converted into angle-axis form, and the initial estimate of the camera parameters $\mathbf{c}$ is complete. These steps are summarised in Figure 4.3.

4.4.3 Camera parameter and world point optimization

The Levenberg-Marquardt method is used to minimize the Mahalanobis distance $\epsilon^\top \Sigma^{-1}_{\{\mathbf{x},\mathbf{X}\}}\epsilon$ where $\epsilon = \{\mathbf{x}, \mathbf{X}\} - \{\hat{\mathbf{x}}, \hat{\mathbf{X}}\}$, the error between estimated and actual measurements. As

Objective Given $n \geq 6$ world and image correspondences for each camera ($\mathbf{x}_i^{\text{im}} \leftrightarrow \mathbf{X}_i^{\text{wo}}$), determine the initial camera parameter estimate using linear methods.
Algorithm

1. Normalise the image and world data to their scaled form

$$\mathbf{X}^{\text{swo}} = \mathbf{S}_3\mathbf{X}^{\text{wo}} \quad \text{and} \quad \mathbf{x}^{\text{sim}} = \mathbf{S}_2\mathbf{x}^{\text{im}} \quad .$$

   The transformations are defined in section 4.4.1.

2. Form the $2n \times 12$ matrix $\mathtt{M}$ by stacking the equations (4.2) for each correspondence $\mathbf{x}_i^{\text{im}} \leftrightarrow \mathbf{X}_i^{\text{wo}}$. The solution of $\mathtt{Mp} = 0$, subject to $||\mathbf{p}|| = 1$, is obtained by taking the unit singular vector of $\mathtt{M}$ with the smallest singular value and forms the 3x4 projective camera matrix $\mathtt{P}$.

3. The camera projection matrix is decomposed into projective camera parameters taking care to align the coordinate frames (Appendix A.2.2) $\mathtt{P} \rightarrow \mathbf{p}_{\text{proj}}$.

Figure 4.3: The monocular linear camera calibration algorithm used to obtain camera parameter estimates to seed the nonlinear camera calibration routine.

mentioned earlier, as well as optimizing the camera parameters, we optimize the position of the world points, using the robot measurements as their initial values.

Rather than solving the normal equations of the Newton iteration

$$[\mathbf{J}^\top \Sigma_{\{\mathbf{x},\mathbf{X}\}}^{-1} \mathbf{J}]\, \Delta = \mathbf{J}^\top \Sigma_{\{\mathbf{x},\mathbf{X}\}}^{-1} \epsilon$$

for the parameter innovation vector, Δ, each LM iteration pre-multiplies the on-diagonal elements of the curvature matrix $[\cdot]$ on the left by $(1+\lambda)$. For $\lambda \ll 1$ the method tends to the Newton method, but for $\lambda \gg 1$ the method tends to pure gradient descent. The resulting parameter innovation Δ is added to the current parameter estimate and the Mahalanobis squared distance recalculated. If the new error is greater than the old error then the old parameter estimate is retained and λ is increased by a factor of 10 for the next iteration. Conversely, if the new error is less than the old error then the parameter estimate is updated and λ is diminished by a factor of 10.

If the cost has not changed in the last six iterations the minimisation is judged to have converged, at which point the final covariance matrix is calculated from

$$\Sigma_{\{\mathbf{c},\hat{\mathbf{x}}\}} = [\mathbf{J}^\top \Sigma_{\{\mathbf{x},\mathbf{X}\}}^{-1} \mathbf{J}]^{-1} \, .$$

In the normal equations, the Jacobian matrix J is partitioned as

$$\mathtt{J} = \begin{bmatrix} \mathtt{J}_{\mathbf{xc}} & \mathtt{J}_{\mathbf{x}\hat{\mathbf{X}}} \\ \mathtt{J}_{\mathbf{Xc}} & \mathtt{J}_{\mathbf{X}\hat{\mathbf{X}}} \end{bmatrix} ,$$

where

$$\begin{aligned} \mathtt{J}_{\mathbf{xc}} &= [\partial\hat{\mathbf{x}}/\partial\mathbf{c}] & \mathtt{J}_{\mathbf{x}\hat{\mathbf{X}}} &= [\partial\hat{\mathbf{x}}/\partial\hat{\mathbf{X}}] \\ \mathtt{J}_{\mathbf{Xc}} &= [\partial\hat{\mathbf{X}}/\partial\mathbf{c}] = 0 & \mathtt{J}_{\mathbf{X}\hat{\mathbf{X}}} &= [\partial\hat{\mathbf{X}}/\partial\hat{\mathbf{X}}] = \mathtt{I} \ . \end{aligned}$$

It is thus very sparse and the computational speed of the LM algorithm can be increased by employing sparse matrix methods for the inversion of the normal equations.

4.4.4 De-normalising the optinized camera parameters

The parameters and parameter covariances are recovered in normalised measurement space.

For each camera, the normalised camera calibration matrix $\mathtt{P}'$ is calculated from the normalised camera parameters. Its unnormalised counterpart is found from

$$\mathtt{P} = \mathtt{S}_2^{-1}\mathtt{P}'\mathtt{S}_3 \ ,$$

and decomposed using the QR method to find the camera parameters explicitly.

This procedure effectively defines a numerical function $\mathbf{c} = \mathbf{c}(\mathbf{c}')$ from which we can compute a Jacobian matrix $[\partial\mathbf{c}/\partial\mathbf{c}']$. The unnormalised covariance matrix of $\mathbf{c}$ is then found as

$$\Sigma_{\mathbf{c}} = \left[\frac{\partial\mathbf{c}}{\partial\mathbf{c}'}\right] \Sigma_{\mathbf{c}'} \left[\frac{\partial\mathbf{c}}{\partial\mathbf{c}'}\right]^{\top} \ .$$

For simplicity we run the LM algorithm on the final unnormalised parameter set without iterating to obtain the unnormalised covariance parameters.

4.5 Implementation on the robot arm

The camera calibration method has been implemented using the slave Puma arm of the Oxford teleoperation system. Two practical issues of particular importance are, first, the accuracy of location of the end-effector in the images where the cameras have quite different viewpoints and, secondly, the robustness of location. The issues are not entirely separable, but the second is the more important. Outlying data resulting from gross mislocation of the end effector can have a disastrous effect on the calibration. A summary of the implementation is given in Figure 4.9.

Objective Using the linear normalised camera parameter estimate and world point estimates minimise the square of the Mahalanobis distance using the Levenberg-Marquardt algorithm to obtain the optimal camera and world point parameters. These steps are summarised in Figure 4.4.

Algorithm

1. **Initialise** the Levenberg-Marquardt algorithm.

   (a) Prerequisites: The vector of measurements $\{\mathbf{x}, \mathbf{X}\}$ which are composed of the image and world measurements with covariance matrix $\Sigma_{\{\mathbf{x},\mathbf{X}\}}$, an initial estimate of the camera parameters $\mathbf{c}$ and the measured world points are used as the world point parameters $\hat{\mathbf{X}}$ and the function $f : \{\mathbf{c}, \hat{\mathbf{X}}\} \mapsto \{\hat{\mathbf{x}}, \hat{\mathbf{X}}\}$ (Section 4.4.3).

   (b) The LM constant is initialised to $\lambda = 0.001$.

2. **Iterate** the Levenberg-Marquardt algorithm.

   (a) The LM Jacobian matrix J is formed from by computing the derivative matrices

   $$\mathtt{J}_{\mathbf{xc}} = [\delta\hat{\mathbf{x}}/\delta\mathbf{c}], \quad \mathtt{J}_{\mathbf{x}\hat{\mathbf{X}}} = [\delta\hat{\mathbf{x}}/\delta\hat{\mathbf{X}}], \quad \mathtt{J}_{\mathbf{Xc}} = [\delta\hat{\mathbf{X}}/\delta\mathbf{c}], \quad \mathtt{J}_{\mathbf{X}\hat{\mathbf{X}}} = [\delta\hat{\mathbf{X}}/\delta\hat{\mathbf{X}}]$$

   and concatenating

   $$\mathtt{J} = \begin{bmatrix} \mathtt{J}_{\mathbf{xc}} & \mathtt{J}_{\mathbf{x}\hat{\mathbf{X}}} \\ \mathtt{J}_{\mathbf{Xc}} & \mathtt{J}_{\mathbf{X}\hat{\mathbf{X}}} \end{bmatrix}$$

   (b) The updated innovation for the parameters is then evaluated, pre-multiplying the on-diagonal elements of the curvature matrix $[\cdot]$ on the left by $(1 + \lambda)$.

   $$\Delta = [\mathtt{J}^\top \Sigma^{-1}_{\{\mathbf{x},\mathbf{X}\}} \mathtt{J}\mathtt{D}]^{-1} \mathtt{J}^\top \boldsymbol{\epsilon}$$

   (c) The new squared Mahalanobis distance is evaluated by augmenting the parameter vector with the parameter innovations $\{\mathbf{c}_{n+1}, \hat{\mathbf{X}}_{n+1}\} = \{\mathbf{c}_n, \hat{\mathbf{X}}_n\} + \{\Delta_c, \Delta_w\}$ and calculating the updated measurement estimates $(\mathbf{X} - \hat{\mathbf{X}})^\top \Sigma_{\mathbf{X}}^{-1} (\mathbf{X} - \hat{\mathbf{X}})$. If the new error is greater than the old error, then revert to the old parameter values, increase the value of λ by a factor of 10, and try again. If the new error is less than the old error, then the new parameters are retained and λ is diminished by a factor of 10. If the cost has not changed significantly for six iterations then the algorithm moves onto the termination stage else we iterate.

3. **Terminate**

   (a) The unnormalised camera matrix and world points are obtained from their normalised forms from $\mathtt{P}_{\mathrm{IM}\to\mathrm{WO}} = \mathtt{D}_{\mathrm{IM}\to\mathrm{SIM}} \mathtt{P}_{\mathrm{SIM}\to\mathrm{SWO}} \mathtt{D}_{\mathrm{SWO}\to\mathrm{WO}}$ and $\mathbf{X}^{\mathrm{WO}} = \mathtt{D}_{\mathrm{WO}\to\mathrm{SWO}} \mathbf{X}^{\mathrm{SWO}}$.

   (b) The set of parameters $\mathbf{p}$ that minimises the square of the Mahalanobis distance $(\mathbf{X} - \hat{\mathbf{X}})^\top \Sigma_{\mathbf{X}}^{-1} (\mathbf{X} - \hat{\mathbf{X}})$ is now **output** along with the covariance of the estimated parameters $\Sigma_{\{\mathbf{c},\hat{\mathbf{x}}\}} = [\mathtt{J}^\top \Sigma^{-1}_{\{\mathbf{x},\mathbf{X}\}} \mathtt{J}]^{-1}$.

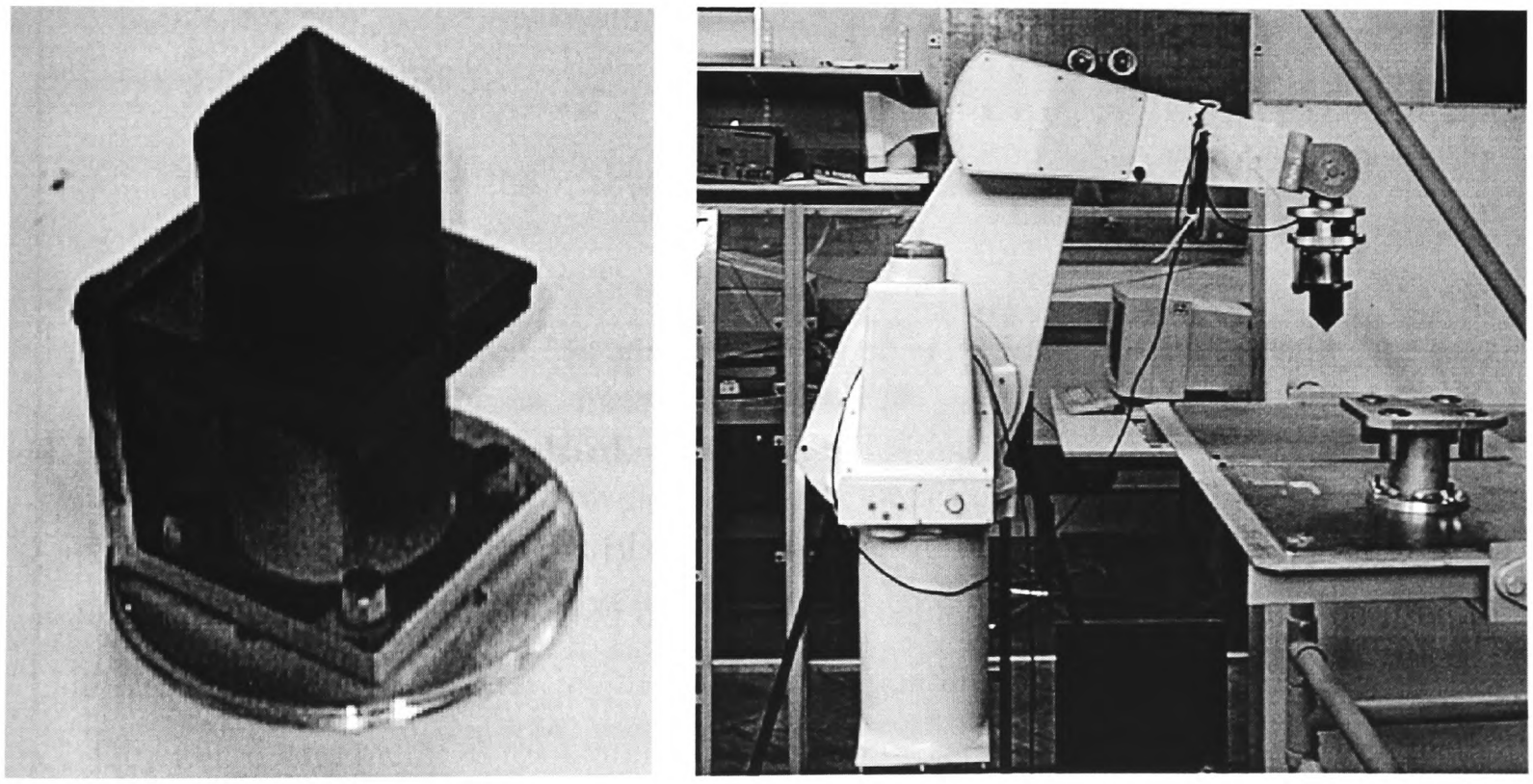
Figure 4.4: The Levenberg-Marquardt algorithm.

4.5.1 End-effector location

Our first implementation used a LED placed on the tip of the robot. Image location was poor when the LED is not pointing directly at the camera and thus could not be used to calibrate multiple cameras viewing the same corresponding point in space.

Other hardware methods to obtain robust image correspondances, by image based methods alone, were considered. They all required some form of novel hardware attachment and in the end we opted for robustness by the software engineering route.

A special cone-shaped tool was machined to assist in the calibration (Figure 4.5). Against a predominantly light background in the laboratory, painting it black provides good image contrast for feature detection. The tip of the cone is located in each image at the intersection of lines fitted to the extremal boundaries of the cone. The fitting uses a number of control points on the lines, whose likely locations in the images are predicted by projecting the estimate tool pose into the image. This model-based procedure requires a working calibration, which may seem a disadvantage but in fact provides safeguards against including mismatches.



(a) (b)

Figure 4.5: (a) The coned calibration tool, designed so that the tip is clearly visible from disparate views. (b) The calibration tool attached to the slave manipulator.

Using an estimate of the pose of the end-effector obtained from the robot, and a working

estimate or the camera calibration, our aim is to predict where in the image we should search for the extremal boundaries of the conical tip of the end-effector. We first determine the 3D lines giving rise to the extremal boundaries.

Referring to Figure 4.6, let O be the optic centre, the origin of camera coordinates, let $\mathbf{A} = OA$ be the tip of the cone, and let $\mathbf{B} = OB$ be the point on the axis level with where the cone turns into a cylinder, effectively the centre of the base of the cone. All these points are in the camera frame. If R is the radius of the base, then the cone half-angle Ψ is $\tan^{-1}(R/|\mathbf{B} - \mathbf{A}|)$. This geometry is known from the robot and the current extrinsic parameters and from prior measurement.

Construct $\mathbf{k}$ as the unit vector in the direction $\mathbf{B} - \mathbf{A}$. Then construct two mutually orthogonal unit vectors in the base: $\mathbf{j} = (\mathbf{k} \times \mathbf{A})/|\mathbf{k} \times \mathbf{A}|$ and $\mathbf{i} = \mathbf{j} \times \mathbf{k}$. For $0 \leq \theta < 2\pi$ the vector

$$\mathbf{d} = R(\cos\theta\mathbf{i} + \sin\theta\mathbf{j} - \cos\Psi\sin\Psi\mathbf{k})$$

touches the surface of the cone and is everwhere normal to it. For two values, $\pm\theta_P$ either side of $\mathbf{i}$, $\mathbf{d}$ will also be normal to the tangent planes through the optic centre. But $\mathbf{A}$ lies in both these planes, so that

$$\mathbf{A} \cdot \mathbf{d} = R\mathbf{A} \cdot (\cos\theta_P\mathbf{i} + \sin\theta_P\mathbf{j} - \cos\Psi\sin\Psi\mathbf{k}) = 0$$

But $\mathbf{A} \cdot \mathbf{j} = 0$, so that

$$\theta_P = \pm\cos^{-1}\left(\cos\Psi\sin\Psi\frac{\mathbf{A}\cdot\mathbf{k}}{\mathbf{A}\cdot\mathbf{i}}\right) \quad .$$

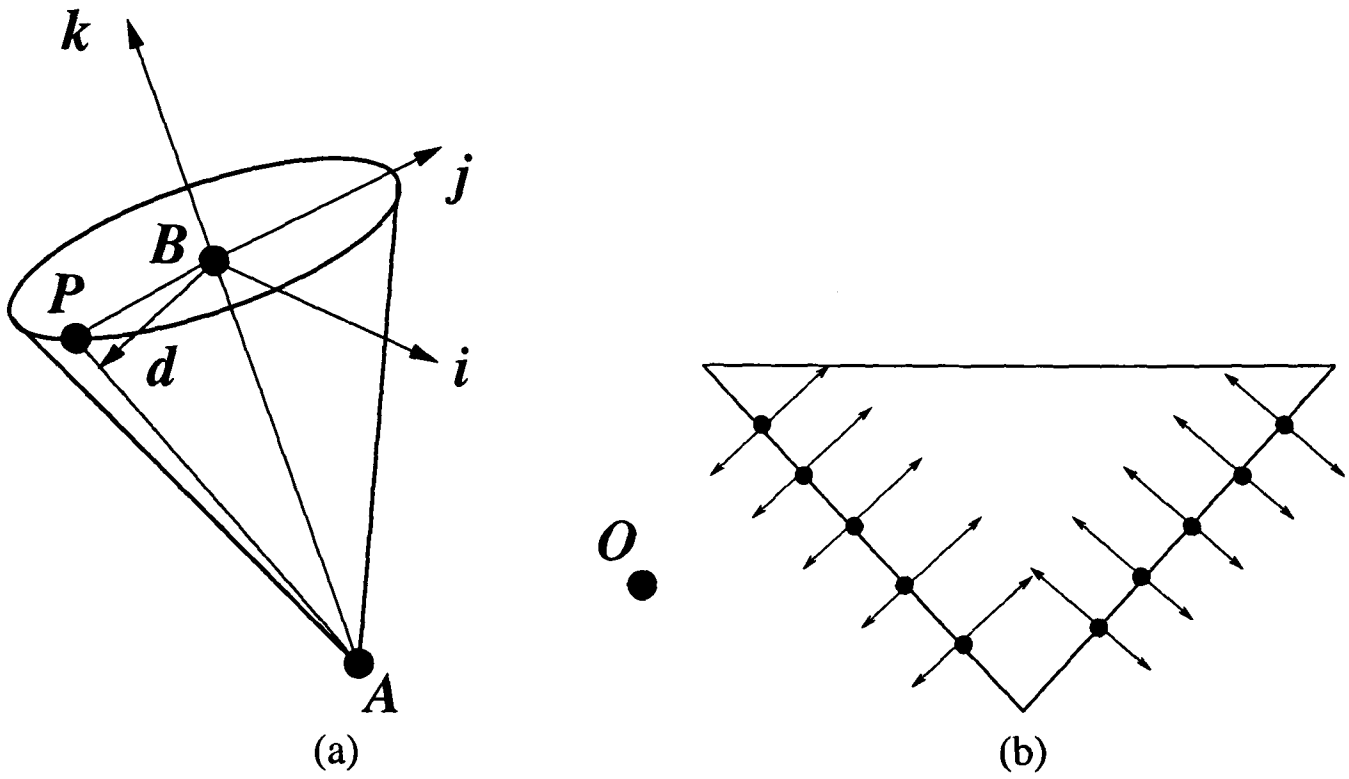
and the points $\mathbf{P}$ are found as

$$\mathbf{P} = \mathbf{B} + R(\cos\theta_P\mathbf{i} + \sin\theta_P\mathbf{j}) \quad .$$

Twenty one control points are positioned along each of the two profile lines

$$\mathbf{X}^{\text{cm}} = \mathbf{A} + \zeta(\mathbf{P} - \mathbf{A}) \quad , \quad \zeta = 0.20, 0.23, \ldots, 0.80$$

and are projected into the image using $\mathbf{x} = \mathtt{K}[\mathtt{I}|\mathtt{0}]\mathbf{X}^{\text{cm}}$. The function projecting each control point into the image is denoted by a vector function $\mathbf{k}(\mathbf{c})$ taking camera parameters as



(a) (b)

Figure 4.6: (a) The geometry of the points defining the profile edge of the tool, **A** lies at the apex of the cone and **B** lies further along the axis. (b) The tool profile, placement of control points and their confidence regions.

arguments. The dot product of the normal of the profile line, denoted by $[n_x\ n_y]$, is taken with the Jacobain of $\mathbf{k}$, $\nabla\mathbf{k}_c$ to obtain the projected Jacobian $\nabla\mathbf{k}_\perp$.

$$\nabla\mathbf{k}_\perp = [n_x\ n_y] \cdot \nabla\mathbf{k}_c \ .$$

From each projected control point the actual image edge is sought along a line of search perpendicular to the projected profile edge. The length of the region either side of the control point is $2\sigma_{cp}$ where

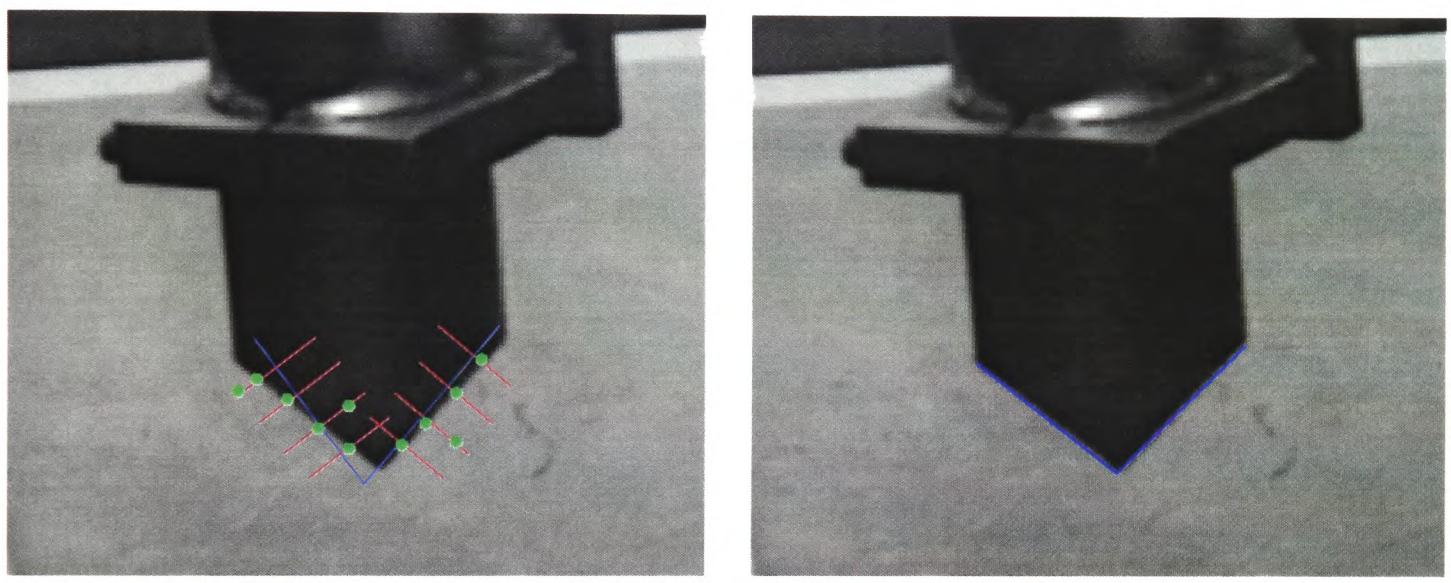
$$\sigma_{cp}^2 = \nabla\mathbf{k}_\perp \Sigma_c \nabla\mathbf{k}_\perp^\top + r^2 \quad .$$

The innovation uncertainty σ_{cp} is composed of the projection of camera uncertainty into the image along the control point search line and the control point measurement process uncertainty r. The schematic of the tool profile, placement of the projected control points and their confidence regions are shown in Figure 4.6.

Edge feature detection and localization involves convolving the 1D image array with a Gaussian of standard deviation of $\sigma = 3.0$, detecting the most significant peak in the finite difference signal, and fitting a quadratic to the strengths of the peak and its two neighbours. Straight lines are fitted to the located peaks using RANSAC to detect outliers

and an orthogonal regression routine to fit the final set of inliers. The intersection of the two profile lines is taken as the location of the tip. Figure 4.8 shows the results of 2000 measurement trials of the static tip. The measured standard deviation was $\sigma_x = 0.094$ and $\sigma_y = 0.124$. The difference is not significant, and we take the image measurement covariance to be

$$\Sigma_{\mathbf{x}} = \begin{bmatrix} 0.012 & 0.000 \\ 0.000 & 0.012 \end{bmatrix}$$



(a) (b)

Figure 4.7: (a) The projection of the tool with a few control point search lines shown and the detected edgels. (b) The outlying edgels have been removed and orthogonally regressed lines fitted to the remaining edgel data.

4.5.2 Tip location robustness

The model-based method of tip location above requires a working calibration. This is achieved by boot-strapping a linear calibration separately in each camera from an initial minimum set of six well-spaced points, and pulling in further points until the calibrations are sufficiently good to ensure that further calibration can continue unsupervised.

Erroneous tooltip positions are removed by gating the measured positions with the predicted position using the innovation covariance as a bound.

With $\mathbf{A}$ measured in the camera frame, the predicted tip location is $\mathbf{a} = \mathtt{K[I|0]A}$. A vector function $\mathbf{k}^A(c)$ is defined for this projection taking camera parameters as arguments.

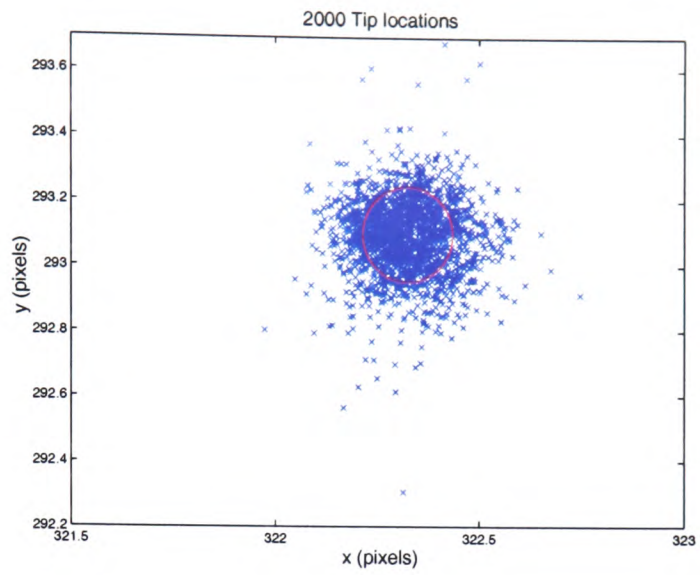


Figure 4.8: Following a run of 2000 sample points the standard deviation of the assumed Gaussian distribution is $\sigma_x = 0.094$ and $\sigma_y = 0.124$; an ellipsoid corresponding to these values overlays the data.

Its Jacobian $\nabla \mathbf{k}_c$ is used so that the innovation covariance can be defined as

$$\Sigma_{\mathbf{a}} = \nabla \mathbf{k}_c \Sigma_{\mathbf{c}} \nabla \mathbf{k}_c^{\top} + \Sigma_{\mathbf{x}}$$

Any measurements which are not in the ellipse centered around the predicted tip position are rejected.

4.6 Experiments

Experiments were conducted to examine the validity of the method proposed in this chapter. An initial baseline camera calibration was performed using a traditional Tsai tile to supply the calibration pattern, where the world points are known to a high degree of accuracy. Secondly, the experiment proper, used the slave manipulator to supply the calibration pattern. The camera parameters and recovered world positions were estimated using three different noise models for the world point measurements. Firstly the measured world points were assumed to be perfectly accurate by assuming zero noise. Secondly the world points were assumed to be independent and Gaussianly distributed (equivalent to zero coherence length) and thirdly the world points were assumed to be dependent and Gaussianly distributed with a coherence length of 0.5m.

The results compare the recovered camera parameters and world points.

1. **Initialise** camera parameters:

   (a) The operator manually moves robot to six world locations and identifies image location of tip.

   (b) The six correspondences ($\mathbf{x}^{\text{im}}_{1\leftrightarrow 6} \leftrightarrow \mathbf{X}^{\text{wo}}_{1\leftrightarrow 6}$) are passed to the linear calibration routine (Figure 4.3).

   (c) The initial calibration parameters obtained by the linear calibration method and the world and image correspondences are passed to the nonlinear calibration routine (Figure 4.4) the camera parameters and their covariances are output.

2. **Iterate** camera parameters:

   (a) The camera parameter estimates are used to project the profile edges of the tool into the image. The profile edge search mechanism is used to detect the actual tool tip (Section 4.5.1) to obtain a possible image tip measurement in the image.

   (b) Image measurements are gated.

   (c) This data collection is repeated for a large number of image points (typically 100).

   (d) All inlying corresponding points ($\mathbf{x}^{\text{im}}_i \leftrightarrow \mathbf{X}^{\text{wo}}_i$) are passed to the linear calibration routine (Figure 4.3).

   (e) Initial camera calibration parameters and the corresponding image and world points are passed to the nonlinear calibration routine (Figure 4.4).

   (f) The stack of input measurements are removed. The camera parameter estimates are either i) used to obtain more robust measurement data (goto **iterate** or ii) passed to the **output** of the algorithm.

Figure 4.9: Camera calibration implementation.

4.6.1 Baseline calibration

A Tsai tile (Figure 4.10) was individually placed around 1m in front of the three cameras overlooking the robot workcell. A single 768×576 image of the tile was taken. Using this image and the known 3D coordinates of the Tsai tile a nonlinear calibratino estimate was obtained. The uncertainty in this estimate was obtained by a Monte-Carlo simulation using ± 0.1 pixel error and a multiplicative world point error of 0.05%, ie. points close to the center of the tile had little covariance while points towards the periphery of the tile had an uncertainty of ± 0.5mm. The results of this calibration is summarised in Table 4.1.

Zero noise: Fixed Tsai tile world points

Camera	f_x	Δf_x	f_y	Δf_y	u_0	Δu_0	v_0	Δv_0
0	3153	30	3117	28	438	40	398	48
1	2784	28	2757	29	414	45	378	40
2	3234	40	3189	32	388	39	385	37

Zero noise: Fixed robot world points

Camera	f_x	Δf_x	f_y	Δf_y	u_0	Δu_0	v_0	Δv_0
0	3059	41	3116	41	563	78	281	82
1	2767	48	2690	48	443	66	146	97
2	3324	73	3373	80	123	149	283	151

Uncorrelated noise: Variable robot world points

Camera	f_x	Δf_x	f_y	Δf_y	u_0	Δu_0	v_0	Δv_0
0	3193	89	3192	91	575	188	298	202
1	2784	145	2730	166	488	180	279	279
2	3189	501	3223	499	224	704	308	1098

Correlated noise: Variable robot world points

Camera	f_x	Δf_x	f_y	Δf_y	u_0	Δu_0	v_0	Δv_0
0	3145	86	3137	86	579	181	285	191
1	2808	152	2747	163	374	189	356	295
2	3243	446	3271	440	218	664	251	1023

Table 4.1: The intrinsic camera parameters estimates and standard deviations.

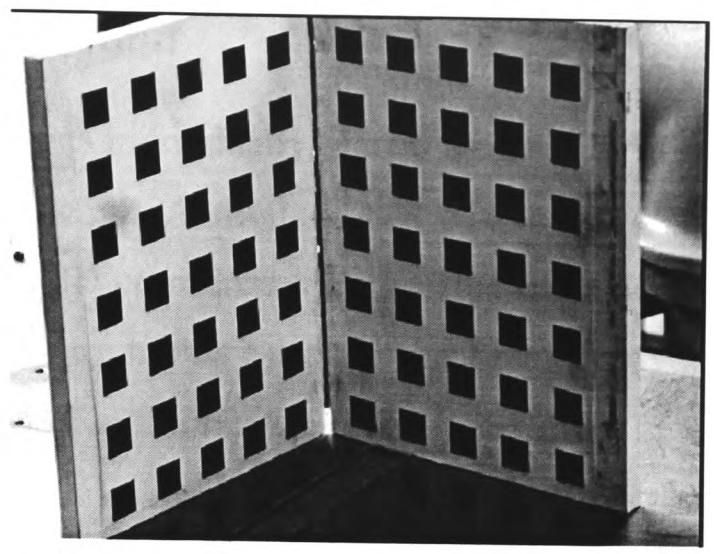


Figure 4.10: A Tsai tile was used to obtain the baseline camera calibration.

4.6.2 Manipulator camera calibration

The robot manipulator was guided to some eighty scene points of which the final fifty were visible in all three cameras. The initial thirty traced out the bounds of the robot cell which unfortunately were only visible from a single camera. Of the remaining fifty, twenty seven formed a tight cube in the middle of the robot workspace of dimensions 10cm $\times$ 10cm $\times$ 10cm for the purpose of examining the movement vectors of the recovered world positions which is are shown in the next section.

The camera intrinsic parameter results are tabulated in Table 4.1.

Figure 4.11 shows the estimated focal length against number of scene points for the first camera. What is interesting to note is the variation of the focal length for the fixed robot world point model indicating that this is indeed a poor description of the robot. The correlated robot world point model offers more consistent results and is similar to the uncorrelated robot world point model till around the 35th point. Here the correlated model suddenly deviates and levels out to a value which is comparable to the results obtained from the calibration obtained from a Tsai tile.

Figure 4.12(a-f) summarises the results for the focal length estimates where the above trends can also be seen.

The principal point estimates are shown in Figure 4.13.

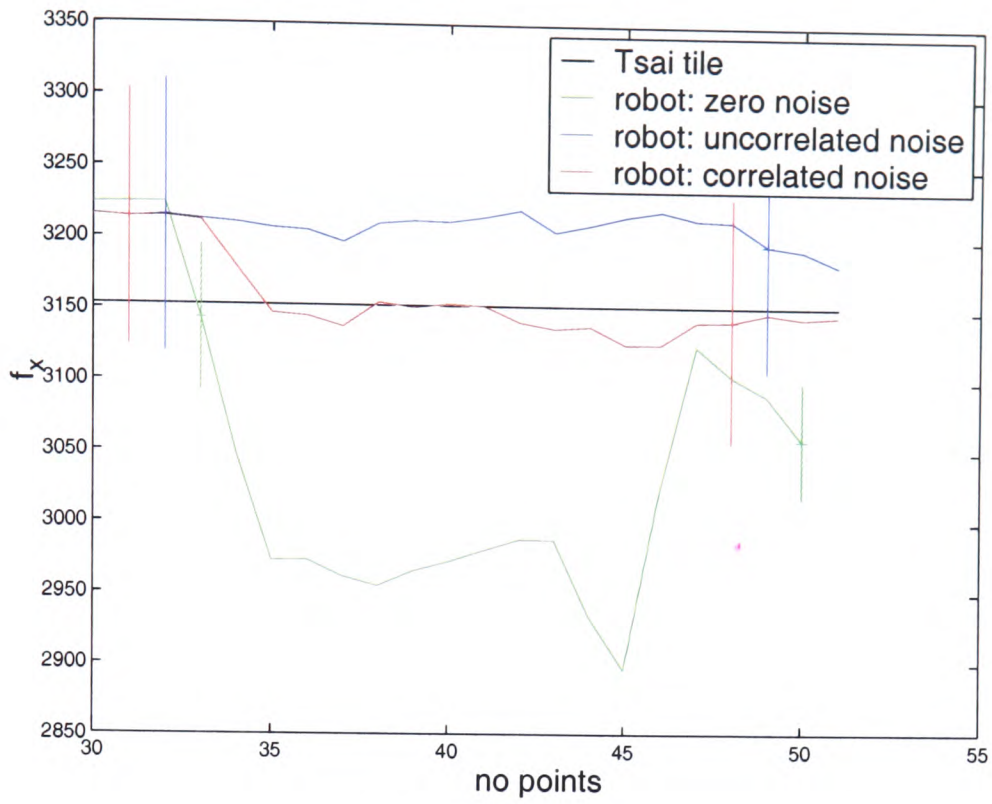


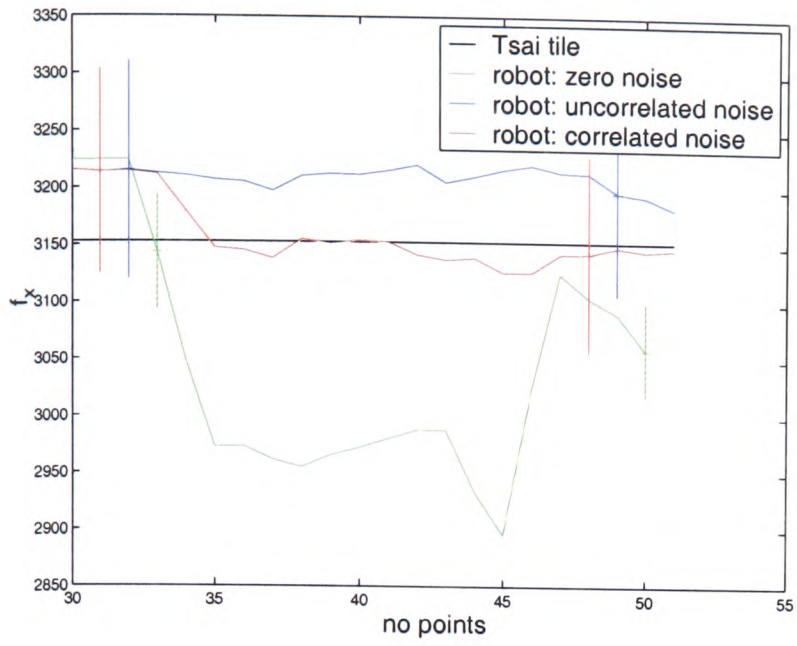
Figure 4.11: The recovered focal length for varying number of scene points. Of note is i) the variation of focal length for the robot world model with zero noise and ii) the divergence of focal length obtained using the uncorrelated and correlated robot model at around the 35th point. The focal length obtained from the uncorrelated robot model then tends towards the result obtained from the Tsai tile calibration.

4.6.3 Recovered manipulator points

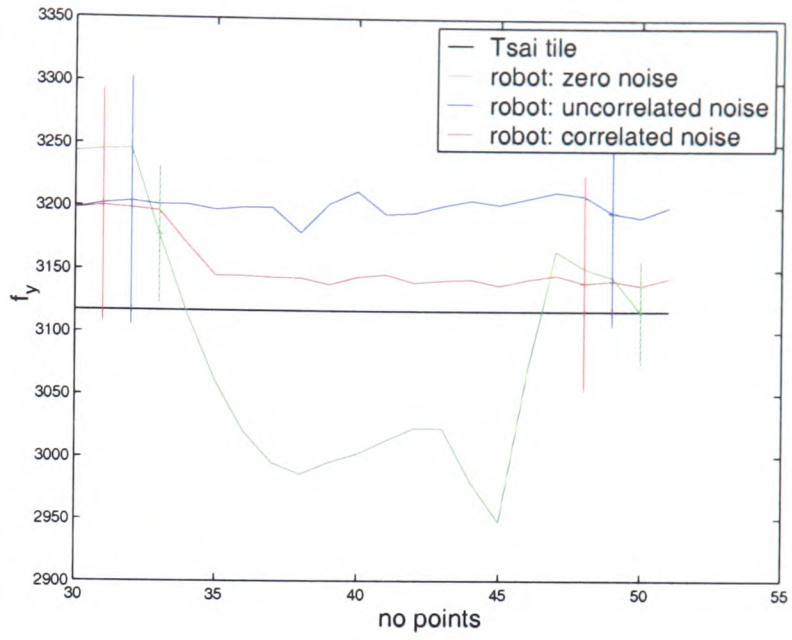
The movement vectors of the recovered world points from their corresponding measured points are shown in Figure 4.14(a); for display purposes the vectors have been magnified by a factor of ten for clarity. If two points are close spatially their movement are expected to be very similar, and, if far apart, to be un-correlated. This correlation is shown in Figure 4.14(b) which is a zoomed in portion of the workcell containing a closely spaced cube, the resulting movement vectors point in roughly the same direction. The length of the largest movement vector is 6.3mm with an average movement of 2.2mm for the eighty scene points.

4.7 Future Work

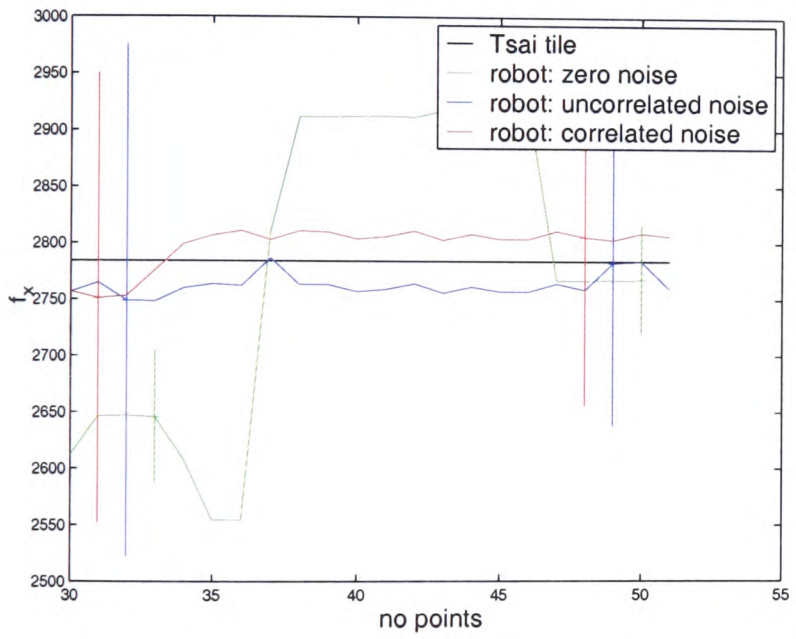
Of the advantages of using the robot to calibrate alluded to in the introduction, we have not investigated that of background and autonomous calibration. There are perhaps two valuable avenues of exploration. The first is to calibrate continuously during use by the



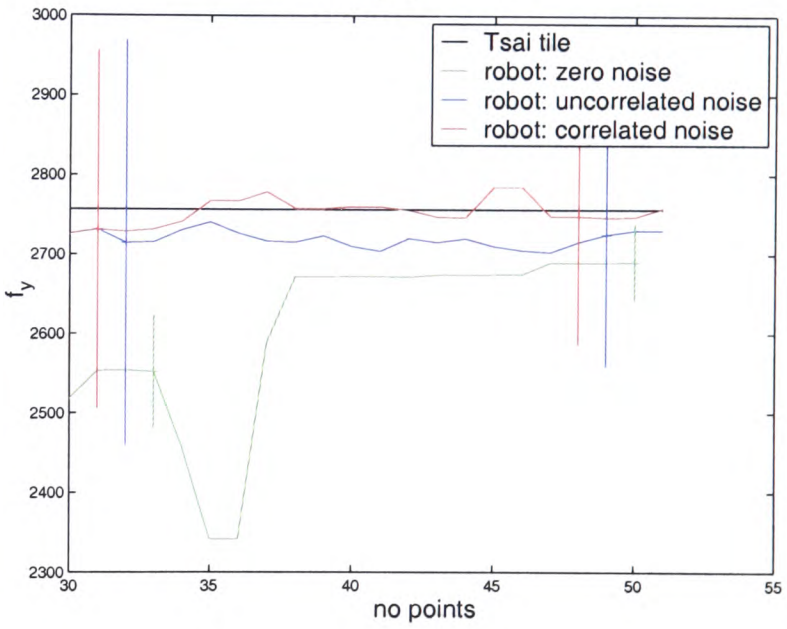
(a)



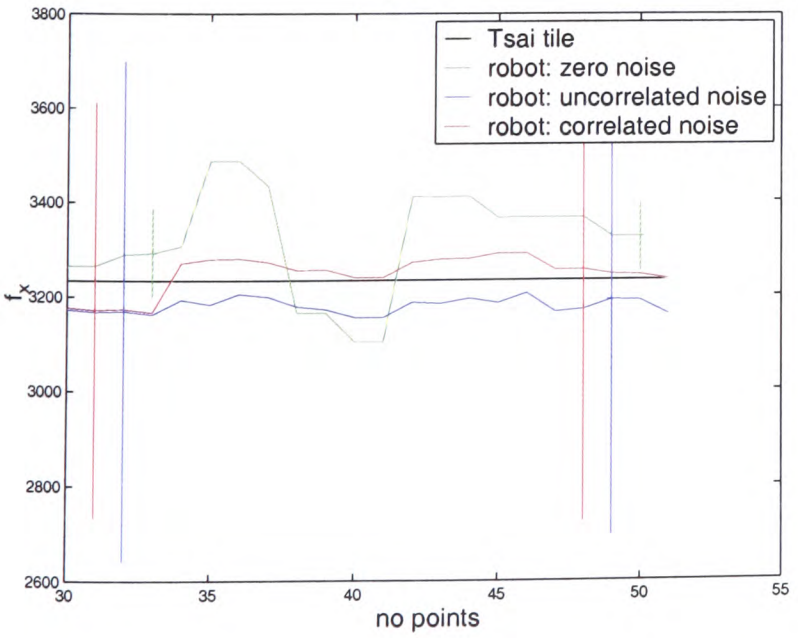
(b)



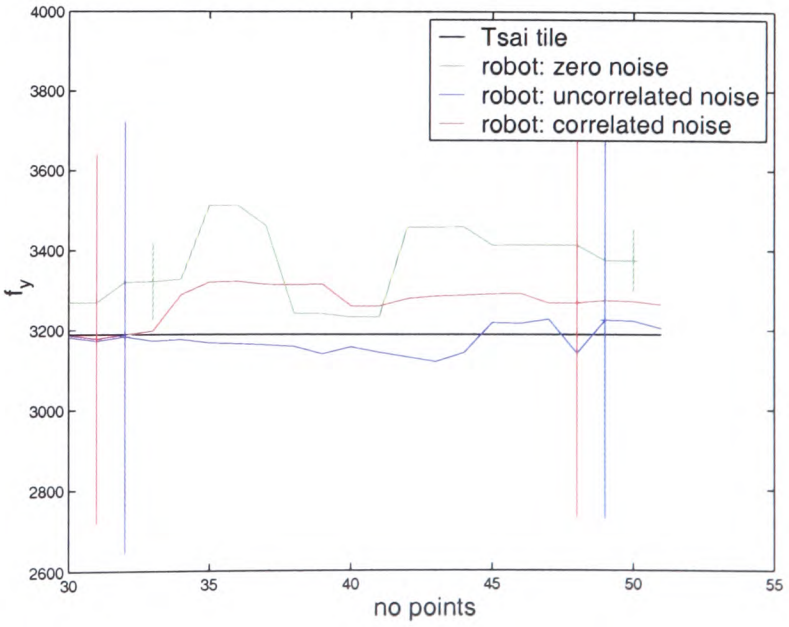
(c)



(d)

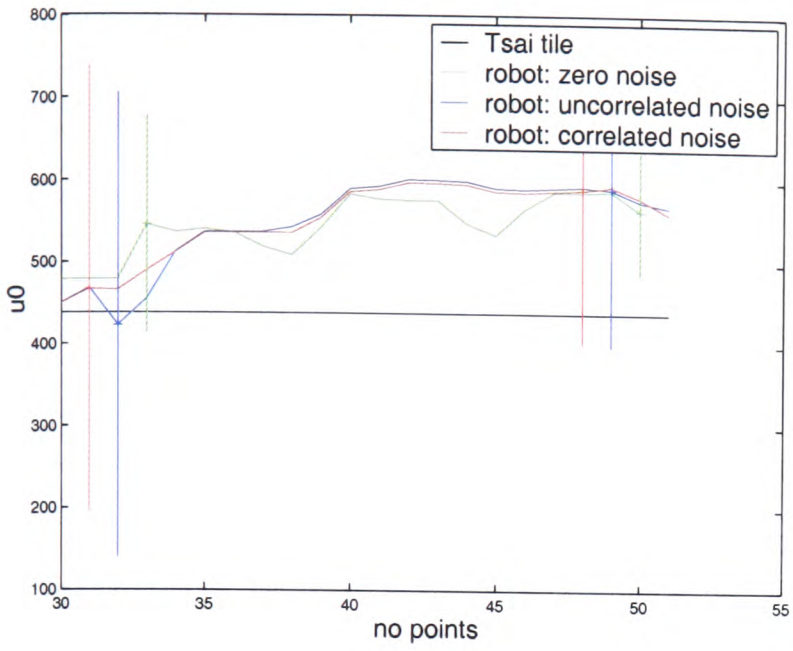


(e)

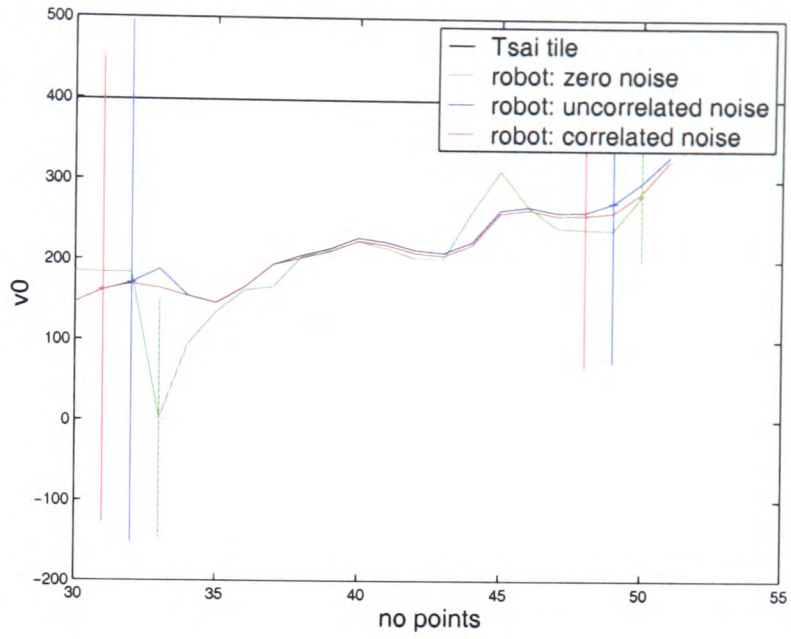


(f)

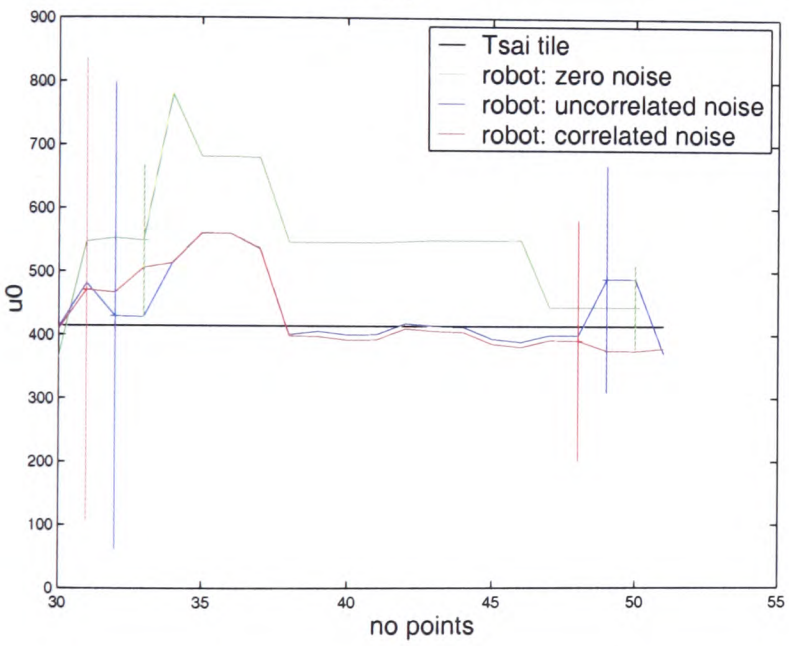
Figure 4.12: The recovered focal lengths for varying number of scene points from (a-b) Camera 1, (c-d) Camera 2, (e-f) Camera 3.



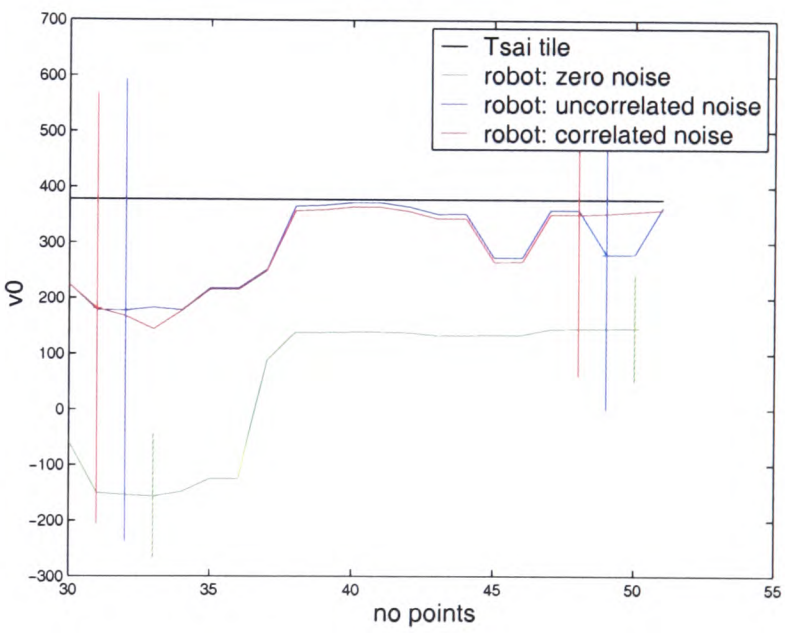
(a)



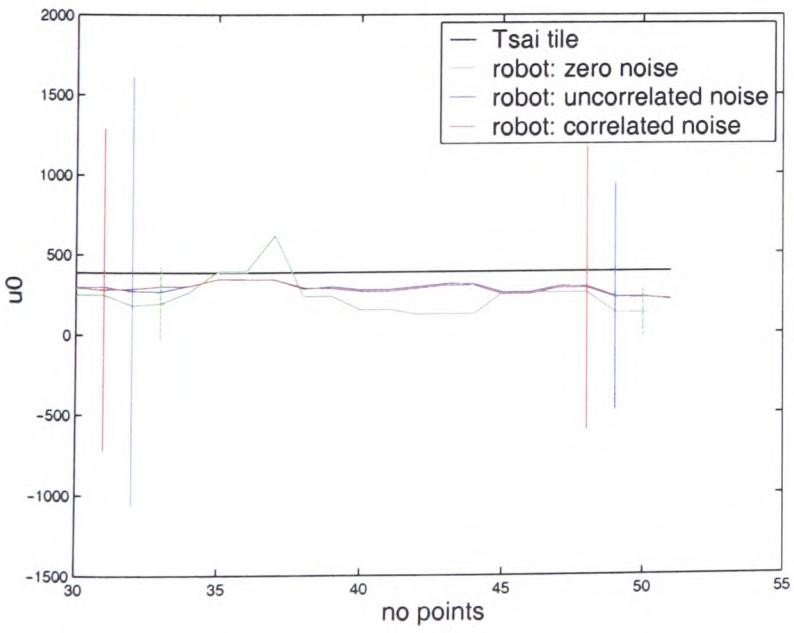
(b)



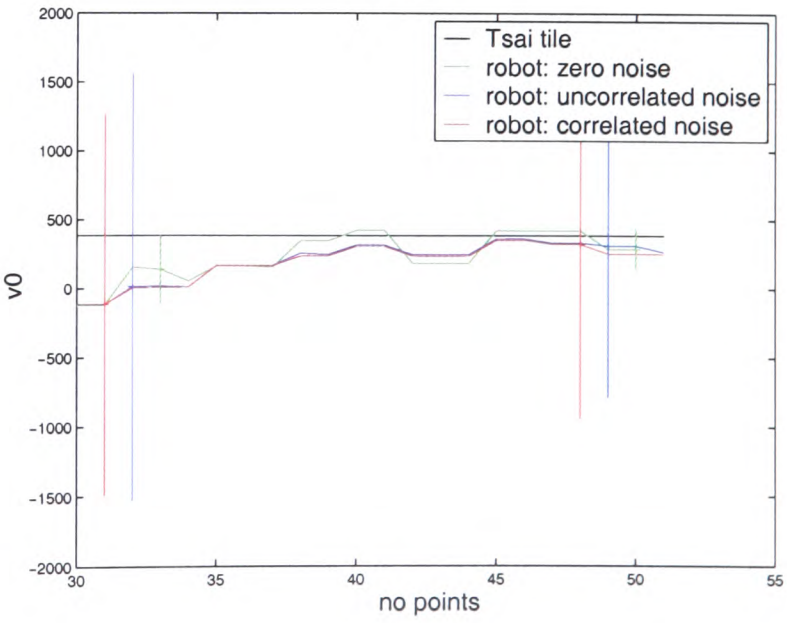
(c)



(d)



(e)



(f)

Figure 4.13: The recovered principal point for varying number of scene points from (a-b) Camera 1, (c-d) Camera 2, (e-f) Camera 3.

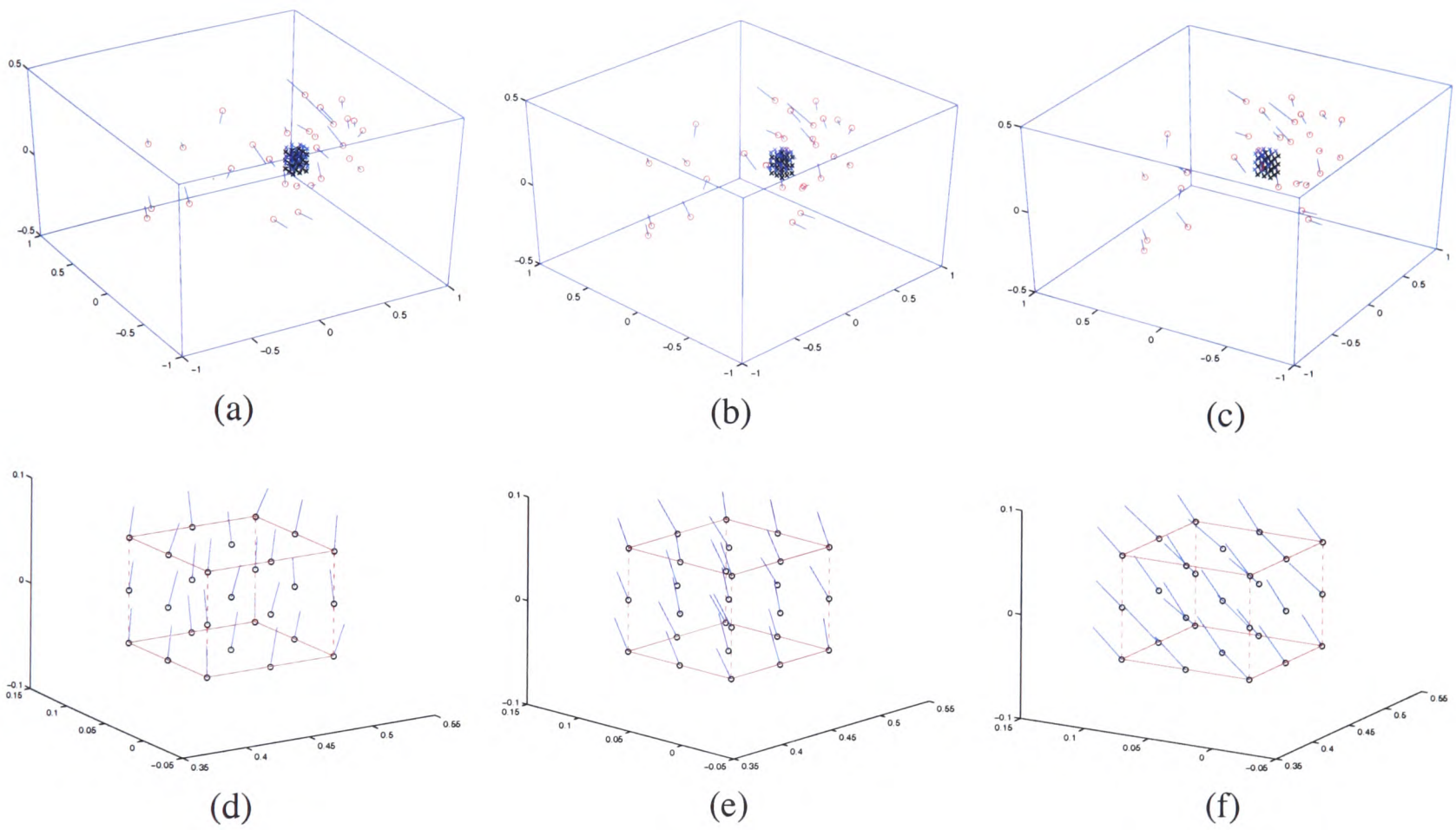


Figure 4.14: (a-c) Each measured world points dotted around the robot workcell are indicated by a circle. The movement of recovered world point is indicated by the attached blue line which has been magnified by a factor of 10 for clarity. The tightly packed cube can be seen in the middle of the workcell. (d-f) The tightly packed cube with coordinated movement vectors.

operator. This has the attractions of placing most calibration points in the areas of the workspace most used and also of not requiring any autonomous movements of the arm, movements which may be undesirable in cluttered or sensitive work environments. The disadvantage is that the calibration may become too specialized to the heavily used area.

The second avenue is to consider autonomous robot arm movements that at each moment best improve the calibration. Given that the information about the calibration is held in the camera covariance matrix, one way might be to generate a local scalar field based on the Fisher information metric (the inverse covariance) or the related Shannon information measure (the log of the Fisher information matrix). However, using such multidimensional measures one runs up against the problem of how to compare improvements in, say, the extrinsic rotation versus improvements in focal length. An alternative is to consider the predicted reduction in reprojection error of a standard set of virtual scene points spread uniformly throughout the workspace. As discussed earlier, the reprojection error ellipse in the image is

$$\Sigma_{\mathbf{x}} = \nabla \mathbf{K_c} \Sigma_{\mathbf{c}} \nabla \mathbf{K_c}^{\top} ,$$

and this might be represented as a scalar say by the product of it eigenvalues or by its larger eigenvalue.

4.8 Conclusions

This chapter has detailed a method of camera calibration using a robot manipulator arm to supply the world points. It was noted that these points cannot be assumed to be fiducial and a model of the robot tip position which respected the physical attributes of the robot arm was presented. This measurement model was used to obtain a MLE of the camera parameters for multiple cameras which was implemented in a Levenberg-Marquardt numerical minimisation algorithm. Experiments were conducted which supported the usefulness of this extra positional information to both obtain accurate camera parameters and to recover accurate world point positions.

The next chapter details an implementation of a calibrated visual pose tracker for telemanipulated objects. The camera calibration is obtained using the method in this chapter

utilising the proposed locally correlated, globally uncorrelated world point measurement model.

5

Visual Pose Tracking

This chapter details the development of the visual model-based tracker used in the Oxford teleoperation system. The approach adopted is an extension to the method first introduced by Harris in his RAPiD tracker [20]. The improvements over Harris' approach include the extension to multiple cameras and performance improvements by the use of robust methods.

5.1 Introduction

Two important choices in the design of the vision system for the work conducted in this thesis were, first, whether to use data-driven structural recovery methods for polyhedra (eg. [106, 107]) or model-based methods [17–24] and, secondly, whether to use calibrated vision to recover Euclidean structure or un- or partially-calibrated vision to recover structure modulo an unknown transformation. For the latter choice, whilst there are certainly tasks in hand-eye coordination for which uncalibrated vision is quite adequate [25, 26], the need to function in a Euclidean robot work space and the a priori undefined nature of the operator's tasks suggest that the calibrated route might be more embracing, though more inflexible. For the former choice, the needs for robustness and real-time performance on

modest hardware still point to a purely model-based approach.

Of the model-based pose tracking methods, the one we pursue here is that of Harris in his RAPiD tracker [20, 108]. Three-dimensional objects are modelled by a set of control points which lie on edges, which may be either surface creases or surface albedo markings, allowing the corresponding lines to be detected in the image. The method assumes any pose change required between the current estimated pose and the actual pose is sufficiently small (i) to allow a linearization of the solution, and (ii) to alleviate the problem of finding correspondences from frame to frame within an image stream. The correspondences sought are between predicted control points and measured image lines, allowing image search in 1D rather than 2D. Its very sparing use of image data is a key advantage of Harris' method, with usually only a few hundred pixels per image being addressed.

However three aspects of Harris' original monocular tracking method appeared unsatisfactory. First it was found to be easily broken by occlusions and changing lighting. Methods of mitigating this problem using robust line detection have been investigated in our laboratory both by Heuring [35, 109] using RANSAC [28] and by Armstrong [110] using case deletion diagnostics. Here in addition we add a robust layer to the pose recovery minimization. Although removing outliers has a marked effect on tracking performance, the second problem found was that the accuracy of the pose recovered in a single camera was poor, with evident correlation between depth and rotation about axes parallel to the image plane. Maitland and Harris [111] had already noted as much when recovering the pose of a pointing device destined for neurosurgical application [112]. They reported improved accuracy using two closely spaced cameras; but the object was stationary, had an elaborate pattern drawn on it and was visible at all times during measurement to both cameras. Gennery too noted that his tracking system often failed using a single camera, rarely failed with two cameras, and that his then current system used three cameras [19]. The third difficulty, or rather uncertainty, was that the convergence properties and dynamic performances of the monocular and multicamera methods were largely unreported.

This chapter provides an experimental appraisal of these issues, and is ordered as follows. The next section defines the coordinate systems used and reviews Harris' method for computing change of pose, but within our notational framework for multiple objects and cameras. Section 5.3 describes the experimental evaluation of

- static accuracies in single and three camera cases, using crease edges on a polyhedral object;
- the effects of incorporating two robust methods on pose convergence; and
- the dynamic performance of the multicamera version.

Section 5.4 describes additional details of the real-time implementation and Section 5.5 illustrates the tracker's performance using two teleoperative experiments, an insertion task and an impact control task.

5.2 Updating the pose of objects

5.2.1 Coordinate frames

Each object model is described in its own frame B by the coordinates $\mathbf{X}$ of a set of control points. The points may be special features (eg points or 'lights'), but more usually are specified to lie on object edges which are geometrically fixed as either crease edges between surfaces or albedo markings on surfaces, as sketched in Figure 5.1. The underlying object need not be polyhedral.

The object's pose is one or other representation of the transformation $\{\mathrm{R}_{\mathrm{W}\to\mathrm{B}}, \mathbf{t}_{\mathrm{W}\to\mathrm{B}}\}$ that takes points in frame B into points in a fixed world frame W. For example, using a non-homogeneous rotation matrix and translation representation $\mathbf{X}^{\mathrm{W}} = \mathrm{R}_{\mathrm{W}\to\mathrm{B}}\mathbf{X}^{\mathrm{B}} + \mathbf{t}_{\mathrm{W}\to\mathrm{B}}$ where $\mathbf{t}_{\mathrm{W}\to\mathrm{B}}$ denotes the origin of B in W. The pose of a camera C relative to the world coordinate system is defined in the inverse manner as the rotation and translation $\{\mathrm{R}_{\mathrm{C}\to\mathrm{W}}, \mathbf{t}_{\mathrm{C}\to\mathrm{W}}\}$ where $\mathbf{X}^{\mathrm{C}} = \mathrm{R}_{\mathrm{C}\to\mathrm{W}}\mathbf{X}^{\mathrm{W}} + \mathbf{t}_{\mathrm{C}\to\mathrm{W}}$. It is a matter of convenience below to introduce a frame A that is aligned with W but has its origin coincident with the object frame B: $\mathbf{X}^{\mathrm{A}} = \mathrm{R}_{\mathrm{W}\to\mathrm{B}}\mathbf{X}^{\mathrm{B}}$.

The method of establishing and calibrating the coordinate frames is deferred to Section 5.4.

5.2.2 Harris' method of recovering change of pose

Here we review Harris' method within our notational framework. In brief, this section calculates the image Jacobian with respect to object pose represented in a 6-vector translation and angle-axis vector for each control point subject to the constraint that only the control

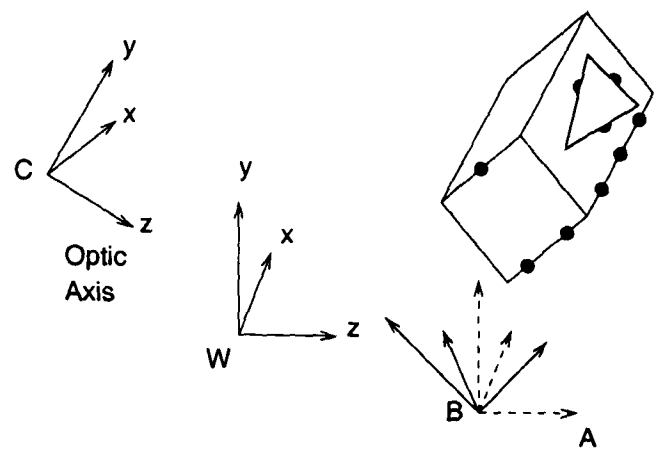


Figure 5.1: The object is modelled within its own coordinate frame B as a set of control points lying on edges, which may be crease or albedo edges. The world and camera coordinate frames are W and C. The aligned frame A, introduced as a notational convenience, has the same orientation as W but has the same origin as B.

point positional information which is meaningful lies perpendicular to the control points corresponding edge. This image Jacobian is then inverted using a least-square method and multiplied by the control point innovation vector to obtain the object pose innovation. A numerical procedure could have been used to calculate the image Jacobian however for completeness we derive it analytically just as Harris did.

The image operations are of the sort routinely used in visual contour tracking . As each new image is captured, the current estimate of pose (or the predicted estimate if the rate of change of pose is modelled) is used to project the visible control points and their lines onto the image, and a search is initiated from each control point in a direction perpendicular to the projected line in order to find any point on the actual imaged line in the new image. The new pose is estimated by minimizing the control point to image line displacements.

Consider an object whose position at some instant is described in the world frame by

$$\mathbf{X}^{\mathrm{W}} \;=\; \mathrm{R}_{\mathrm{W}\to\mathrm{B}}\mathbf{X}^{\mathrm{B}} + \mathbf{t}_{\mathrm{W}\to\mathrm{B}} \;=\; \mathbf{X}^{\mathrm{A}} + \mathbf{t}_{\mathrm{W}\to\mathrm{B}}\ .$$

The object's angular and rectilinear velocities are Ω and $\mathbf{v}$ respectively, so that in a small time $\delta\tau$ the object will move to a new position

$$\mathbf{X}^{\mathrm{W}'} = \mathbf{X}^{\mathrm{A}} + (\Omega\delta\tau) \times \mathbf{X}^{\mathrm{A}} + \mathbf{t}_{\mathrm{W}\to\mathrm{B}} + \mathbf{v}\delta\tau\ .$$

By writing the product of the time interval and velocity screw as

$$\delta\mathbf{s} = \delta\tau[v_x, v_y, v_z, \Omega_x, \Omega_y, \Omega_z]^\top$$

and composing the matrix

$$\mathtt{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & Z^{\mathrm{A}} & -Y^{\mathrm{A}} \\ 0 & 1 & 0 & -Z^{\mathrm{A}} & 0 & X^{\mathrm{A}} \\ 0 & 0 & 1 & Y^{\mathrm{A}} & -X^{\mathrm{A}} & 0 \end{bmatrix}$$

the new position can be written as

$$\mathbf{X}^{\mathrm{W}\prime} = \mathbf{X}^{\mathrm{W}} + \mathtt{G}\delta\mathbf{s} \ .$$

To obtain the measurement equation, we need to transform the new positions into the camera frame, and thence project into the image. The camera is an idealized device with unity aspect ratio and unity focal length. In the camera frame, the modified coordinates are related to the originals by

$$\mathbf{X}^{\mathrm{C}\prime} = \mathbf{X}^{\mathrm{C}} + \mathtt{R}_{\mathrm{C}\to\mathrm{W}}\mathtt{G}\delta\mathbf{s} \ .$$

The projection into the idealized image is $\mathbf{x} = -\mathbf{X}^{\mathrm{C}}/Z^{\mathrm{C}}$ (where we will keep $\mathbf{x}$ as a 3-vector $(x, y, -1)^{\top}$ for the moment). The change in image coordinate as the object is moved is thus

$$(\mathbf{x}' - \mathbf{x}) = -\left[\mathbf{X}^{\mathrm{C}\prime}/Z^{\mathrm{C}\prime} - \mathbf{X}^{\mathrm{C}}/Z^{\mathrm{C}}\right] \ .$$

If the change in depth is small, so that $Z^{\mathrm{C}\prime} = (1+\alpha)Z^{\mathrm{C}}$, with $|\alpha| \ll 1$, then

$$\begin{aligned} (\mathbf{x}' - \mathbf{x}) &\approx -\frac{1}{Z^{\mathrm{C}}}\left[\mathbf{X}^{\mathrm{C}\prime} - \mathbf{X}^{\mathrm{C}} - \alpha\mathbf{X}^{\mathrm{C}\prime}\right] \\ &= -\frac{1}{Z^{\mathrm{C}}}\left[\mathtt{R}_{\mathrm{C}\to\mathrm{W}}\mathtt{G}\delta\mathbf{s} - \alpha(\mathbf{X}^{\mathrm{C}} + \mathtt{R}_{\mathrm{C}\to\mathrm{W}}\mathtt{G}\delta\mathbf{s})\right] \ . \end{aligned} \tag{5.1}$$

But α may be expressed as

$$\alpha = (Z^{\mathrm{C}\prime} - Z^{\mathrm{C}})/Z^{\mathrm{C}} = \frac{1}{Z^{\mathrm{C}}}[001]\mathtt{R}_{\mathrm{C}\to\mathrm{W}}\mathtt{G}\delta\mathbf{s}$$

and so ignoring $\delta\mathbf{s}^2$ terms

$$\begin{aligned} (\mathbf{x}' - \mathbf{x}) &\approx -\frac{1}{Z^{\mathrm{C}}}\left[\mathtt{R}_{\mathrm{C}\to\mathrm{W}}\mathtt{G}\delta\mathbf{s} - \frac{1}{Z_C}[001]\mathtt{R}_{\mathrm{C}\to\mathrm{W}}\mathtt{G}\delta\mathbf{s}\mathbf{X}^{\mathrm{C}}\right] \\ &= -\frac{1}{Z^{\mathrm{C}}}\left[\mathtt{R}_{\mathrm{C}\to\mathrm{W}}\mathtt{G}\delta\mathbf{s} + ([001]\mathtt{R}_{\mathrm{C}\to\mathrm{W}}\mathtt{G}\delta\mathbf{s})\mathbf{x}\right] \ . \end{aligned}$$

The above could be used to recover the change of pose if point to point matches could be established between several points $\mathbf{x}$ and their correspondences $\mathbf{x}'$, for example when

the control points are corners or lights. However in this work the control points typically lie on lines or curves, and we thus only obtain information on point-to-line or point-to-curve matches because of the aperture problem — we know that $\mathbf{x}'$ lies on a particular line or curve, but not exactly where.

All the information is preserved if the inner product is formed between $(\mathbf{x}' - \mathbf{x})$ and the unit normal $\hat{\mathbf{n}}$ to the curve or line at the point $\mathbf{x}$. The measurement equation becomes

$$\hat{\mathbf{n}}^\top(\mathbf{x}' - \mathbf{x}) = -\frac{1}{Z^{\mathrm{C}}}\hat{\mathbf{n}}^\top[\mathrm{I}_3 + \mathbf{x}[001]]\mathrm{R}_{\mathrm{C}\to\mathrm{W}}\mathrm{G}\delta\mathbf{s}$$

or, using 2-vectors for image quantities,

$$\hat{\mathbf{n}}^\top(\mathbf{x}' - \mathbf{x}) = -\frac{1}{Z^{\mathrm{C}}}\begin{bmatrix} n_x & n_y \end{bmatrix}\begin{bmatrix} 1 & 0 & x \\ 0 & 1 & y \end{bmatrix}\mathrm{R}_{\mathrm{C}\to\mathrm{W}}\mathrm{G}\delta\mathbf{s}\,.$$

As Figure 5.2 illustrates, $\hat{\mathbf{n}}^\top(\mathbf{x}' - \mathbf{x})$ is the perpendicular distance between the curves, which we will assume have a radius of curvature much greater than this distance. Note that the choice of the positive direction of $\hat{\mathbf{n}}$ is arbitrary.

The system

$$\begin{bmatrix} m_i \end{bmatrix} = \begin{bmatrix} \mathbf{a}_i \end{bmatrix}\delta\mathbf{s} \tag{5.2}$$

where $m_i = \hat{\mathbf{n}}_i^\top(\mathbf{x}_i' - \mathbf{x}_i)$, and so on, is built up for several points i, and solved for greater than 6 using a least-squares method — here singular value decomposition. Note that the method assumes calibrated cameras, for which we use the method of chapter 4.

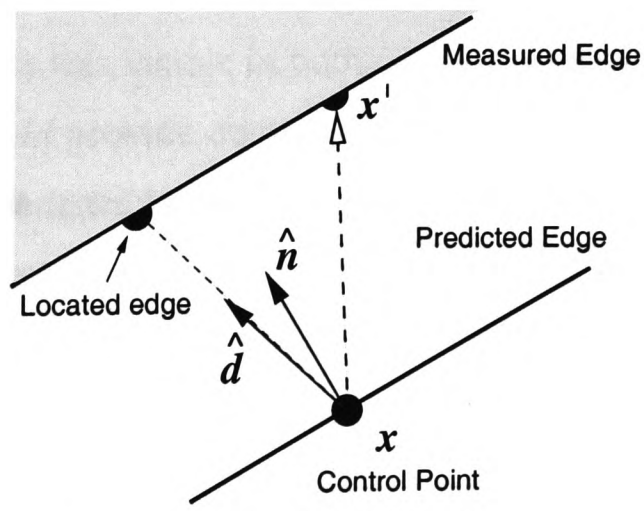


Figure 5.2: Because of the aperture problem, only the perpendicular distance along $\hat{\mathbf{n}}$ is measurable. It is not necessary to search for the edge along $\hat{\mathbf{n}}$: it is quicker to search along one of the eight cardinal directions $\hat{\mathbf{d}}$, here a diagonal.

Harris pointed out that if the pose change is small, the lines or curves are close to parallel and the perpendicular distance

$$\hat{\mathbf{n}}^\top(\mathbf{x}' - \mathbf{x}) \approx d\hat{\mathbf{n}}^\top\hat{\mathbf{d}}\,,$$

where $\mathbf{d} = d\hat{\mathbf{d}}$ is *any* vector from $\mathbf{x}$ to the new position of the line. It is considerably faster to search in pixel-based hardware along one of the eight cardinal directions (N, NE, E etc), and so $\hat{\mathbf{d}}$ is chosen to be the cardinal direction closest to $\hat{\mathbf{n}}$.

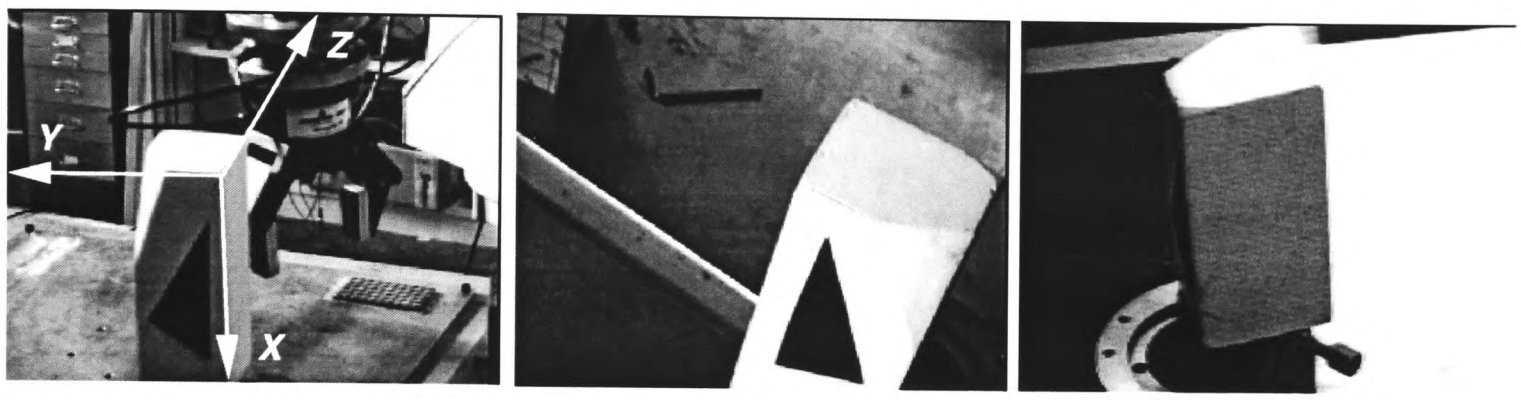


Figure 5.3: View of the object used for static accuracy tests from the three cameras.

5.2.3 Using multiple cameras

Although the monocular method is a successful tracker, its ability to recover pose monocularly is less satisfactory. Maitland and Harris [111] improved the accuracy by extending the method to multiple cameras (as we independently did).

Maitland and Harris used a stereo rig with quite closely spaced cameras, and the tracked planar surface of the object was visible in both cameras at all times. Perhaps because this viewing configuration *could* provide correspondence, they did not stress that there is no need to establish correspondence between control points, and each camera can be treated independently in terms of measurement. The independence of the cameras has no particular impact on image search, as it is already 1-dimensional (and because the image search is between point and a line there is no possibility of reducing the search to a predictable neighbourhood around a point). The considerable benefit of independence between views is that it retains the monocular system's robustness to obscuration and loss of control points, the more important for the widely spaced cameras used in this work. Gennery [19] made similar remarks about the advantages of camera independence, and makes reference to the

work of earlier authors where independence was implicit but not exploited because only a single camera was used.

The extension to multiple cameras is straightforward: with several points i and cameras c, one forms the system

$$\left[\ \mathbf{a}_{ic}\ \right]\delta\mathbf{s} = \left[\ m_{ic}\ \right]\ ,$$

where $m_{ic} = \hat{\mathbf{n}}_{ic}^{\top}(\mathbf{x}'_{ic} - \mathbf{x}_{ic})$ and the image Jacobian $\mathbf{a}_{ic}$ is the row vector

$$\mathbf{a}_{ic} = -\frac{1}{Z^c{}_i}\hat{\mathbf{n}}_{ic}^{\top}[\mathtt{I}_3 + \mathbf{x}_{ic}[001]]\mathtt{R}_{c\to w}\mathtt{G}_i\ ,$$

and is solved as before.

5.3 Experimental Evaluation

5.3.1 Static accuracy

Maitland and Harris [111] were principally concerned with the static positional accuracy at the tip of a planar pointer — the tip was to mark the position for a surgical procedure [112]. With 64 control points viewed by both cameras, a camera separation of 0.76m and a range to the target of $\sim$ 1m, they found a standard deviation in the position of $\sim$ 0.4mm. Angular errors were not given. In our first experiments here we use three cameras and use only the crease edges in the polyhedral object (Figure 5.3). The range to the object from each camera was $\sim$ 1.5m, and the cameras, spaced by some 2.2m, had near orthogonal views.

The object's x-axis was first aligned (approximately) with the world frame's x-axis. The object was then rotated about this axis in steps of 10° and the rotational and translational components of pose recovered. After each movement the pose was allowed to settle so that dynamic effects were eliminated.

The comparisons of recovered rotation vs. set rotation for one camera and three cameras are given in Figures 5.4(a1) and (a3) respectively. To obtain a measure of error, each point and error bar in the graph represent the average and standard deviation of four measurements. For three cameras, the slope of the plot of recovered rotation angle vs. set rotation angle was 1.005 ± 0.009, and the mean change in rotation between steps was found to be $(10.1 \pm 2.1)°$. For the single camera, pose recovery fails at about 140° because of lack of

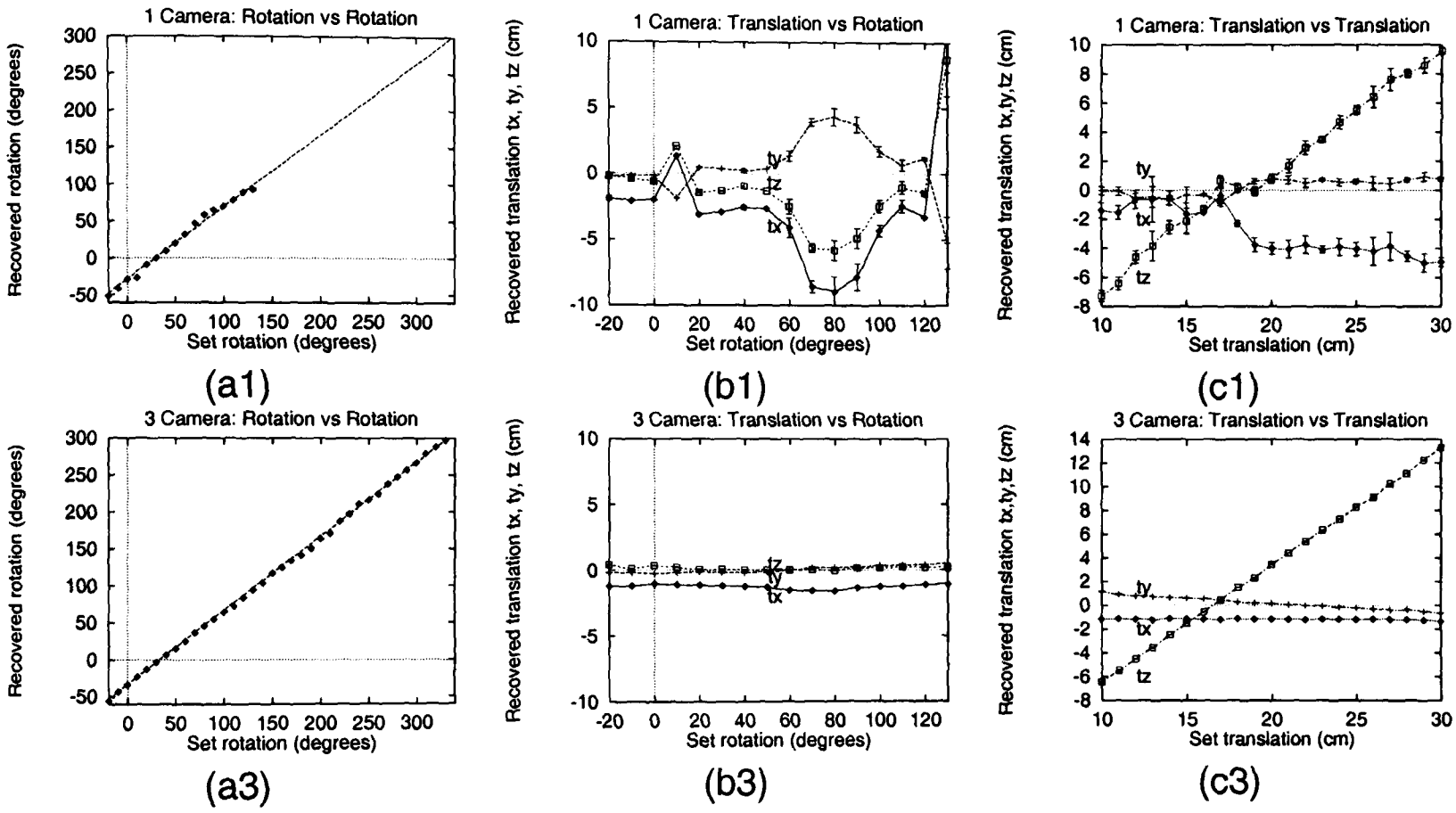


Figure 5.4: (a) Recovered rotation versus set rotation for (a1) 1 camera compared with that from (a3) 3 cameras. (The zero offset of 33° arises trivially because no attempt was made to align the y- and z-axes.) (b) Recovered translation components versus set rotation for (b1) 1 camera and (b3) 3 cameras. (Again the offset arises trivially because no attempt was made to align the origins of the $x-$axes.) (c) Recovered translation components versus a set translation for (c1) 1 camera and (c3) 3 cameras.

data — but this is particular to this example. Of more general interest is that there was increased error in the recovered rotation, but not dramatically so: the mean change in angle was now $(9.6 \pm 4.0)°$.

More interesting is the comparison between recovered translational parameters, shown in Figure 5.4(b1, b3). For three cameras, the t_y and t_z are, as expected, close to constant and zero throughout, and t_x is close to a constant offset. However, the components recovered from the single camera alone are evidently erroneous and highly correlated.

Taking the values recovered in the three-camera experiment as veridical, the error in the translation $\delta\mathbf{t}_j$ was found over the rotation range $-20°$ to $+130°$. The eigenvalues of the scatter matrix

$$\mathtt{M} = \sum_j [\delta\mathbf{t}_j - \bar{\delta\mathbf{t}}][\delta\mathbf{t}_j - \bar{\delta\mathbf{t}}]^\top$$

were found as $\lambda_1 = 1.2 \times 10^{-1}$, $\lambda_2 = 2.6 \times 10^{-1}$, and $\lambda_3 = 1.9 \times 10^{+2}$, indicating that the translation was determined with similar accuracy along the directions of eigenvectors $\mathbf{e}_1$ and $\mathbf{e}_2$, but with much poorer accuracy along the direction of the third eigenvector in the W frame, $\mathbf{e}_3 = (+0.71, -0.43, +0.56)$. Multiplying this by the rotation matrix $\mathtt{R}_{\mathtt{C}\to\mathtt{W}}$ for the single camera (camera 1) gives the direction in the camera's frame as $\mathbf{e}_3^{\mathtt{C}} = (-0.20, -0.42, +0.98)$ which, as expected, is close to the optic axis of the single camera.

A second comparative run was made using a set translational movement rather than rotational movement. The object was moved some 200 mm in steps of (10 ± 0.5) mm along a straight line close to the world frame's z-direction and certainly in the world's y, z-plane. Figures 5.4(c1) and (c3) compare the recovered translation components for one camera and three cameras respectively. Again, each point gives the mean and standard deviation of several measurements. For three cameras, the step size recovered was (9.8 ± 0.9) mm, whereas that for one camera was (9.2 ± 4.2) mm. Assuming the values obtained using three cameras to be veridical, errors were computed for the one camera case and, using the same eigen-analysis, the errors were again found to lie predominantly along the optic axis of the single camera.

5.3.2 Convergence and robustness

Robust pose

In this work, the linear system $\mathtt{A}\delta\mathbf{s} = \mathbf{m}$ is solved by first applying a robust estimator to eliminate outliers, and then by using singular value decomposition on the remaining inliers. Because we do not know the standard deviation of the measurements a priori we use least median of squares [113]. In repeated trials, the minimal groups of 6 matches are randomly selected to determine a value for $\delta\mathbf{s}$, and this value used to determine the median of the squared magnitudes of the deviations $e_i^2 = (m_i - m_i^{\text{fitted}})^2$. The solution with the smallest median is used to estimate the standard deviation σ of the data, exploiting the fact that $\text{med}|e_i|/\Phi^{-1}(0.75)$ is an asymptotically consistent estimator of σ when the e_i are distributed like $N(0, \sigma^2)$. Here Φ is the cumulative distribution function for the Gaussian pdf, and Φ^{-1} its inverse. The base estimate of σ is [114]

$$\sigma = \frac{1}{\Phi^{-1}(0.75)}\left(1 + \frac{5}{n-p}\right)\sqrt{\text{med}\, e_i^2}$$

or $\sigma = 1.48(1 + 5/9)\sqrt{\text{med}\, e_i^2}$ for $n = 15$ trials and $p = 6$ parameters (see below). Measurements were split between inliers and outliers at the 95% confidence limit using

$$i \in \begin{cases} \text{inliers} & \text{if } |e_i| \leq 1.96\sigma \\ \text{outliers} & \text{otherwise} \end{cases} .$$

In experiment, the standard dev. σ derived was typically of order 4×10^{-4} in the ideal image with focal length unity. Our cameras have focal length of around 3000 pixels. Now $|e_i|$ is a measure of the distance from the predicted edge to the actual edge. Using Rousseeuw's [114] expression in reverse we find $|e_i| \sim 0.5$ pixel in the physical image. This is commensurate with the edge search mechanism, which operates only to pixel accuracy.

The number of n of random subsamples required depends as $n \log[1 - (1 - \epsilon)^6] = \log[1 - \Psi]$ on the level of contamination ϵ, the number of features in the sample (here 6), and the desired probability Ψ that one of the subsamples contains no outliers. In a series of trials contamination levels were never greater than 20%, and so just 15 trials would meet a 99% confidence criterion. Typically, with the robust line fitter running, the number of outliers at this stage was much lower — around 3% — and case deletion diagnostics

(cf [115]) would provide an alternative and cheap method of locating outliers. The outliers are excluded to the final least squares fit for the change in pose δs, which is made using singular value decomposition.

Robust collinearity

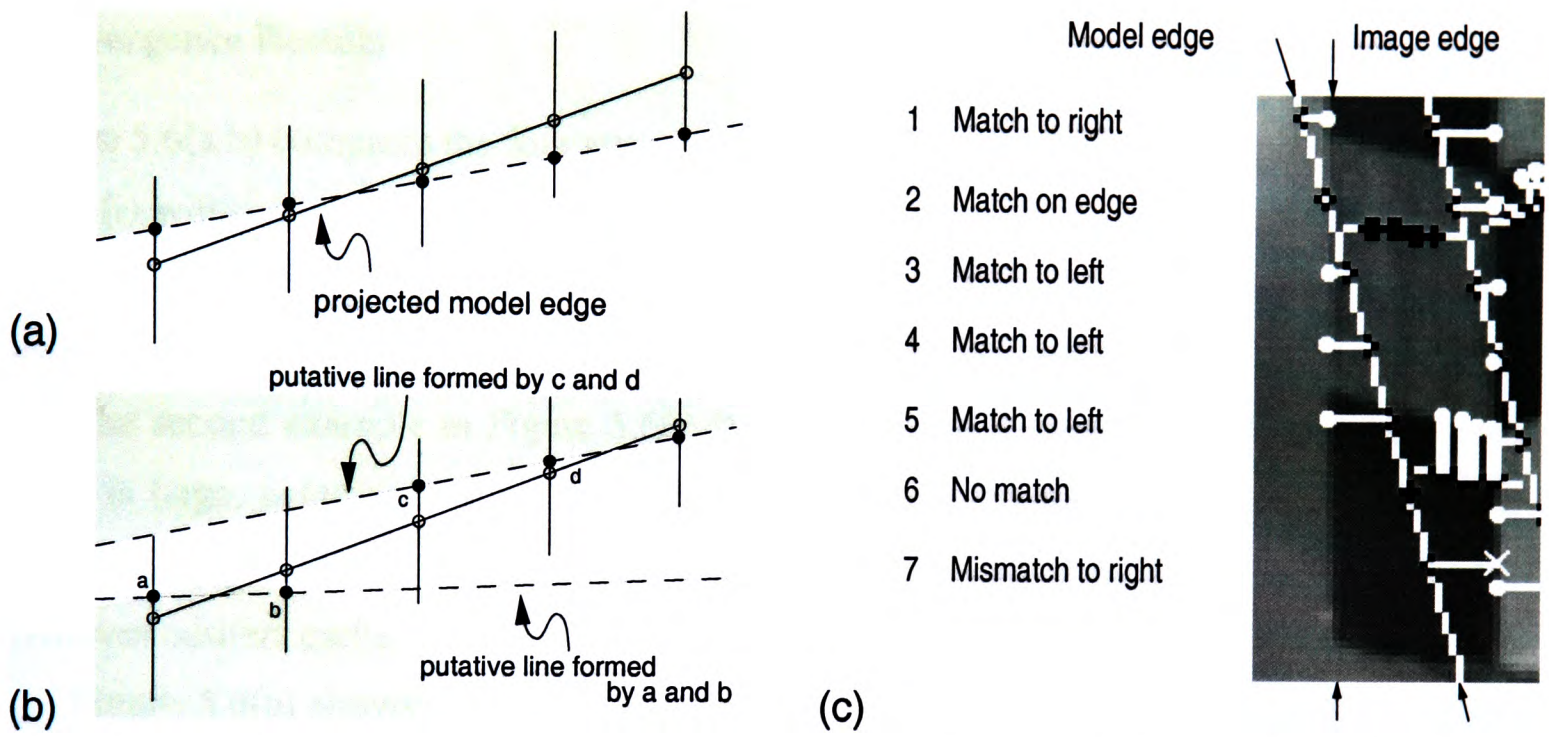


Figure 5.5: (a) The incremental search for an edge feature and (b) the rejection of outliers by the collinearity condition. (c) An example during a run. Of the seven control points on this line, five have correct matches, one had no match, and the last had a mismatch which was correctly detected as an outlier.

Although it is possible to run the tracker with just one control point on a line, using more than two immediately introduces the constraint that the measured control points should be collinear. A random sampling method is effective in ensuring this by rejecting outliers. The desired situation and a failure situation are illustrated in Figures 5.5 (a) and (b). Such "siren" edges arise from independent objects, from shadows cast by the object itself, or, if the pose error is large from other edges on the object itself.

Once the edge locations are found along the cardinal directions, pairs of points are picked at random to define a line, and the perpendicular distance from each edge location to the line computed. As earlier, after computing the standard deviation, inliers and outliers are determined. Figure 5.5(c) shows an example from experiment. Control points 1–5 of the seven control points on the model line are correctly matched to the image line, the 6th

is unmatched because the edge lies beyond the search distance (here 20 pixels) from the control point, but the 7th is mismatched to another edge. The mismatch was correctly found as an outlier. It would of course be possible to re-search the image for evidence of an edge in the expected position, though cost-benefit ratio favours spending the time in iterating the pose updating, as mentioned below.

Convergence Results

Figure 5.6(a,b) compares the four rates of convergence from an incorrect pose to the correct pose (i) without robust calculation, (ii) with robust lines only, (iii) with robust pose only, and finally (iv) with both robust lines and pose computation. In a variety of experiments, using *both* robust methods has always provided the fastest convergence.

The second example in Figure 5.6(c,d) is included as a caveat that, if the initial pose error is large, using only one robust method can prove unsatisfactory. In such cases we find that robust collinearity alone is usually of greater benefit than robust pose alone, as it removes outliers earlier.

Figure 5.6(e) shows the model being pulled onto the image in three iterations of the pose update algorithm when both robust methods are used. In [108], Harris and Stennett suggested that one iteration is sufficient. However as we discuss below, and is apparent in Figure 5.7(c), we find that if the object is moving using just one iteration can lead to steady phase lag between image and model. On the basis of the convergence results we use three iterations per image.

The predominance of phase lag deserves further comment. If the image measurements in equation (5.1) are perfect, the linearization results in an *underestimate* of $|\alpha|$ for a receding object ($\alpha > 0$), but an *overestimate* of $|\alpha|$ for a looming object ($\alpha < 0$). We therefore would expect phase lag and phase lead, respectively, for these cases, and indeed this small effect has been observed with synthetic data. However, the search for a matching edge starts at the expected position, and works symmetrically outwards along the search direction for a limited distance, tending to bias $|\mathbf{x}' - \mathbf{x}|$ towards smaller values. The associated lag dominates for all but the smallest motions.

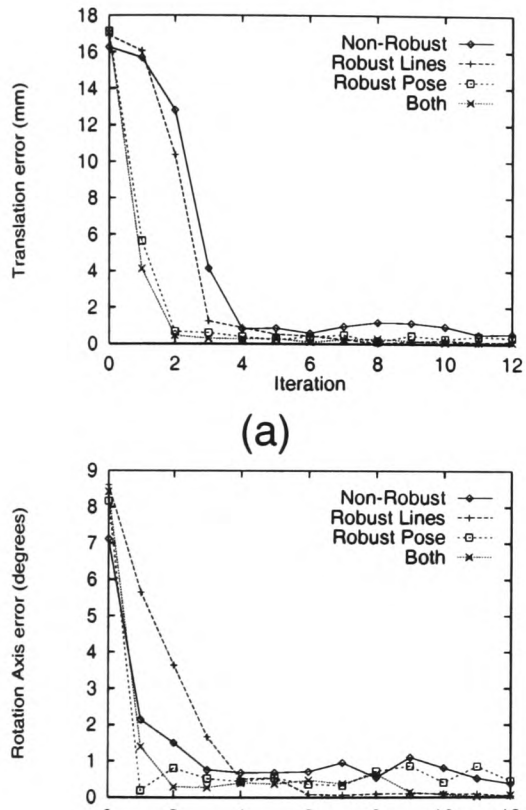


Figure 5.6: Convergence from an initially incorrect pose to the correct pose. In each figure the four lines represent convergence (i) without robust calculation (ii) with robust lines, (iii) with robust pose, and (iv) with both robust lines and pose. Part (a) shows the error in translation (as a distance) , and (b) represents the rotation error as angular error between the recovered and correct axis of rotation axis. Parts (c) and (d) show a second example with a larger initial pose error, providing the caveat that omitting the collinearity check at large pose errors is not advisable. (e) shows the three iterations required using both robust methods.

5.3.3 Dynamic performance

Harris' RAPiD tracker included a constant velocity Kalman filter. Although Evans has given expressions for the covariance matrices [116], these were in terms of a representation other than the angle axis representation used later in [20]. It is rather unclear from the latter publication exactly what covariances are used there. In the filter implemented here we assume that they are of the standard form for discrete-time constant velocity filters, and assuming independent components. Although this is an obvious liberty in the case of the angle axis values, in our work we have found the exact nature of the filter's internal operation to be less important than the way in which filtering is used in the broad.

The *change* in pose is used to recompute *absolute* pose. As explained in [20] translation requires only a simple addition

$$\mathbf{t}(\tau + \delta\tau) = \mathbf{t}(\tau) + \mathbf{v}\delta\tau \ ,$$

but for rotation it is more convenient[1] to transform to a quaternion representation, using the quaternion product

$$\begin{aligned} \delta\mathring{\mathbf{q}} &= (\cos(|\delta\mathbf{a}|/2), \sin(|\delta\mathbf{a}|/2)\hat{\delta\mathbf{a}}) \\ \mathring{\mathbf{q}}(\tau + \delta\tau) &= \delta\mathring{\mathbf{q}}.\mathring{\mathbf{q}}(\tau) \ . \end{aligned}$$

and then to back-transform to obtain $\mathbf{a}(\tau + \delta\tau)$.

In this chapter, each new absolute pose measurement is combined with a running estimate using a constant velocity Kalman filter with twelve components in the state

$$\mathbf{x}_\tau = (t_x, t_y, t_z, a_x, a_y, a_z, \dot{t}_x, \dot{t}_y, \dot{t}_z, \dot{a}_x, \dot{a}_y, \dot{a}_z)^\top$$

with covariance $\mathtt{P}_{\tau|\tau}$, and is assumed to evolve as

$$\mathbf{x}_{\tau+\delta\tau} = \mathtt{F}\mathbf{x}_\tau + \mathbf{v}_x$$

where

$$\mathtt{F} = \begin{bmatrix} \mathtt{I}_6 & \delta\tau\mathtt{I}_6 \\ \mathtt{0}_6 & \mathtt{I}_6 \end{bmatrix}$$

[1] It is *possible* to write down an update rule purely within the angle-axis representation, but much of it would merely replicate the rule for quaternion multiplication.

and $\mathbf{v}_x$ is Gaussian process noise with zero mean and covariance $\mathtt{Q}$. The covariance has the commonly used form for discrete-time constant velocity models. If σ_{t_x} denotes the typical size of the unmodelled second derivative of t_x, and similarly for the other five components, then

$$\mathtt{Q} = \begin{bmatrix} (\delta\tau^4/4)\mathtt{D} & (\delta\tau^3/2)\mathtt{D} \\ (\delta\tau^3/2)\mathtt{D} & (\delta\tau^2)\mathtt{D} \end{bmatrix}$$

where $\mathtt{D}$ is the 6×6 diagonal matrix with diagonal elements $\sigma^2_{\dot{t}_x}$, $\sigma^2_{\dot{t}_y}$, and so on. The filter prediction equations are standard

$$\hat{\mathbf{x}}(k|k-1) = \mathtt{F}\hat{\mathbf{x}}(k-1|k-1) \quad , \quad \mathtt{P}(k|k-1) = \mathtt{F}\mathtt{P}(k-1|k-1)\mathtt{F}^\top + \mathtt{Q}(k) \quad . \tag{5.3}$$

Feature measurements are related to the state through

$$\mathbf{z} = \mathbf{h}(\mathbf{x}) + \mathbf{w}$$

where $\mathbf{w}$ is Gaussian measurement noise with zero mean and diagonal covariance $\mathtt{W}$ and $\mathbf{h}(\mathbf{s})$ is the projection of state into control point for image measurements. Equation 5.2 defines the measurement innovation vector as a stack of perpendicular distances between the predicted control point and the measured control point scalers denoted as $\mathbf{m}$. The image Jacobian $\nabla\mathbf{h}_i$ of $h_i(\mathbf{x})$ is defined in equation 5.2 as $\mathtt{a}_i$.

The control point displacements vector $\mathbf{m}$ is incorporated into the filter by the Kalman update equations

$$\hat{\mathbf{x}}(k|k) = \hat{\mathbf{x}}(k|k-1) + \mathtt{W}[\mathbf{m}(k)] \quad , \quad \mathtt{P}(k|k) = \mathtt{P}(k|k-1) - \mathtt{W}\mathtt{S}\mathtt{W}^\top \quad . \tag{5.4}$$

The Kalman gain and innovation covariance matrices are standard.

$$\mathtt{W} = \mathtt{P}(k|k-1)\mathtt{a}^\top\mathtt{S}^{-1} \quad , \quad \mathtt{S} = \mathtt{a}\mathtt{P}(k|k-1)\mathtt{a}^\top + \mathtt{R} \quad . \tag{5.5}$$

The innovation inverse, $\mathtt{S}^{-1}$ is robustly obtained using Singular Value Decomposition (SVD) to reduce the effects of rank deficiency.

The filter is tuned by collecting a short sequence of pose data with no filter to obtain a rough estimate of the parameters, then using the part-tuned filter to collect a longer spell of data and retuning, and so on. Moving the object via the teleoperated robot arm, we found no noticeable anisotropy in the three translation components, and similarly none in the

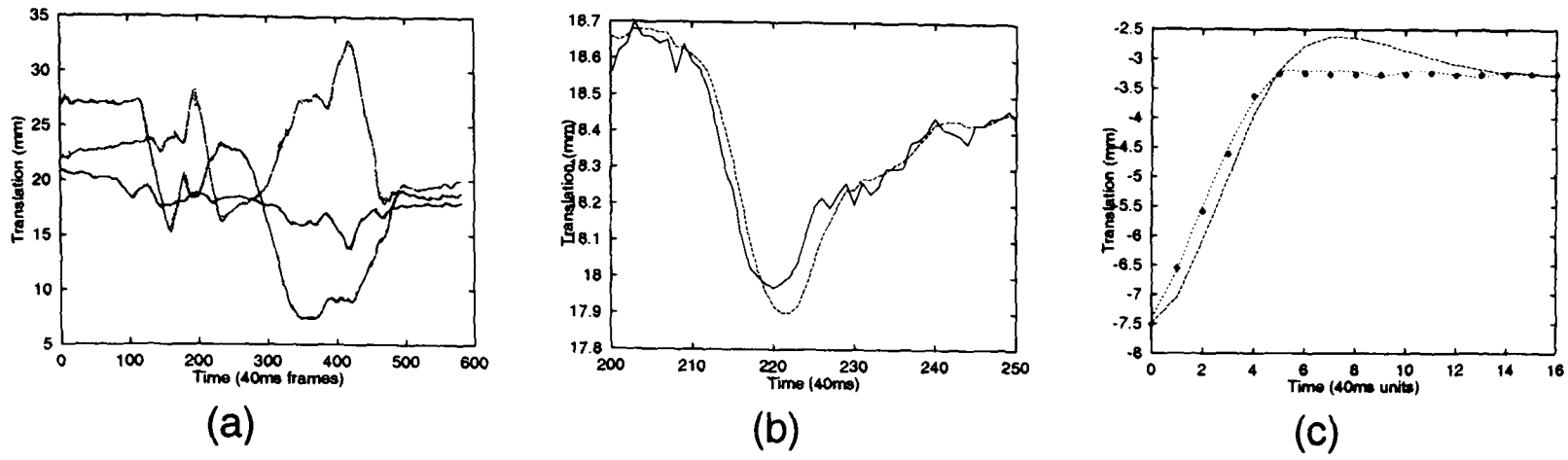


Figure 5.7: (a) Typical translation data used to tune the filter offline. (b) An enlargement, showing more clearly the filtered signal (dashed). (c) The response to a step change in translational velocity. The test object was driven by the robot arm at 20 mm·s^{-1} and stopped within 5 ms. The points are the robot's position and the dashed curve shows the response of the filter. The filter is only used for prediction, and the dotted curve shows the recovered pose after 3 iterations per 40 ms cycle of the modified Harris tracker.

three rotation components. The dynamic measurement noise for translation and rotation was set to $\pm$1mm and $\pm$0.1 rad, and the process noise given by $\sigma_t^2 = 0.04\text{mm}^2\cdot\text{frame}^{-4}$, and $\sigma_{\hat{a}}^2 = 5 \times 10^{-6}\text{rad}^2\cdot\text{frame}^{-4}$. The tolerable accelerations are therefore $\sim$ 120mm·s^{-2} and $\sim$ 1.4rad·s^{-2}. Figure 5.7(a) shows a section of the three components of translation used to tune the filter offline, and the enlargement in (b) shows more clearly the filtered signal (dashed).

It is important to stress that the filter is used only for predicting the starting pose for the iterated pose updating using the modified Harris tracker. The pose presented to the operator uses the final iterated pose, and is filtered more lightly [3]. This is done to avoid imposing the phase lag introduced by the tracking filter onto the operator's internal filtering processes. The benefit of so doing can be seen clearly in Figure 5.7(c). The figure shows the response to a step change in the robot arm's translational velocity. The test object was driven by the robot arm at 20 mm·s^{-1} and stopped within 5 ms. The marked points are determined using the robot, and the dashed curve shows the lagging response of the filter. The dotted curve shows the actual recovered pose using three iterations of the pose update cycle per image.

5.3.4 Camera parameter recalibration within the tracker

It was noted in the introduction to this chapter that Harris' tracker requires knowledge of the cameras' intrinsic and extrinsic parameters ie. calibrated cameras. In practise situations may arise where the calibrated parameters may be in err. For example, cameras may be nudged, may be erroneously calibrated or may have motorised zoom lens systems. The present tracking framework allows for these errors to be corrected by augmenting the state vector with the camera parameters of interest.

Camera parameter filtering

The camera state is given by

$$\mathbf{x}_c = [f_x \; f_y \; u_0 \; v_0 \; a_x \; a_y \; a_z \; t_x \; t_y \; t_z]^\top \; ,$$

and the overall state vector and state covariance is extended to incorporate the camera parameters

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_o \\ \mathbf{x}_c \end{bmatrix} \quad , \quad \mathsf{P} = \begin{bmatrix} \mathsf{P}_{oo} & P_{oc} \\ \mathsf{P}_{co} & P_{cc} \end{bmatrix} .$$

The Kalman prediction equations for object state and covariance are as before (equation 5.3). The prediction of the camera state, covariance and cross-covariance depend on the camera's process model. Here we assume that the camera parameters are held constant which results in the direct transfer of camera state to the predicted camera state

$$\hat{\mathbf{x}}_c(k|k-1) = \hat{\mathbf{x}}_c(k-1|k-1) \quad , \quad \mathsf{P}_{cc}(k|k-1) = \mathsf{P}_{cc}(k-1|k-1) \quad .$$

The predicted cross-variance matrix is

$$\mathsf{P}_{oc} = \mathsf{P}_{oc}^\top = \mathsf{F}\mathsf{P}_{oc} \; .$$

The Kalman update equations (equation 5.4) are as before however the image jacobian does not solely depend on a (equation 5.2) but rather on the concatenation of the image Jacobian with respect to object state and the image Jacobian with respect to the camera state $\nabla \mathbf{h}_c$

$$\nabla \mathbf{h} = [\mathsf{a} \;\; \nabla \mathbf{h}_c] \; .$$

$\nabla \mathbf{h}_c$ is obtained numerically using a finite difference approximation.

5.4 Implementation details

5.4.1 Camera calibration

The method detailed in chapter 4 is used for camera calibration.

5.4.2 Pose initialisation

Whilst the pose of the objects attached to the slave manipulator are initialised by knowledge of the robot's end-effector pose, unattached objects are manually initialised by the operator using a method given by Hébert and Faugeras [117].

The operator supplies the correspondence between pairs of image points $\mathbf{x}_a$ and $\mathbf{x}_b$ in cameras with projection matrices $\mathtt{P}_a$ and $\mathtt{P}_b$ and with the corresponding object location $\mathbf{X}_0^{\mathrm{o}}$. First the world point of the corresponding image points are obtained by a linear triangulation procedure. The equation set is formed from

$$\begin{bmatrix} (\mathtt{P}_{a20}x_a - \mathtt{P}_{a00}) & (\mathtt{P}_{a21}x_a - \mathtt{P}_{a01}) & (\mathtt{P}_{a22}x_a - \mathtt{P}_{a02}) \\ (\mathtt{P}_{a20}x_a - \mathtt{P}_{a10}) & (\mathtt{P}_{a21}x_a - \mathtt{P}_{a11}) & (\mathtt{P}_{a22}x_a - \mathtt{P}_{a12}) \\ (\mathtt{P}_{b20}x_b - \mathtt{P}_{b00}) & (\mathtt{P}_{b21}x_b - \mathtt{P}_{b01}) & (\mathtt{P}_{b22}x_a - \mathtt{P}_{b02}) \\ (\mathtt{P}_{b20}x_b - \mathtt{P}_{b10}) & (\mathtt{P}_{b21}x_b - \mathtt{P}_{b11}) & (\mathtt{P}_{b22}x_b - \mathtt{P}_{b12}) \end{bmatrix} \begin{bmatrix} X_x^{\mathrm{w}} \\ X_y^{\mathrm{w}} \\ X_z^{\mathrm{w}} \end{bmatrix} = - \begin{bmatrix} (\mathtt{P}_{a23}x_a - \mathtt{P}_{a03}) \\ (\mathtt{P}_{a23}y_a - \mathtt{P}_{a13}) \\ (\mathtt{P}_{b23}x_b - \mathtt{P}_{b03}) \\ (\mathtt{P}_{b23}y_b - \mathtt{P}_{b13}) \end{bmatrix},$$

and the homogeneous world coordinate is assembled as $\mathbf{X}^{\mathrm{w}} = [X_x^{\mathrm{w}}; X_y^{\mathrm{w}}; X_z^{\mathrm{w}}; 1]$. With $n \geq 3$ scene points $\mathbf{X}_0^{\mathrm{w}}, \mathbf{X}_1^{\mathrm{w}}, \mathbf{X}_2^{\mathrm{w}}, \cdots, \mathbf{X}_n^{\mathrm{w}}$ recovered, the $n-1$ unit vectors

$$\hat{\mathbf{n}}_i^{\mathrm{w}} = \frac{(\mathbf{X}_i^{\mathrm{w}} - \mathbf{X}_0^{\mathrm{w}})}{|\mathbf{X}_i^{\mathrm{w}} - \mathbf{X}_0^{\mathrm{w}}|}$$

are found, and the rotation that best maps the corresponding unit vectors on the object

$$\hat{\mathbf{n}}_i^{\mathrm{o}} = \frac{(\mathbf{X}_i^{\mathrm{o}} - \mathbf{X}_0^{\mathrm{o}})}{|\mathbf{X}_i^{\mathrm{o}} - \mathbf{X}_0^{\mathrm{o}}|}$$

onto them determined using the method of Hébert and Faugeras. One finds the unit quaternion $\mathring{\mathbf{q}}$ that minimises

$$\sum_i^{n-1} |\hat{\mathbf{n}}_i^{\mathrm{w}}\mathring{\mathbf{q}} - \mathring{\mathbf{q}}\hat{\mathbf{n}}_i^{\mathrm{o}}|^2$$

or equivalently

$$\mathring{\mathbf{q}}^{\top} \sum_i^{n-1} (\mathtt{A}^{\top}\mathtt{A})\mathring{\mathbf{q}}$$

where

$$\mathtt{A} = \begin{bmatrix} 0 & n_x - b_x & n_y - b_y & n_z - b_z \\ n_x - b_x & 0 & -n_z - b_z & n_y + b_y \\ n_y - b_y & n_z + b_z & 0 & -n_x - b_x \\ n_z - b_z & -n_y - b_y & n_x + b_x & 0 \end{bmatrix}$$

The required $\mathring{\mathrm{q}}$ is the unit eigenvector of $\mathtt{A}^\top\mathtt{A}$ that has the smallest eigenvalue.

The translation $\mathbf{t}$ is found from the center of mass of the world and object points.

$$\mathbf{t} = \bar{\mathbf{X}}^{\mathrm{w}} - \mathring{\mathrm{q}} \otimes \bar{\mathbf{X}}^{\mathrm{o}} \otimes \mathring{\mathrm{q}}^*$$

5.4.3 Visibility calculations

The basic geometry of a polyhedral object model is specified by a list of vertex positions, with a list of faces described by their bounding vertices. The control lines are then listed explicitly, each with a number of control points whose positions along the lines are described by a parameter ζ between 0 and 1. If a control line is not a crease edge between faces, it must be a marking on a face. The face is specified so that a control point can never be hidden by its own face.

To be deemed visible at all, a control line must be associated with at least one face whose surface normal has a z-component pointing towards the camera. (However, faces which point forward, but only just, are deemed not visible.) Then the ranges of the parameter ζ are found for which the line is not occluded by all faces other than those one or two with which the line is associated. Only if the control point parameter is well within a visible range is that point used.

5.4.4 Measurement Gating

Each measurement is gated by adjusting the search window radius to $\gamma\sqrt{\mathtt{S}}$. $\mathtt{S}$ is the innovation covariance matrix defined in equation 5.5 and γ is obtained from chi-squared distribution tables and is set by specifying a measurement capture likelihood. $\gamma = 3$ corresponds to a likelihood of 99.7% of the true feature falling in the validation region and is used here.

5.4.5 General

The system has been implemented for polyhedral objects viewed by three cameras. Monochrome video streams are synchronised and captured into the RGB planes of a PCI-bus colour framegrabber. On a single 400 MHz processor it is just possible within the 40ms interframe time to handle the image search for some 100 control points, the visibility calculations for some 20 potentially obscuring faces, filtering, robust collinearity and robust pose.

5.5 Use within teleoperation

5.5.1 Example 1: A manipulation task

The first example of the use of the tracker within teleoperation is a manipulation task in which a peg is inserted into a hole. To ensure object track for the entire operator motion, the time constant T_p of the master-slave forward-position filter, a first-order pole with unity gain

$$F(s) = \frac{1}{T_p s + 1}$$

was chosen to achieve a bandwidth of 1Hz. In normal use the filter is present to prevent resonances from high-frequency structural modes, that would otherwise be fed around the loop, thereby destabilizing the system and T_p is selected to achieve a bandwidth of 2.5Hz (Daniel and McAree [7]).

Figure 5.8 shows views of the recovered wireframes of a peg and hole during two insertion runs. Both peg and hole are tracked. The leftmost views illustrate the commonly used operator tactic of using a view from above to correct for misalignment in up to five degrees of freedom, and an orthogonal side view for correcting the final one, the vertical displacement. We find that the operator uses this view to take the peg to the top of the hole, and then reverts to the top view for the final insertion. The middle montage of views shows a second tactic of tipping the peg into the hole. On the right are views around the peg and hole after insertion. The obvious misalignment is real, and not systematic noise — the peg has sufficient clearance around it when in the hole to twist by several degrees.

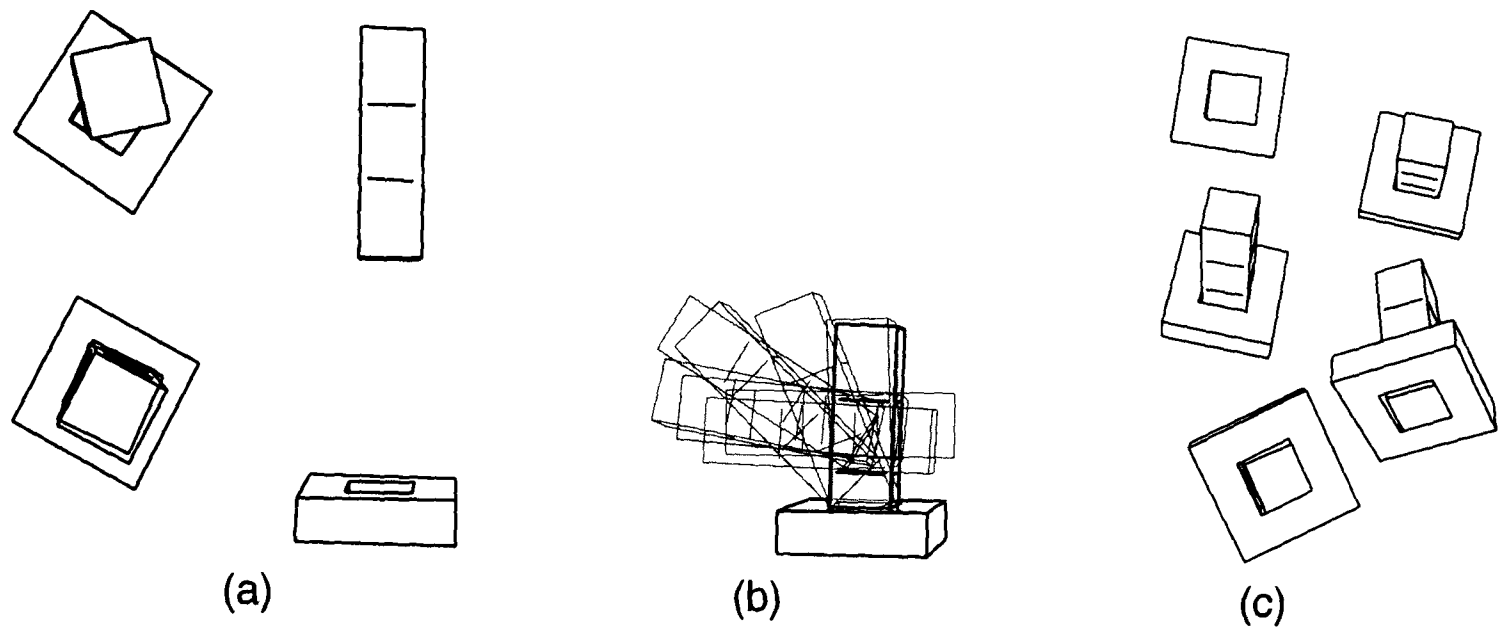


Figure 5.8: (a) A successful tactic is to use the top then side then top view. (b) Another is tipping the peg. (c) Views once the peg is inserted.

5.5.2 Example 2: Impact Control to Facilitate Teleoperation

In many autonomous and teleoperated robotic tasks, the robot end-effector must come into contact with its environment, creating a high risk of damage when fast, non-compliant robots move in stiff, uncertain surroundings. Some workers (eg. Mills [118]) have proposed binary controllers, adopting position control for space motions and switching to force control once contact is established. However, such two-mode methods fail to suppress the impact, and a number of workers have proposed additional control phases as contact approaches. Khatib and Burdick [119] for example proposed an intervening phase in which the velocity is damped prior to impact, and Li and Daniel [120, 121] introduced a fourth stage during which the end-effector velocity was limited immediately prior to contact by exploiting a special property of the force control law. Their free space controller was able to blend into a contact controller.

In more recent work McAree and Daniel [122] developed a transition controller based on a linear quadratic regulator which minimized the effects of impact using measurements of the distance between the slave and objects in its workspace supplied by the tracker developed in this chapter. The controller formulation and its use to facilitate teleoperation is explored in the following sections.

Statistics of impact; Controller Formulation

The controller proposed by McAree and Daniel [122] minimised a cost function which ascribes a cost to the effects of impacts. A basic premise of their work was that it is the transient of impact that is responsible for operator discomfort. Figure 5.9 shows the actual response for the Puma 560 manipulator at impact.

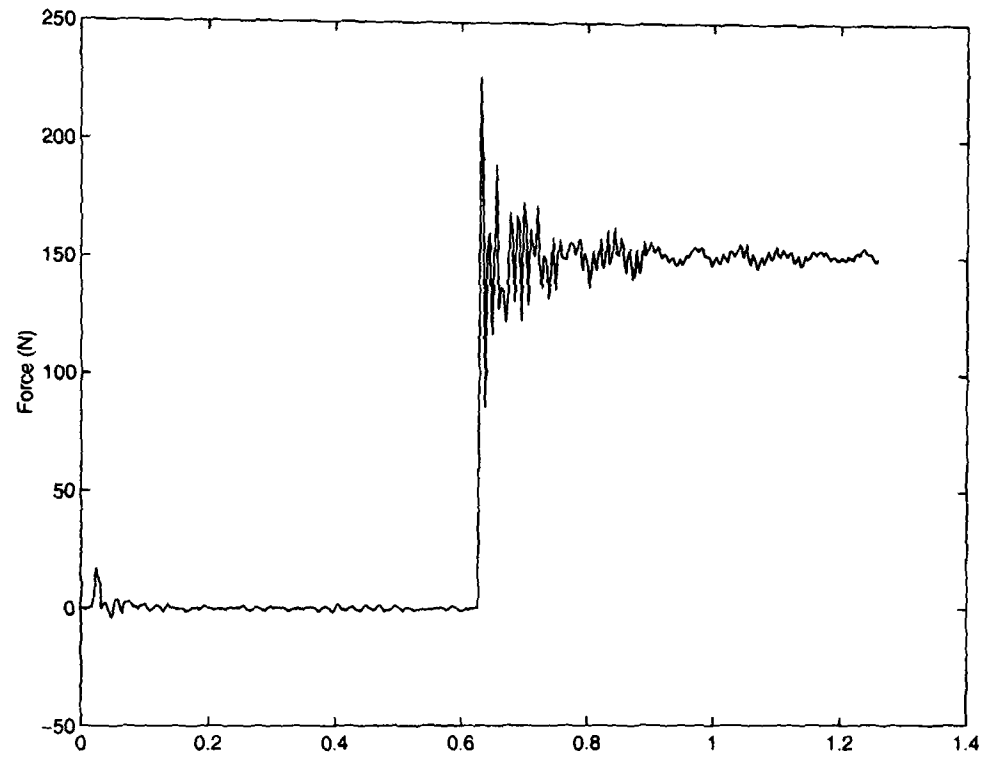


Figure 5.9: Force response of a Puma 560 manipulator on impact with a stiff environment at a velocity of 0.2ms^{-1}. The impact transient causes instability hence the aim of the impact controller is to suppress impact transients whilst trading off speed to contact.

Ideally the impact should occur without transient, but this is only possible if the actuators apply an appropriately sized impulse at the instant of impact to instantaneously change the slave's velocity to the steady-state in-contact velocity. On a realistic slave arm several factors work against *pulsing* the slave to cancel out the impact transient, firstly the actuators are limited in the rate at which they can deliver power, secondly the approach requires an accurate model of the slave and environment dynamics and thirdly the time of impact needs to be judged precisely. This does not occur in practise and it is the operator who takes the brunt of the impact. The controller formulation aims to reduce the size of the impact transient to lessen the effect of impact on the operator. The impact cost function is composed of the deviation of slave arm away from the nominal or transient trajectory.

Some insight into the relationship between impact velocity and transferred energy comes

by considering Figure 5.10 which abstracts the dynamics of a force reflecting system by a model comprising two mass-spring-damper systems. The system on the left hand side of Figure 5.10 represents the combined dynamics of the operator and the master robot. The system on the right represents the slave robot moving towards a stiff environment. Both systems are under PD control with the slave's effective free space stiffness denoted by K_s and damping by C_s and the master's effective stiffness by K_m and damping by C_m. The input x_d is the operator's intellectual demand; x_m is the motion of the master; x_s the command to the slave; x_e is the motion of the slave (end-effector motion); the environment is located at x_i.

To couple the master and slave, the position of the master is fed-forward to the slave arm while the measured slave contact force is fed-back to the master (ie. *force-position* teleoperation; Brooks [123]).

Considering the slave's decoupled in-contact equation of motion

$$M_s\ddot{x}_e + C_s\dot{x}_e + K_s(x_e - x_s) + K_e(x_e - x_i) = 0$$

The controller cost function costs the deviation of the slave path away from the nominal

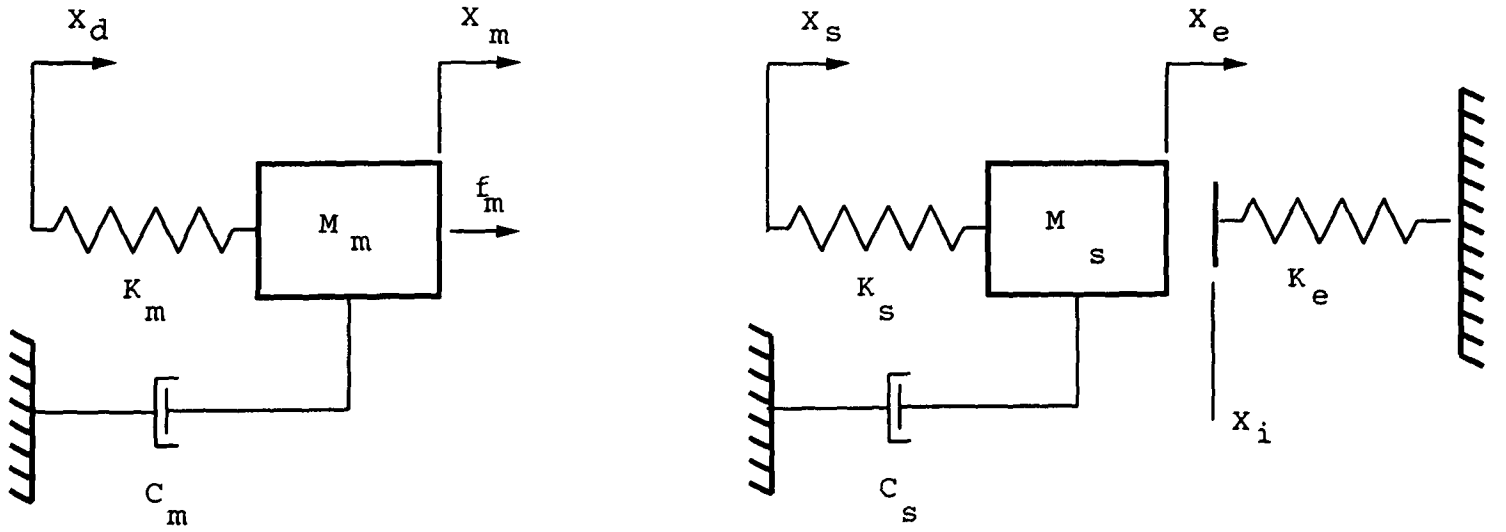


Figure 5.10: The dynamics of force-position teleoperation abstracted as two masses both under proportional plus derivative control.

(or transient free) path, x_n.

$$J = \int_0^{t(x_e=x_i)} [x_e(t) - x_n(t)]^2 dt + \mu \int_{t(x_e=x_i)}^{\infty} [x_e(t) - x_n(t)]^2 dt \,. \qquad (5.6)$$

The (unit-less) parameter μ weights free-space deviation (first integral) against in-contact deviation (second integral) with a view toward discouraging excessively large controller effort following impact. Defining v_{ss} as the steady-state in-contact velocity then the second integral of equation 5.6 is evaluated using the two mass model and Parseval's theorem resulting in

$$J = \int_0^{l:x_e=x_i} [x_e(t) - x_n(t)]^2 dt + \lambda(v_{ss} - \dot{x}_e|_{t(x=x_i)})^2 \,,$$

where μ has been combined with $\frac{M_s^2}{C_s(K_s+K_e)}$ into a new parameter λ. Thus by combining, from first principles, the combination of costing the square deviation of the measured and nominal trajectory and a two mass master-slave model results in a cost function which, intuitively, trades off the fidelity of free-space pre-impact motion against a reduction in impact speed and hence an increase in time to contact.

The position of the environment x_i is measured by the tracker and the probability density function $f_i(x_i)$ describes the sensor noise process. The pdf is assumed Gaussian with mean x_i and variance σ_i^2, which is calibrated in Section 5.5.2.

The expected cost of impact $E[J]$ is

$$E[J] = \int_0^{\infty} f_i(x_i) J dx_i \,.$$

Finding the slave motion which minimises the expected cost (MEC) of impact results in the following equations of motion:

$$\dot{x}_e(t) = \sqrt[3]{\frac{[x_e(t) - x_n(t)]^2 - c}{2\lambda f_e[x_e(t)]}} \,,$$

$$x_e(t_0) = x_n(t_0)$$

where $c = -2\lambda f_e[x_e(t_0)]\dot{x}_e(t_0)^3$ and taking $f_i(x_i)$ as Gaussian with variance σ_i^2, the likelihood of impact function ie. the proximity of the slave end-effector to the environment, $f_e(x_e)$ is obtained by a process of truncation and re-scaling on $f_i(x_i)$.

$$f_e[z(t')] = \begin{cases} \sqrt{\frac{2}{\pi}} \frac{\exp(\frac{-z(t')^2}{2})}{1-\mathrm{erf}(\frac{z(t')}{\sqrt{2}})}, & t > t', \\ 0, & t \leq t', \end{cases}$$

where $z(t') = \frac{x_e(t')-x_i}{\sigma_i}$ and erf is the error function which evaluates the standard normalised probability integral.

MEC Trajectory Simulation

Figure 5.11 shows the simulated MEC trajectories with different weighting λ and sensor variance σ_i^2. The initial speed of the end-effector is $\dot{x}_e = 0.2$m/s and the environment is located at $x_i = 0.15$m. The dashed line in Figure 5.11(a) is the unimpeded trajectory corresponding to $\lambda = 0$, the larger the value of the weighting λ the slower the slave at impact. Similarly with sensor variance σ_i^2; for $\sigma_i^2 = 0$ the moment of contact is known perfectly and the MEC trajectory instantaneously changes velocity at impact, the larger the value of sensor variance the more "conservative" the controller becomes and the slower the slave at impact.

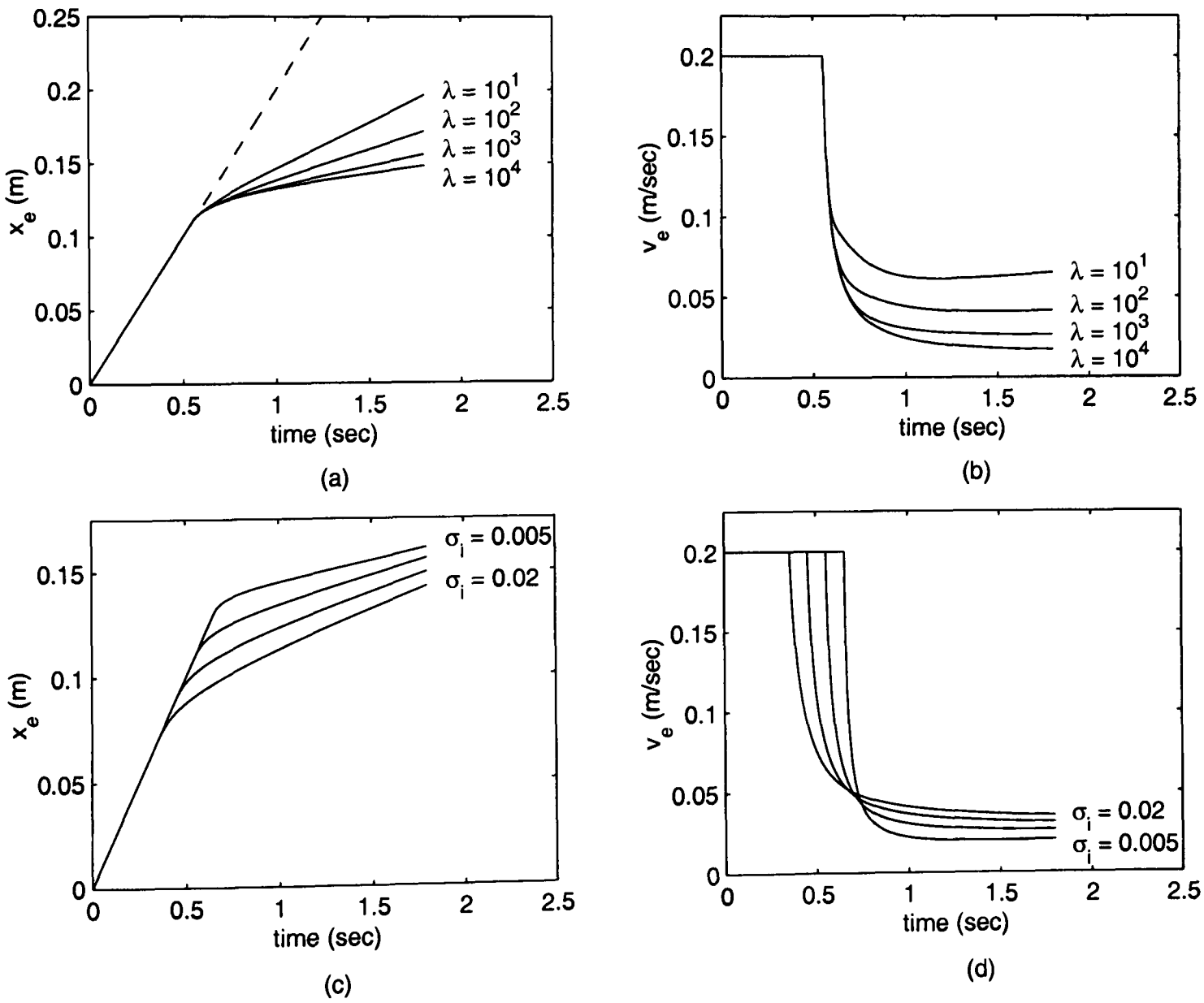


Figure 5.11: The MEC trajectories with different weighting λ and sensor variance σ_i^2. (a) position and (b) velocity for $\lambda = [10^1, 10^2, 10^3, 10^4]$ with constant $\sigma_i = 0.01$m. (c) position and (d) velocity trajectories for $\sigma_i = [0.005, 0.01, 0.015, 0.02]$m with $\lambda = 1000$. The command velocity is $\dot{x}_e = 0.2$m/s and the environment located at $x_i = 0.15$m.

Impact Control using Vision

The impact controller was first calibrated by estimating the error of the visual proximity signal σ_i over some 2000 seconds. Figure 5.12 shows a histogram plot of measurement error for a stationary object. The sensor is well approximated by a Gaussian distribution of $\sigma_i = 0.5$mm.

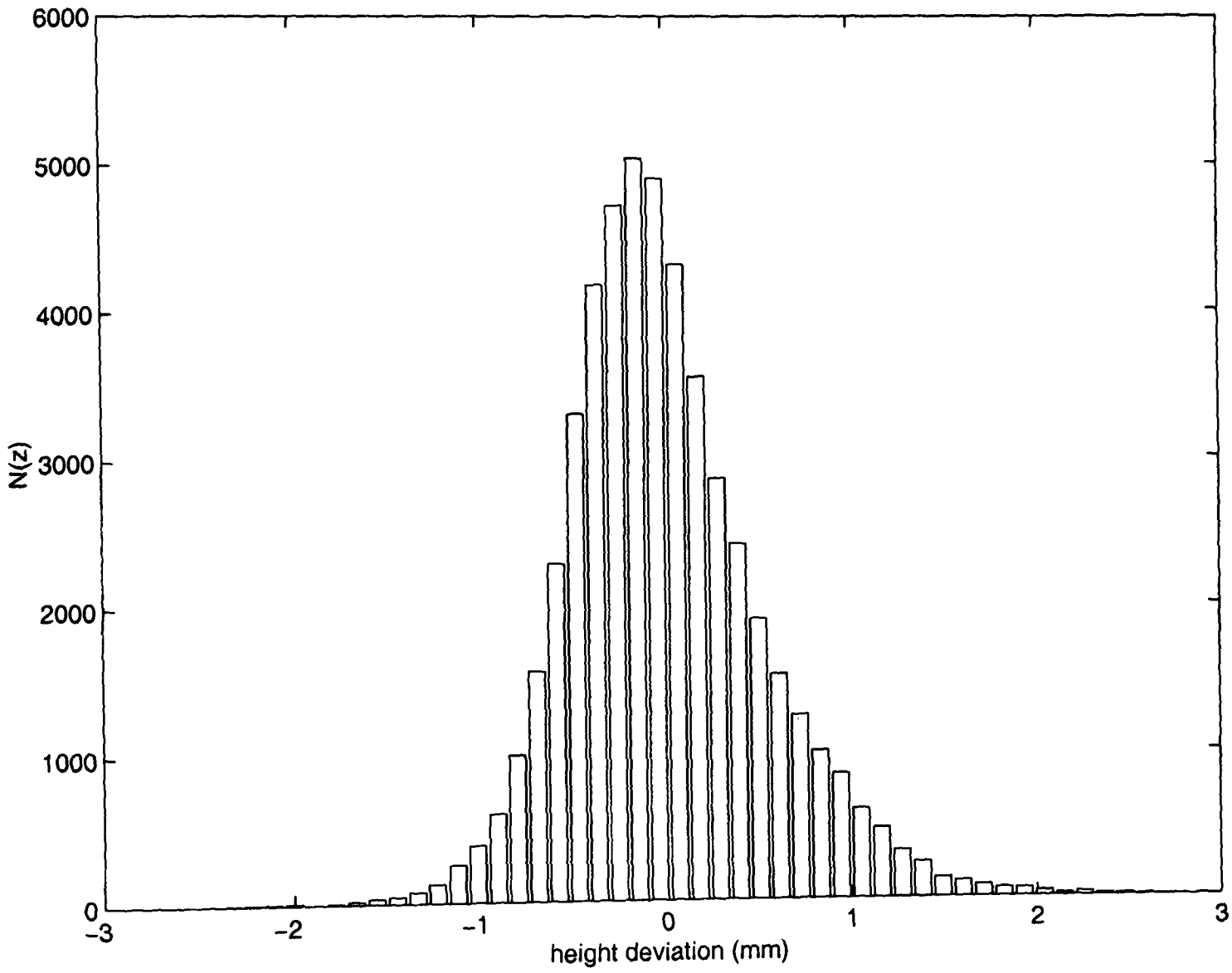


Figure 5.12: Histogram of z deviation for an object mounted on the slave-arm end-effector collected over some 2000 seconds at frame-rate (25Hz). Slight skewing of the distribution is evident but the form overall is well approximated by a Gaussian distribution of $\sigma_i = 0.5$mm.

The experimental setup is shown in Figure 5.13(a). The robot arm drives the upper block into the lower one at 20 mm·s^{-1}, and when the force sensor measurement reaches 50N the robot is commanded to stop. Impacts are made with and without visual feedback. Figure 5.13(b) shows the temporal variation of force with no visual proximity detection.

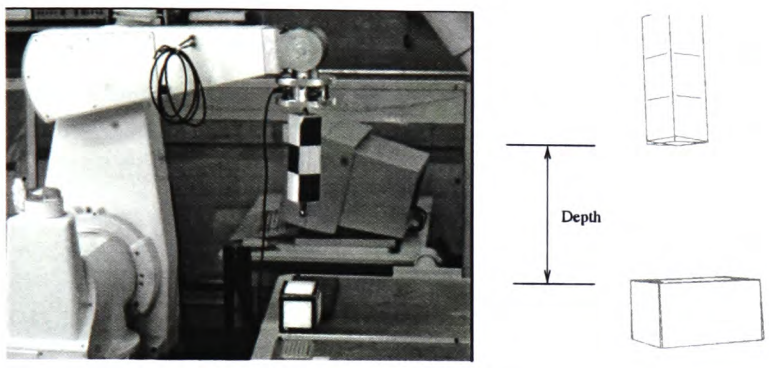
In each of ten impact experiments, the final force consistently overshoots, reaching some 150 N. The results in Figures 5.13(c and d) were obtained using proximity information recovered by tracking both blocks. In (c) a deceleration phase begins 2 seconds before predicted impact, and the final force consistently reaches a plateau at the desired 50 N. In (d) however, braking begins only 1 second before predicted impact. This requires the arm to decelerate at its limiting value and, in some 30% of the trials, the final force is too large. (The various velocity and accelerations limits in the experiments are imposed in software to prevent damage to equipment.)

5.6 Future Work

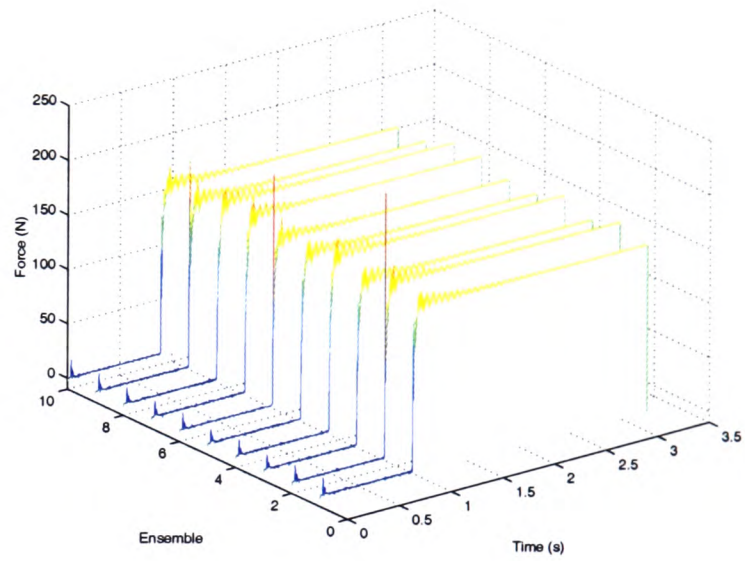
Although the tendency of the tracker to be upset by mismatches has been reduced using two stages of robust outlier detection the tracker still fails when the operator is performing a manipulation manoeuvre at the very limits of manipulation. Section 3.2.1 comments that a bandwidth of 2.5Hz is characteristic of the operator's arm motions. This figure has been reported by a number of authors (Daniel and McAree [7], Harris and Wolpert [64]). Possible improvements to increase the combined teleoperation and tracking robustness are threefold; firstly, the operator's forward position commands could be retarded at high frequencies. Satisfactory results have been obtained by prefiltering the master to slave's forward position signal with a low-pass filter of 1.0Hz bandwidth (Section 5.5.1) and some teleoperation systems due to design limitations are indeed limited to 1Hz (Daniel [124]). Secondly, the tracking rate could be upgraded from frame (25Hz) to field rate (50Hz) provided the computational resources could be made available. Thirdly, robustness could be increased by intelligent feature search algorithms such as those proposed by Lowe [37].

5.6.1 Improving Robustness Through Feature Saliency Measures

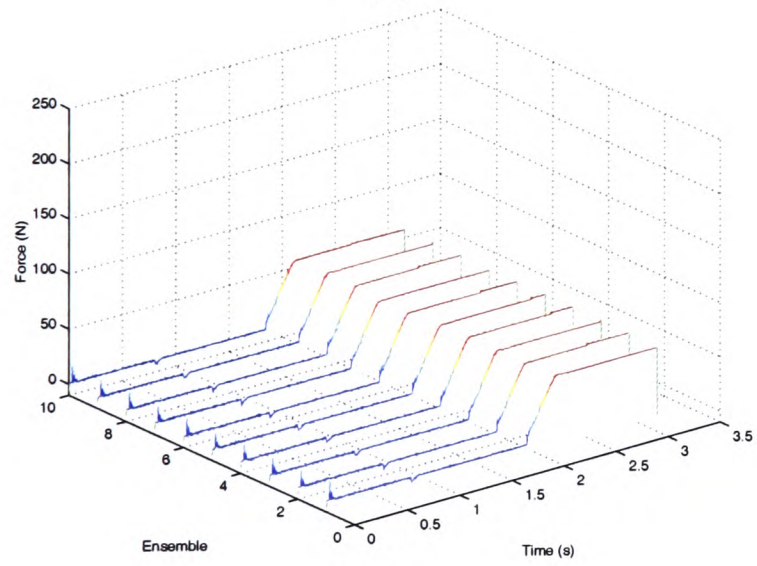
Within the tracker detailed in this chapter, image measurements are modelled as being drawn from a Gaussian distribution. A consequence of this Gaussian assumption is that feature mismatches even if they occur at a few σ from the true feature location will considerably skew the pose estimate. This will often result in tracker failure and motivated the use of robust methods throughout this chapter. A complimentary robustness layer which



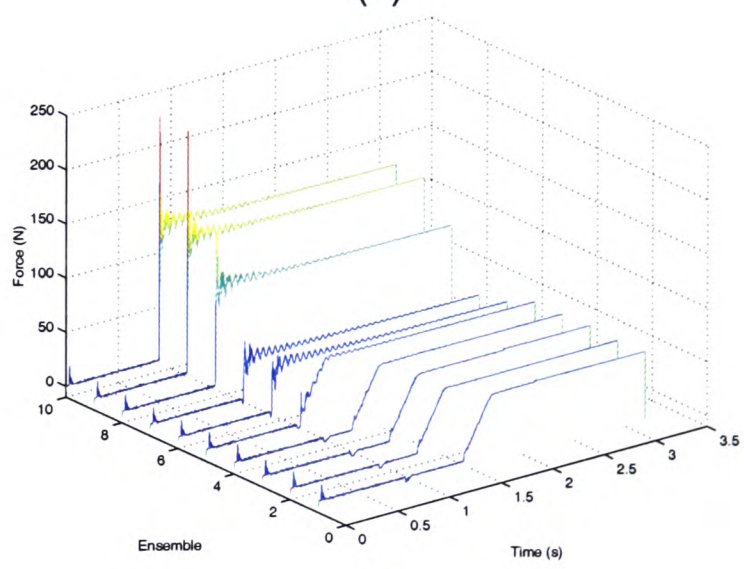
(a)



(b)



(c)



(d)

Figure 5.13: (a) The impact experiment. The upper block is driven downwards to impact upon the lower, and the arm commanded to stop when the force reaches 50N. (b) In 10 runs with no impact prediction the final force is well in excess of 50N. (c) Using impact prediction the arm is decelerated in good time, and the arm is then able to push down under force control until the 50N target is reached. (d) If the deceleration phase starts at the limit, the force is sometimes exceeded. (The experiments here are sorted in order of increasing final force, not in the order of performance.)

seems suited to environments, such as within teleoperation, where it is possible to adorn objects with markers has been described by Lowe [37].

Lowe [37] noted that the chance of feature mismatch is lower for rare, or salient, image features and can be calculated using Bayes rule. First the following events are defined:

Event	Description
ν	innovation scaler ν is obtained
fg	innovation originated from corresponding scene point (foreground)
bg	innovation originated from background clutter

Feature saliency can be stated as the conditional probability of the innovation originated from the correct scene point given the innovation, $P(\text{fg}|\nu)$. Bayes rule transforms this conditional into known probabilities

$$P(\text{fg}|\nu) = \frac{P(\nu|\text{fg})P(\text{fg})}{P(\nu)} .$$

In our work the combined likelihood and prior probability distributions $P(\nu|\text{fg})P(\text{fg})$ is given by the innovation distribution which is Gaussian with mean $\mathbf{m}$ and covariance $\mathbf{S}$ (Section 5.3.3). The denominator or normalising term $P(\nu)$ is the combined probability that the innovation originated from either the model or background clutter

$$P(\nu) = P(\nu|\text{fg})P(\text{fg}) + P(\nu|\text{bg})P(\text{bg}) .$$

The background distribution $P(\nu|\text{bg})P(\text{bg})$ or clutter model (Lowe [37]; Isard and Blake [43]) is commonly assumed to be uniform with density λ — low λ corresponding to an image feature with high saliency.

Once this score has been calculated for each image measurement, measurements are incorporated into the tracker according to the adopted robustness strategy. Lowe [37], for example, used a greedy strategy to incorporate the single image measurement with the highest saliency score. This procedure was iterated a set number of times for each image frame.

This approach offers further robustness gains for environments, such as ours, where the appearance of the tracked objects and surroundings can be easily altered to generate high saliency image features.

5.7 Conclusions

This chapter has described a system for the recovery in real-time of the 3D pose of moving objects using the extension to multiple cameras of the method of Harris [20]. The tendency of the tracker to be upset by mismatches has been reduced using two stages of robust outlier detection. The chapter has quantified the characteristics of the system's static accuracy, convergence, and dynamic performance, and have illustrated the system's utility in a manipulation task. Given the lower accuracy of our object modelling, our results for static accuracy agree with those obtained by Maitland and Harris. The only issue where we would not concur with the conclusions of [20, 108, 111] is in the need to perform multiple iterations of the pose updating. We find that multiple iterations within a single image, before treating the pose as new measurement to the filter, help to reduce lag.

This chapter has also investigated how information derived from the visual sensor can ease operator burden in force reflection teleoperation through impact control. A consequence of feed forwarding visual information to the impact controller is that the magnitude of impacts reflected to the operator is reduced and hence the force reflection ratio can be increased, thus improving force fidelity.

The use of a Kalman filter as the data fusing mechanism within the tracker raises the fundamental issue of the trade-off between pose sensor noise and bandwidth before display to the operator. Experiments to assess the impact of filter tuning for optimal manipulation is discussed in the next chapter.

6

Operator Modelling

This chapter details two sets of operator experiments which have been performed using the system developed throughout this thesis. The purpose of these experiments is i) to aid in the understanding of the characteristics of sensors needed to perform a dynamic manipulation task, inparticular concentrating on aspects important to the method of visual pose tracking presented in chapter 5, and ii) to propose an information processing/representation model of human manipulation. The second set of experiments explores with how visual biases impede operators performing manipulation type tasks, and is detailed in the Section 6.6.

The first set of experiments proposes alternative models of the operator's performance during a visually-guided insertion task using a tele-operated robot, and develop an argument to test whether the performance is consistent with the operator's visuo-motor control loop maintaining a state model of the process as in a Kalman filter.

Operator performance is measured as various degrees of noise are added to the pose and as various amounts of smoothing are applied before passing to a visual display. With relatively high noise added to the pose measurements, best operator performance is achieved when they are low-pass filtered to a frequency comparable with the natural frequency at which the operator interacts with the environment.

The second set of experiments investigate the effect of biases on operator performance.

6.1 Introduction : Noise and Bandwidth Experiments

The previous chapter, detailing the visual pose tracking system, have raised the issue over the choice of passing raw unfiltered or filtered pose measurements to the operator. When the operator views raw unfiltered pose measurements the objects are responsive to manipulation however they constantly appear to jiggle or shake even when they are stationary. Alternatively when the operator views filtered pose measurements the objects appear static when stationary however the objects are sluggish when manipulated. There is a trade-off between agility, or bandwidth, and noise.

To clarify what is meant by "raw", Figure 6.1 show the output of one translational pose parameter recovered by our pose tracker as an object is moved with constant velocity and brought to rest. The marked points are from the robot kinematics, and the lagging curve is the output of a Kalman filter that was tuned by experiment to maximize tracking robustness over prolonged periods. On receipt of a new image, the filter is used to predict a starting pose for iterative refinement to a raw measured value, generating the prompt curve in the figure. This value is then incorporated in the filter as a measurement. Although the filtered curve is robust for automated tracking, there is no reason to suppose that the operator likes looking at the virtual view generated using it. The operator may prefer to see the raw measurements, or raw measurements after filtering using some other bandwidth. By experiment, we show that when noisy, sampled pose is used to generate a virtual view of the object for display to the operator, optimal performance for a part insertion task is achieved when the pose measurements are low-pass filtered to a frequency comparable with the frequency at which the operator interacts with the environment — a frequency determined by the coupled chain of operator's arm, master input device and slave robotic arm, whose predominant dynamics have been set to match those of the human operator's arm.

At a practical level in this chapter, we ask and answer the question illustrated in Figure 6.2: what is an appropriate method of treating pose measurements prior to passing to the graphics renderer for display to the operator? This could be re-phrased as: do operators perform better when the pose measurements are displayed filtered or raw?

However, we are also concerned to use our observations to assess the validity of two

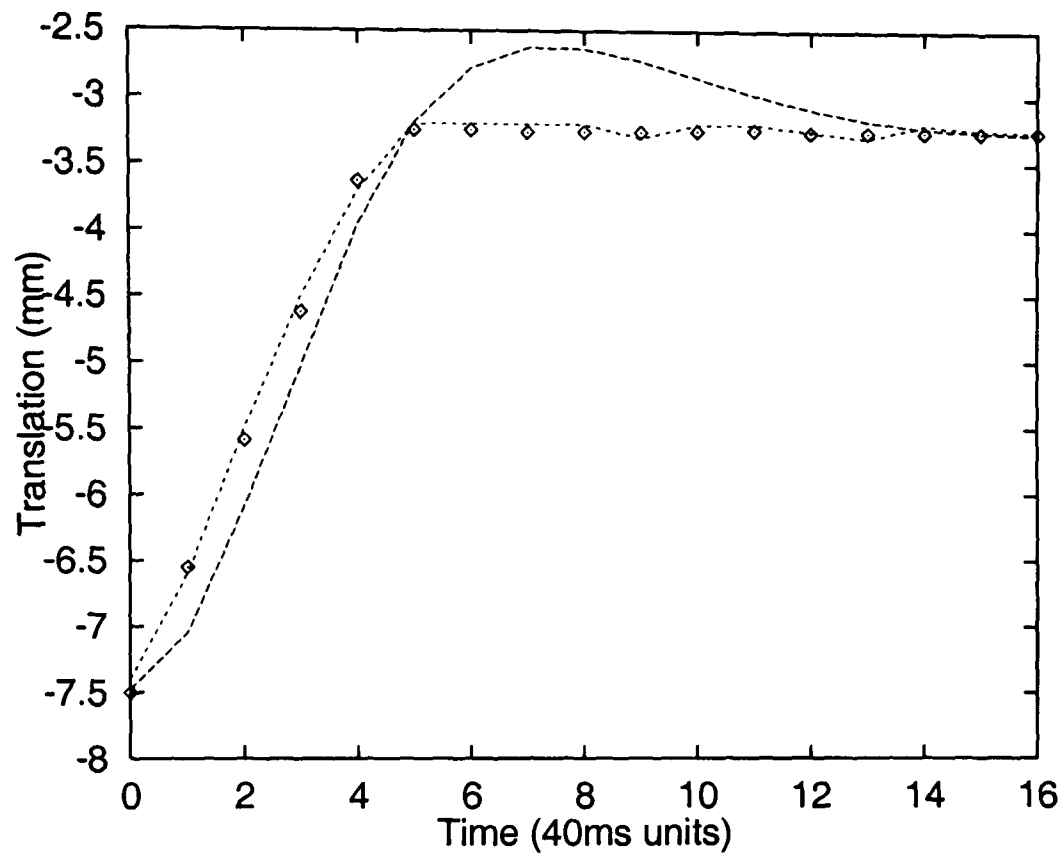


Figure 6.1: The filtered pose (lagging curve) gives better endurance to the visual object tracker over time. The raw pose measurements (prompt curve) are obtained using the filter to predict then using iteration on the current frame to refine. These are more faithful to the actual pose measurements (points), but are noisier, indeed sometimes much noisier than this example.

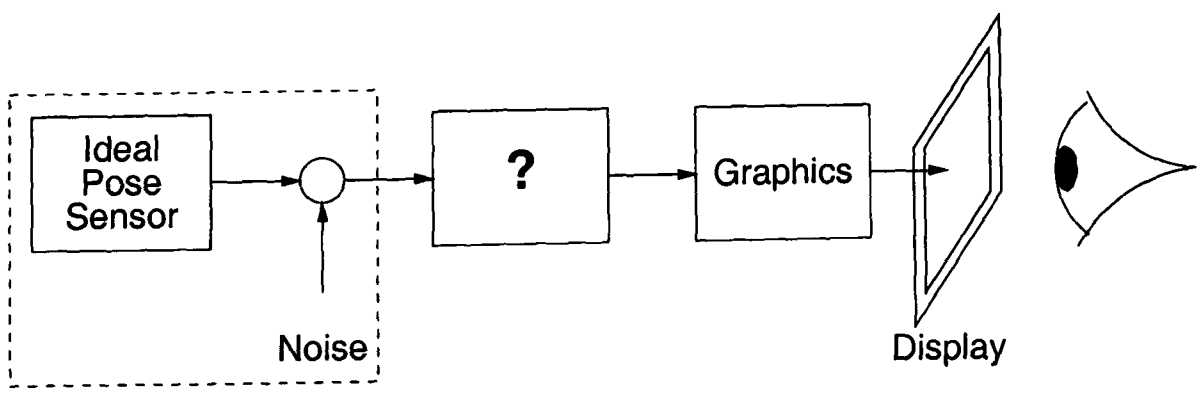


Figure 6.2: The question addressed in the chapter is how should noisy pose measurements be treated prior to rendering and display to the operator.

models of operator performance that we develop below. Those involved in the robotic realization of tele-operative systems have been able to produce a deal of good empirical evidence on operator *performance* (Massimino and Sheridan [65], Posner [15]), but have been reluctant to develop agreed ways of representing the operator in terms of models of human response developed by colleagues in psychophysics and neuro-physiology (perhaps understandably so, given the size and complexity of the relevant literature).

Our observations allow us to falsify, at least for the insertion task described later, a model of human performance that has the operator running an internal filter in his or her head to maintain and update a state representation of the task, a model akin to that suggested by McRuer (1980). Instead, the observations appear more consistent with an operator whose aim is to minimize variance in pose at the conclusion of a free space ballistic movement the purpose of which is to achieve close proximity between the end-effector and a target. This has similarities with the work of Harris and Wolpert [64] who used a minimum variance criterion to synthesize trajectories for human arm and eye movements during goal-directed reaching and saccadic redirections of gaze, respectively, and found them to correspond well with those observed in practice. They also used it with success to account for the trade-off between speed and accuracy observed by Fitt [125] in his well-known tapping test. However, in our work minimum variance is achieved using a simple adaptive gain controller *without* an internal state.

In Section 6.2 our hypothesized operator models are detailed and expressions for expected operator performance under various noise and filter bandwidths are developed. Section 6.3 describes the experimental design, and Section 6.4 shows the results obtained for operator performance, and analyzes them using statistical techniques. Finally conclusions are drawn from the empirical results about the design of pose display systems.

6.2 Operator Models

Human sensori-motor responses are represented as a combination of sensor modules, computational modules with particular transfer functions, and motor actuation modules. While the inputs to and outputs from the sensors and motors are quite well determined, much more uncertainty shrouds the computation in between — in essence the control methodol-

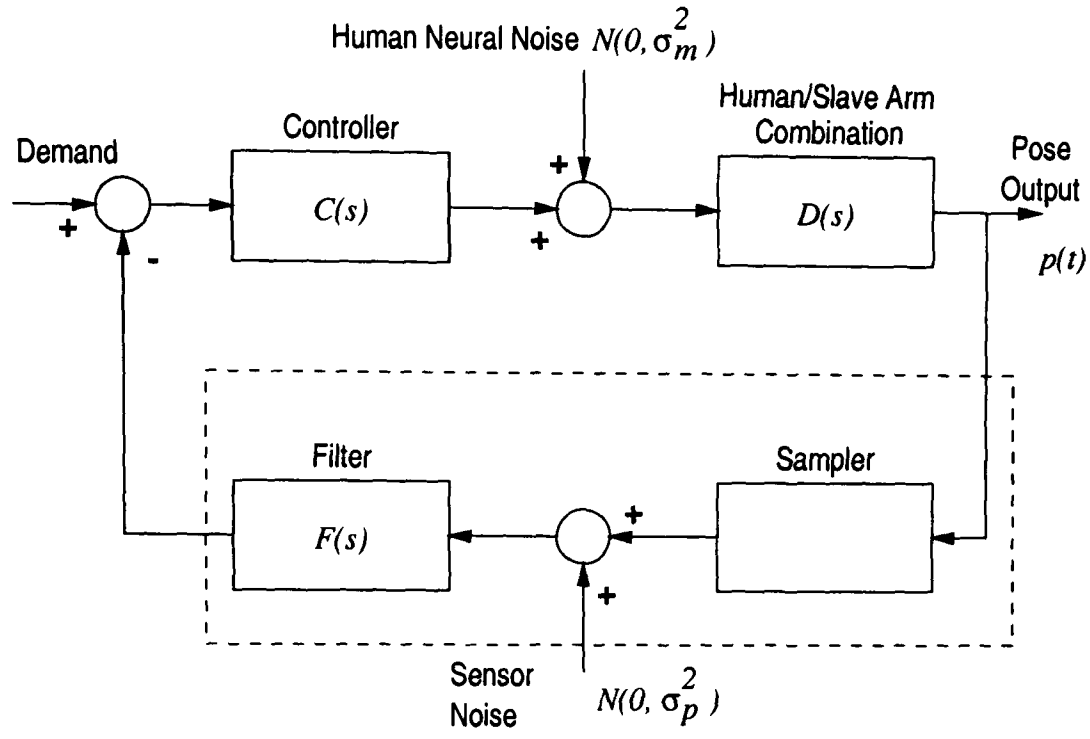


Figure 6.3: A model of the operator as within a teleoperation system using pose feedback. The human operator consists of a controller driving a neuromuscular actuation system. Motor noise is due to variances of the neural control signal and also the uncertainty in arm dynamics. The output is the pose of the controlled object, which is corrupted by noise and filtered before fed back to the operator.

ogy adopted by the central nervous system is unclear, the more so as it may adapt as the task or sensor data change.

A control block diagram of the operator manipulating an object with pose feedback is shown in Figure 6.3. The controller with transfer function $C(s)$ is of particular interest, but for both of the models we wish to develop an expression not so much for $C(s)$ itself, but rather for the cost of control (reflected in operator performance) as a function of the parameters we vary in the treatment of pose data — the bandwidth B of the filter $F(s)$ and noise N controlled by the variance σ_p^2. The cost will be based on the variance in the output in response to a null demand with white noise disturbances both in the pose measurements and in the input to the muscular plant.

To the neuronal signal output from the controller, we add Gaussian motor noise, and this drives the human arm and hence the slave arm. The time to complete the peg-in-hole insertion task is dominated by free space motion where there is no contact [64], and $D(s)$ is modelled as a second order critically damped system. The output of the slave arm is the pose $p(t)$. The pose is measured by a computer vision system which samples, adds Gaussian sensor noise and filters the corrupted signal, before feeding it back to the

controller.

6.2.1 The operator as an optimal stochastic controller

Our first hypothesized model is that the operator behaves as an optimal stochastic controller. Optimal stochastic controllers have two parts, as shown in Figure 6.4(a)). The first of these optimally estimates the current internal state $\mathbf{x}$ from noisy measurements $\mathbf{y}$ using Bayesian methods, and the second takes (potentially at least) the full set of conditional probabilities $p(\mathbf{x}|\mathbf{y})$ to control the plant. For a linear system with Gaussian noise in the measurements the optimal stochastic controller would be a Kalman Filter. Unfortunately, there is little chance that the operator is linear and we cannot hope to second guess what makes up the internal state $\mathbf{x}$.

However, we *can* make statements about the information throughput to the operator. Estimation theory (eg. [126]) shows that the optimal estimator produces an estimate whose covariance matrix reaches the Cramér-Rao lower bound, the inverse of the Fisher information matrix, when all the information available in the observations is used.

But we argue that maximizing the information available to the operator/controller is synonymous with maximizing the channel capacity K *into* the operator. Now according to Shannon's formula (eg. [126]) a Gaussian channel with a bandwidth B, signal S and noise N, has an information capacity in bits per second of

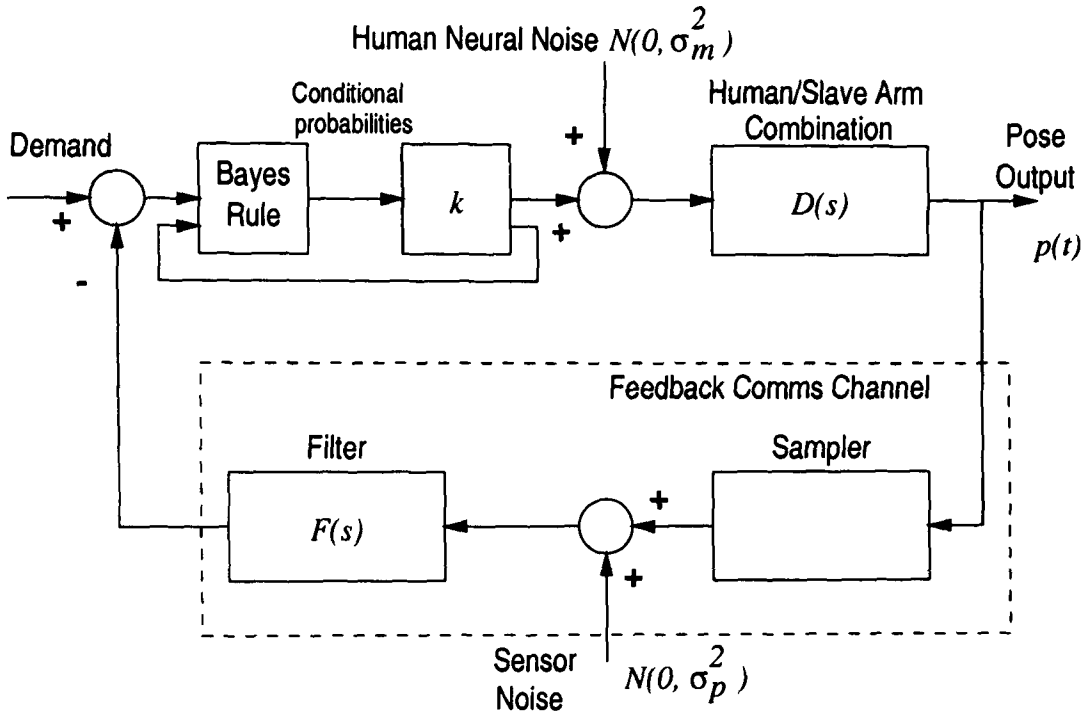
$$K = B \log_2(1 + \frac{S}{N}).$$

But if K is the information rate into the operator, then that operator's time to complete a task should vary as the inverse, $1/K$, or at least as some monotonic function of the inverse.

The pose information offered to the operator has two parts — signal and additive noise. The incoming pose *signal* has bandwidth commensurate with the bandwidth of manipulation, say $f_S = 2.5$ Hz; but the noise has a bandwidth, f_N, half that of the sampling frequency of the pose sensor. The noise added is drawn from a zero mean Gaussian of variance σ_{p}^2. The noise and signal is then filtered by low pass filter $F(s)$

$$F(s) = \frac{1}{1 + \tau s}$$

τ is related to the bandwidth of the combined effects of the filter and the bandlimited white



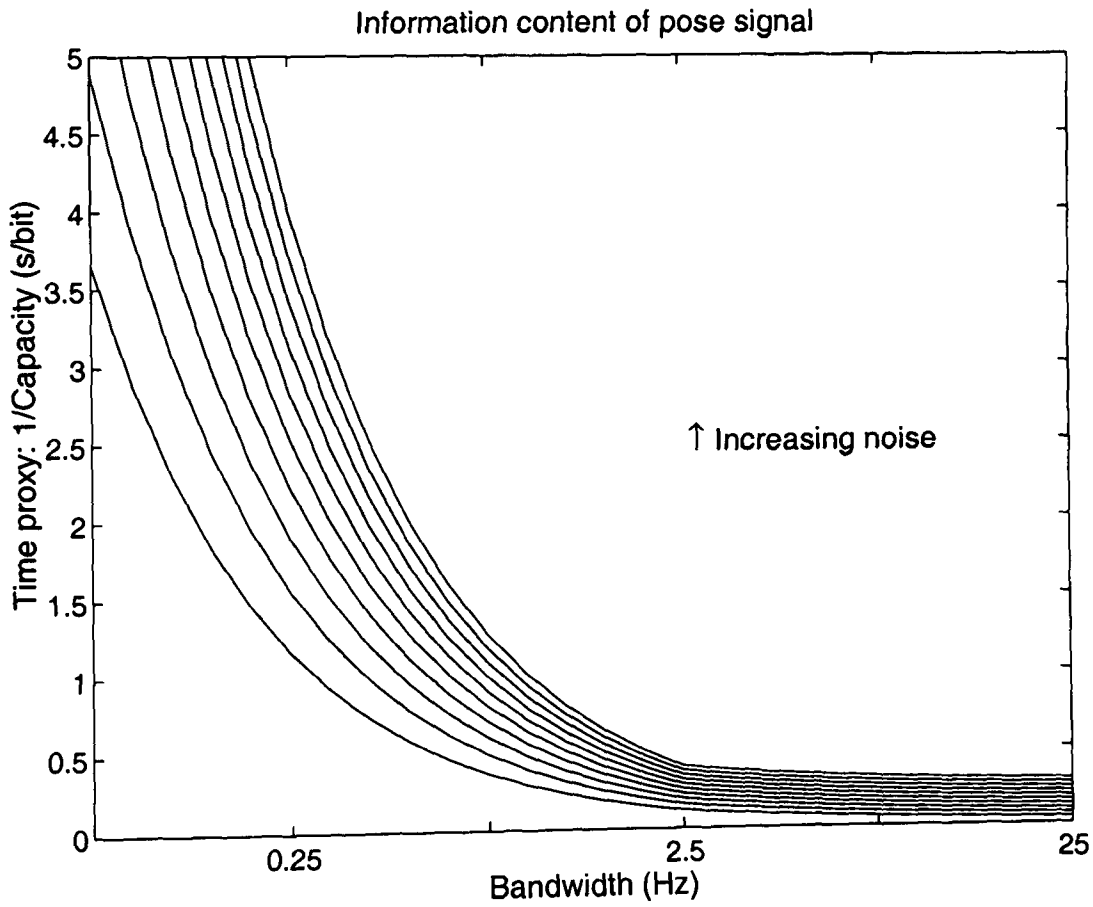


Figure 6.4: (a) In the Bayesian scheme the controller is a combination of a state estimator and gain. (b) As proxy for the time to complete a task, we invert the channel capacity K (the rate of information flow as defined by Shannon's theorem). At any value of the noise $1/K$ decreases with increasing bandwidth.

noise by the equation

$$B = \frac{1}{2\pi\tau}\tan^{-1}(\omega_c\tau)$$

where ω_c is the cutoff frequency of the sampled white noise (25Hz) and B is the brick wall bandwidth (Brown and Hwang [127]).

Hence the signal and noise are respectively

$$S = \begin{cases} aB & B \leq f_S \\ af_S & B > f_S \end{cases}$$
$$N = \begin{cases} bB & B \leq f_N \\ bf_N & B > f_N \end{cases},$$

and the signal to noise ratio is

$$S/N = \begin{cases} a/b & B \leq f_S \\ af_S/bB & f_S < B \leq f_N \\ af_S/bf_N & B > f_N \end{cases}.$$

(Note that a and b are constants of proportionality and only used in the ratio a/b, which is the signal to noise ratio below f_S.)

Figure 6.4(b) shows curves of $1/K$ versus bandwidth B for a number of different signal to noise ratios. If the inverse information rate is taken as a proxy for time to complete a task then the time is expected to drop sharply up to the f_S but then continue to fall though at a much reduced rate.

Summary of Hypothesis 1

The operator acts as an optimal estimator using an internal state to represent the teleoperation process. The time to completion for the operator should be the inverse of the Gaussian channel capacity into the operator. Under such assumptions, operator performance is maximized when no information is blocked: in other words, it is best *not* to apply a filter to the input stream of pose measurements, no matter what the noise. *The operator prefers raw measurements.*

6.2.2 The operator as an adaptive gain controller

The second postulated operator model uses an adaptive gain controller, with the gain chosen to minimize the variance between desired and actual output. Although not concerned

directly with control, Harris and Wolpert [64] used a minimum variance criterion to synthesize trajectories for arm and eye movements during goal-directed reaching and saccadic redirections of gaze, respectively, and found them to correspond well with those observed in human subjects.

Because the controller does not now contain an inscrutable internal state, it is possible to perform a more detailed analysis. The plant — the coupled chain of operator-master-slave — is modelled as a critically damped second-order system

$$D(s) = \frac{\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2}$$

with a natural frequency of $f_0 = \omega_0/2\pi = 2.5$ Hz, $\zeta = 1$, and unity d.c. gain. The neural signal to the plant from the controller is corrupted by noise, which for simplicity we take to be a disturbance with fixed variance σ_{m}^2. (Harris and Wolpert [64] assumed a variance proportional to the strength of the neural signal.) The plant pose output $p(t)$ is sensed and corrupted with noise with variance σ_{p}^2 before being passed through a filter with first-order roll-off

$$F(s) = \frac{1}{1 + s\tau} \ .$$

Setting the controller gain to k, and setting the input demand to zero, the Laplace transform of the output is

$$\begin{aligned} P(s) &= \frac{D(s)}{1 + kF(s)D(s)}\mathcal{L}(\sigma_{\mathrm{m}}) - \frac{kF(s)D(s)}{1 + kF(s)D(s)}\mathcal{L}(\sigma_{\mathrm{p}}) \\ &= H_m(s)\mathcal{L}(\sigma_{\mathrm{m}}) - H_p(s)\mathcal{L}(\sigma_{\mathrm{p}}) \ . \end{aligned}$$

Given that the output should be zero, the variance of the output is just $E[p(t)^2]$ which, denoting the autocorrelation of $p(t)$ as R_{pp} and the power spectral density as S_{pp}, can be re-expressed as

$$\min_k E[p(t)^2] = \min_k R_{pp}(0) = \min_k \frac{1}{2\pi}\int_{-\infty}^{\infty} S_{pp}(\omega)d\omega \ .$$

For a module with transfer function $H(\omega)$, the power spectral densities of input and output are related by

$$S_{\mathrm{out}} = H(\omega)H^*(\omega)S_{\mathrm{in}} \ ,$$

and noting that the noise processes are white, the variance is found as the sum of residues at stable poles:

$$\min_k E[p(t)^2] = \min_k \sum_{\substack{\text{stable} \\ \text{poles}}} \left[\; H_m(s)H_m(-s)\,\sigma_{\mathrm{m}}^2 \;+\; H_p(s)H_p(-s)\,\sigma_{\mathrm{p}}^2 \; \right] .$$

The minimum cannot be found analytically, and so is found by numerical search between k values of zero and that at which the system becomes marginally stable.

For the first operator model we were able to use the inverse information rate as the cost of control and a measure of the time taken for an operator to complete a task. This is of course directly related, via the Cramér-Rao analysis, to the variance of the signal. Thus for the second operator model a commensurate cost of control is the variance itself. In Figure 6.5(b) we plot $\min_k E[p^2]$ as a function of increasing bandwidth and different noise values.

Summary of Hypothesis 2

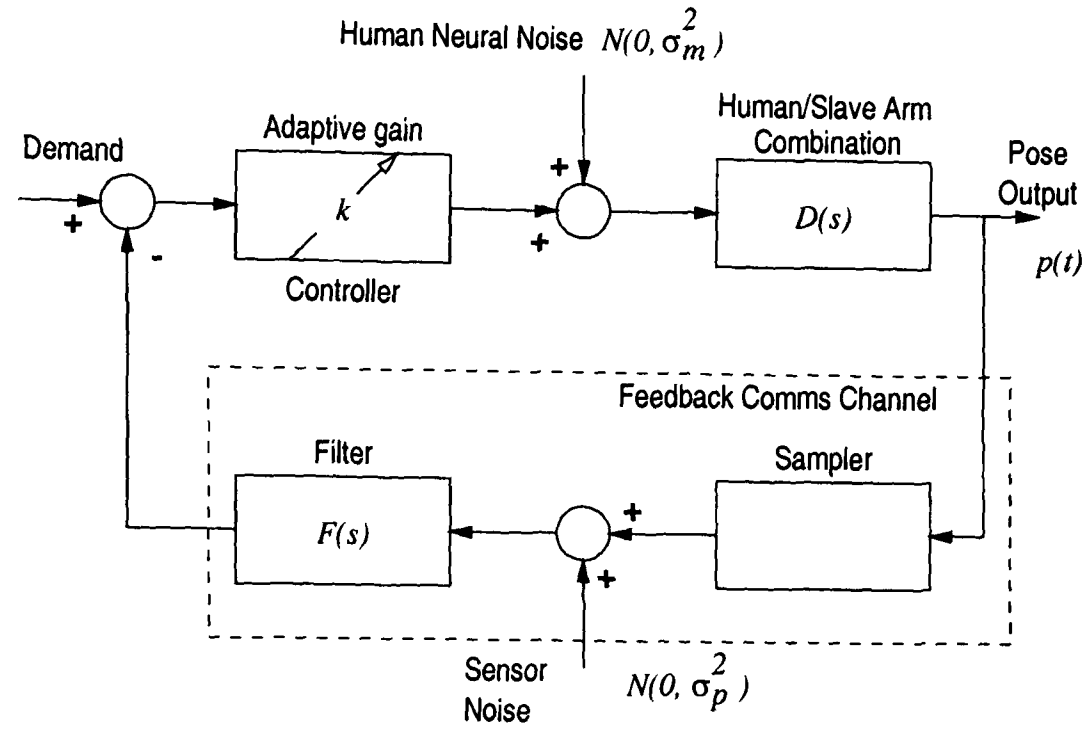
The operator holds no internal state describing the pose adjustment process and a simple optimal controller is used. The operator (internally) adjusts an adaptive gain to minimize variance. We assume that the adaptive gain is determined rapidly at the start of the task and is subsequently held constant.

In low noise the operator's performance is expected to fall then level out as the bandwidth is increased. In high noise, the performance decreases as the bandwidth increases, and in mid noise and above a minimum is apparent at around the interaction frequency f_S.

The significant difference with the first hypothesis is that in high noise the operator will perform better when the pose is filtered.

6.3 Experimental design

Experiments were designed to distinguish between the two hypothesized models of operator control by measuring the times taken to complete an insertion task. The task starts with ballistic movements in free space of a peg attached to the slave's end effector, continues with alignment with a target hole, and concludes with insertion into the hole, and is executed under various values of the noise added to the pose data and bandwidth of the filter



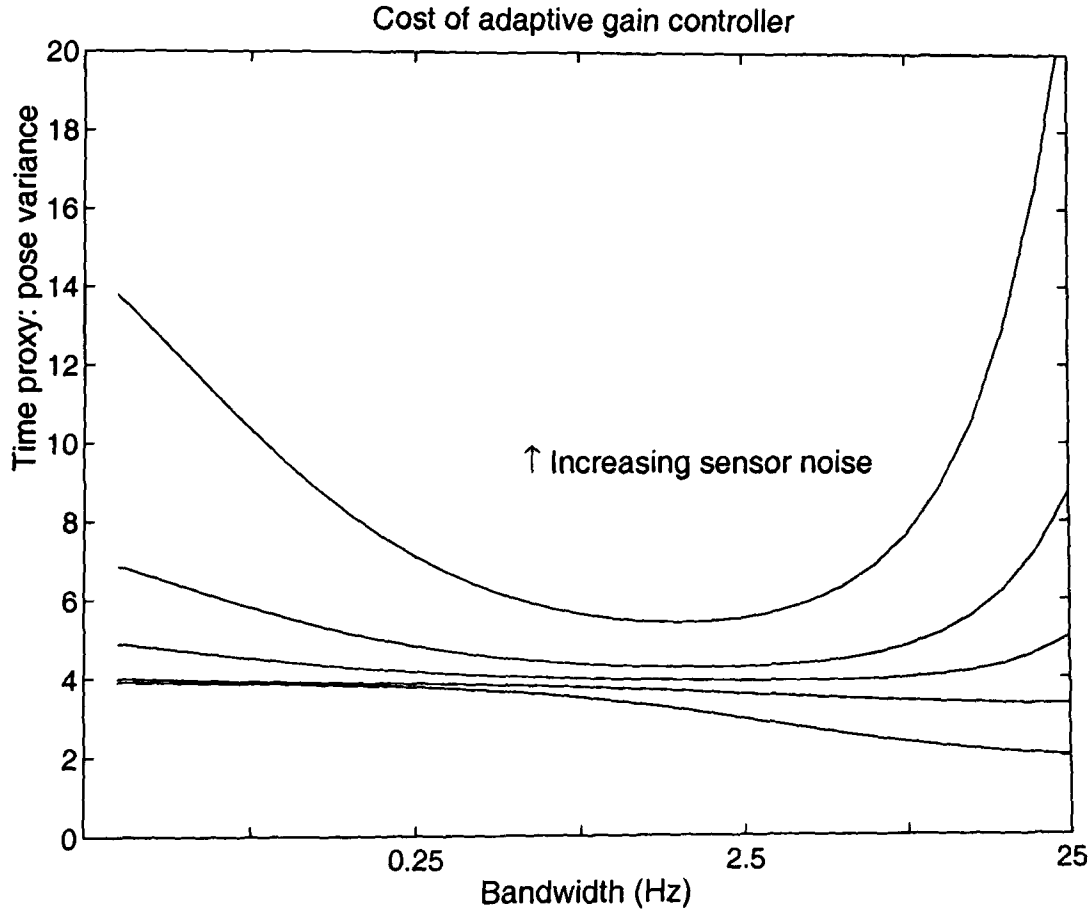


Figure 6.5: (a) The controller is an adaptive gain. (b) Operator performance determined from the variance in the pose plotted as a function of bandwidth and for several values of sensor noise.

used before displaying. Once aligned and in contact, insertion itself is straightforward, and so the task is dominated by non-contact motions.

6.3.1 Equipment

In this chapter we wish to add *controlled* amounts of noise to the pose. The test object, a machined peg, is therefore rigidly attached to the slave manipulator and has its pose determined directly via the slave robot's joint angles and its forward kinematics. For the purpose of these experiments, the position and orientation obtained in this way are assumed error-free. A controlled level of noise is added, and the corrupted signal passed through a filter before being used to set the positions of the objects within the virtual viewing system.

The peg is designed to key precisely in one position into the hole object, and so the position and orientation of the hole can be similarly determined via the slave's kinematics. This is done once, and the object then held fixed throughout the experiments.

6.3.2 Filter Synthesis

Before using the pose to display, noise is added to it and the result is passed through a low pass filter. Noise was drawn from a Gaussian distribution with zero mean and variance σ_p^2 and added to the components of the pose

$$p' = p + \mathcal{N}(0, \sigma_p^2) \ .$$

Use of the robot kinematics would allow pose to be recovered at 330 Hz but, to mimic recovery by a computer vision system running at frame rate, sampling was performed only at 25 Hz. This noise source is therefore bandlimited at $f_c = \omega_c/2\pi = 12.5$ Hz.

The corrupted pose data p'_t are passed through a single-pole low pass filter with transfer function

$$\frac{V(s)}{P'(s)} = \frac{1}{1+\tau s}$$

to generate the viewable pose v_t. Filtering bandlimited white noise with the pole filter results in a system which has an equivalent brick wall noise bandwidth of

$$B = 1/(2\pi\tau)\tan^{-1}(2\pi f_c \tau)$$

where τ is the filter's time constant. The continuous transfer function is transformed into the z-domain using the Tustin transform giving

$$\frac{V(z^{-1})}{P'(z^{-1})} = \frac{1}{1 + (1 - z^{-1})/(\alpha(1 + z^{-1}))}$$

where $\alpha = T/(2\tau)$ and T is the sample period. Rearranging and taking the inverse transform yields the current temporal output for display as

$$v_t = \left(\frac{1-\alpha}{1+\alpha}\right) v_{t-1} + \left(\frac{\alpha}{1+\alpha}\right) (p'_t + p'_{t-1})$$

in terms of the previous temporal output, and the current and previous inputs p'_t and p'_{t-1}.

6.3.3 Experimental Procedure

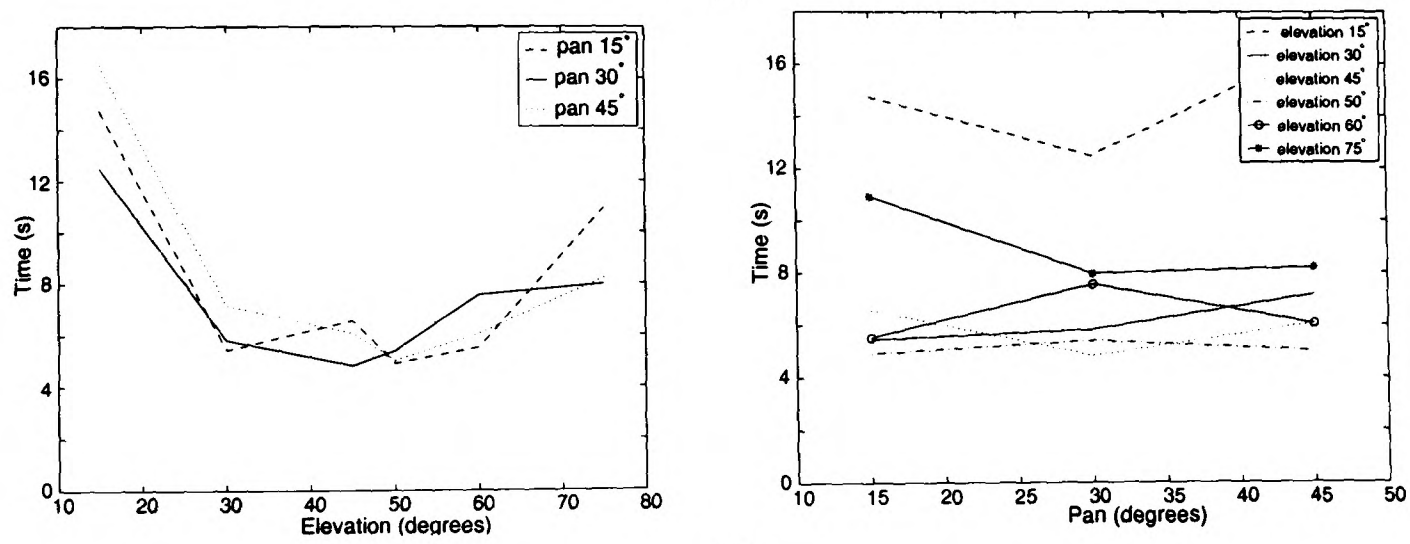
The data collected from the experiments were the time to completion of a series of peg in hole tasks.

Nine graduate students were used as test subjects. To mitigate the effects of learning on the experimental results, a training session was held for each subject two days prior to the actual experiments. Each was taught to use the force-reflecting joystick for input movements, until he could perform the experimental manipulation tasks under zero noise and no filtering with ease and with near constant time to completion.

In regular use the operator might wish to move the virtual viewpoint during an insertion task. However, to eliminate this variable from the experiments, a separate independent trial was made which measured time to completion under no noise but using different viewpoints. The optimum fixed view is that shown in the middle of Figure 6.6(b).

The actual experimental task consisted of the operator using the master input device to move the peg from a starting position some 300mm from the hole and inserting it. Each subject completed nine runs which used all the combinations of three different noise values $\sigma = \{0.5, 2, 5\}$mm and three different bandwidth values $B = \{0.25, 2.5, 25\}$Hz. To counteract the effects of fatigue and learning the experimental order was different for each subject. The experiment started when the master input device was engaged and was completed when the peg was seated in the hole. After each run was finished the peg was reset to a similar but not identical location.

(a)



(b)

Figure 6.6: Results from the trial to fix the optimum view. (a) Views of the rendered peg and hole using Geomview. The middle view is that found as the optimum one determined from the task completion times shown in (b) as a function of elevation angle and pan angle.

6.4 Results and Analysis

The results from the 81 separate runs are summarized in Figure 6.7. The times, averaged over the nine participants, to complete each insertion task, along with the standard deviations, are shown as functions of bandwidth for the low, moderate and high noise. The connecting lines are merely to guide the eye. The salient features are:

- Similar completion times across data sets, independent of noise, at both 0.25 Hz (22 seconds) and 2.5 Hz (10 seconds).
- The consistent reduction of completion times between 0.25Hz and 2.5Hz.
- As the bandwidth is increased, the negligible change in completion times at low and mid noise contrasts with the sharp increase in completion time at high noise.

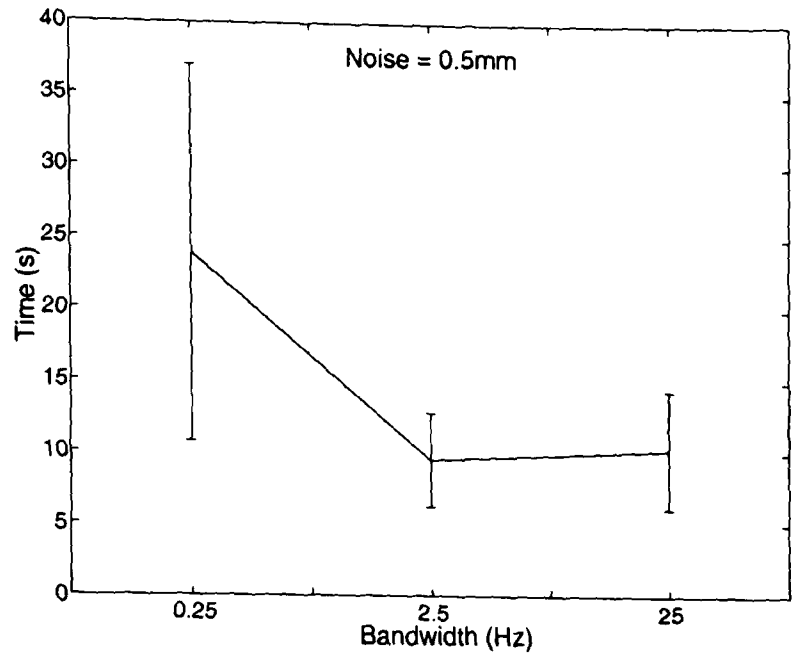
The experimental results were analyzed using ANOVA to test for the significance of these changes in operator performance as the experimental conditions or treatments are altered. ANOVA (analysis of variance) is a test which studies the effects of many factors, here noise and bandwidth, separately (their main effect) and together (their interactive effect).

1. **The Null Hypothesis:** Before proceeding with a more detailed analysis of the experimental data the null hypothesis must be examined. The Null hypothesis in ANOVA tests whether the means are drawn from the same population. Its purpose is twofold. The first is to examine whether the changes in times to completion as the two parameters B and N are varied could be attributed to chance alone; and the second is to ask if the parameters interact in anyway.

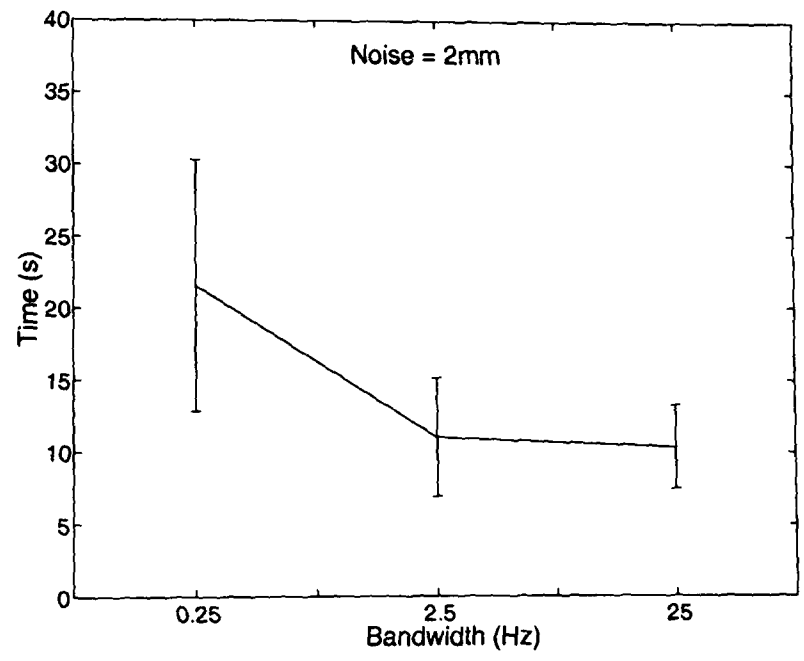
   For the full two way ANOVA test the p value for noise is $p = 0.0226$ ($p < 0.05$ indicates a significant result), bandwidth $p = 0.000$ and the interaction $p = 0.0393$. We therefore reject the null hypothesis.

2. **Low Bandwidth Hypothesis:** For filters with bandwidth of 0.25Hz and 2.5Hz, the means are drawn from the same filter populations.

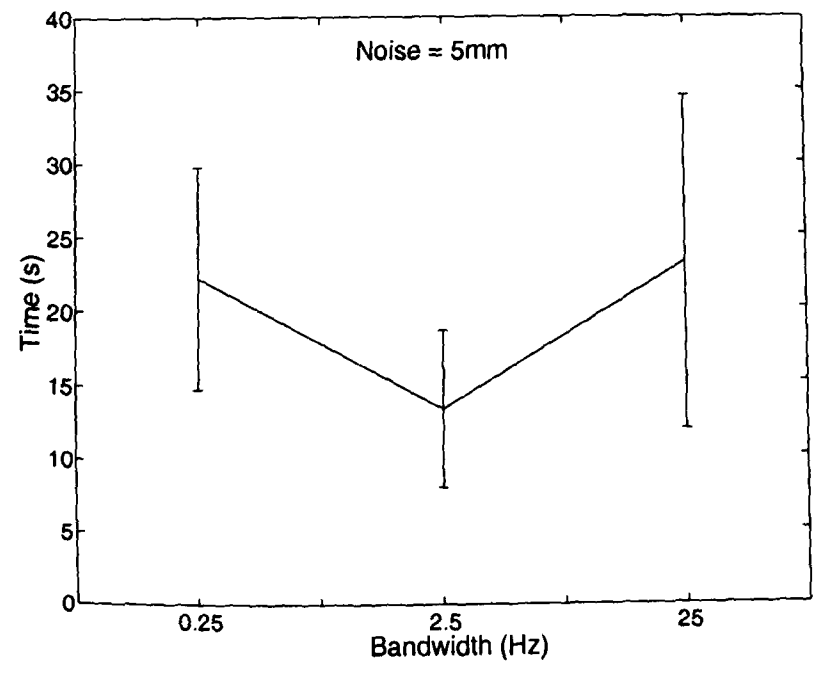
   For both operator models at low bandwidth the predominant effects are due to changes in bandwidth and not in noise. We test for this trend in the results by performing ANOVA on the sets of experiments noise $\sigma = \{0.5, 2, 5\}$ mm and bandwidth



(a)



(b)



(c)

Figure 6.7: The mean times to complete the insertion task as a function of filter bandwidth for (a) low noise, (b) moderate noise, and (c) high noise.

$B = \{0.25, 2.5\}$ Hz. The p values obtained for noise $p = 0.8628$, bandwidth $p = 0.000$ and the interaction $p = 0.5696$. This indicates that the response due to the change in noise has little effect, however a change in bandwidth would produce a significant change, but doesn't interact with the change due to noise. Hence once the filter is chosen (for low bandwidth values) noise has little effect on the operators performance. We therefore accept the hypothesis.

3. **High Bandwidth Hypothesis:** For filters with bandwidth of 2.5 Hz and 25 Hz, the means are drawn from the same population.

   As we have said in the introduction to this section: the key difference between the two operator models should be the time to completion at high noise values for bandwidths above and around the natural frequency of the operators arm dynamics (2.5Hz). Figure 6.7 shows that the times to completion for large noise (5mm) and large bandwidth (25Hz) is considerably more than other values for filters of bandwidth 2.5Hz and for lower noise values with the same filter. We thus run ANOVA over this data set: $\sigma = \{0.5, 2, 5\}$ mm and bandwidth $B = \{2.5, 25\}$ Hz. The p values obtained were for noise $p = 0.0002$, bandwidth $p = 0.0437$ and interaction $p = 0.0200$. We therefore reject the hypothesis.

6.5 Conclusions : Noise and Bandwidth Experiments

The human operator of a remote slave robot who views a graphical rendition of moving objects whose pose has been previously recovered by a computer vision system, sees a sampled, noisy, and perhaps temporally filtered version of reality. One aim of this chapter has been to understand how to treat those noisy pose measurements before using them to generate the graphical display.

The issue of noisy pose is not one that has been raised by work on predictive displays. There, interactions between objects are modelled in the scene database, and resulting motions and poses are assumed absolute. In our case, where the feedback is prompt enough to be of immediate use, our object models really form part of the *sensing* process, allowing us to make measurements on the actual work space.

Empirically we find that under large noise conditions, optimum operator performance is

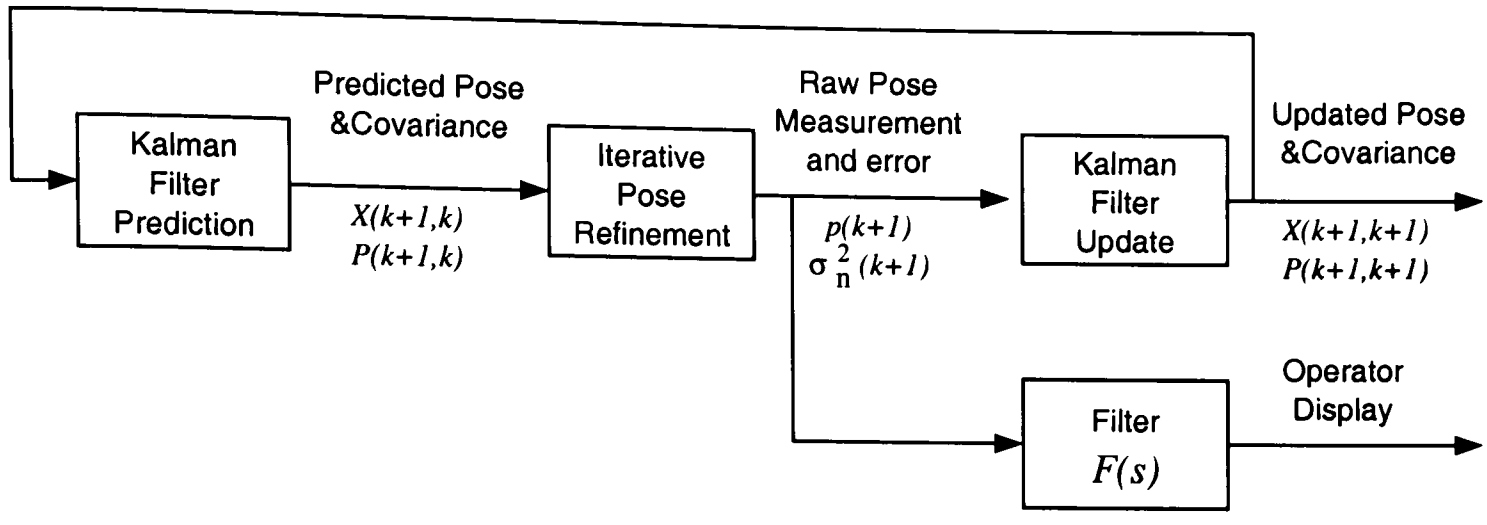


Figure 6.8: Filtering for viewing should be independent of that for pose tracking, if the pose tracker cannot achieve a robustness bandwidth of 2.5 Hz. If the noise is high $F(s)$ should have a bandwidth of some 2.5 Hz, and if low $F(s)$ can be omitted.

achieved when pose measurements are filtered to match the operator's bandwidth of interaction with the environment, here some 2.5 Hz. A conclusion of this chapter is that when designing a pose recovery and display system, filtering for display should be considered separately from filtering for pose tracking *if* the bandwidth of the pose tracker is much lower than 2.5 Hz, as summarized in Figure 6.8. In the previous chapters on the design of the pose recovery system it has been a fundamental aim to increase the bandwidth to 2.5 Hz thus matching the pose recovery filter to the operators input and output characterisics. This, perhaps surprisingly, is in contrast to the design issues of force feedback rather than visual feedback for manipulation. Here the force a person can feel is much higher than the frequency range over which they can act via the human motor system.

Our experiments were designed not just to measure operator performance, but also to allow us, at least potentially, to falsify one or both of two hypothesized models of operator performance.

The first model supposed that the operator runs an optimal stochastic controller which includes updating an internal state representing relevant aspects of the world. Most work on sensor integration with engineering takes the line that the more sensor data the merrier, under the presumption that all data can be integrated to improve performance as a simple consequence of Bayes' theorem. In our case that would mean that the operator would perform better without filtering no matter what the value of the noise. However, our results suggest that the human operator is not a Bayesian engine.

There is of course other evidence in other tasks. We have already noted that combining vision with another sensing modality can produce visually dominated perceptions, and have given an example where the subdominant haptic mode is coerced into giving perceptions which agree with vision. In other tasks there are quite unexpected effects (Calvert *et al* [128]). For example, subjects who hear one word dubbed over the video of a speaker's lips saying another report perceiving neither the visual nor the auditory information correctly (McGurk and MacDonald [129]). Perhaps vision again attempts to dominate, but our ability to lip read is generally too poor to see the correct utterance.

We are unable to falsify the second hypothesis. Our experimental measurements are consistent with the operator running a minimum variance strategy, here modelled in it simplest form as an adaptive gain controller. This model correctly predicts the increased cost of control when the noise is large and the filter's bandwidth is increased well beyond the natural frequency of master-slave apparatus.

Notice that, at high noise and low bandwidth this model also predicts a rise in time to completion, but not so at low noise. It seems likely that other mechanisms explain the poor performance below 2.5 Hz. Wolpert *et al* [130] suggested that the cerebellum contains an internal model of the human's motor apparatus. For motion planning, this internal model is inverted to produce the necessary motor commands, and thus the consequences of actions are predictable *before* feedback from the senses. For smooth, predictive and planned operator motion, they suggest that feedback from the senses must come at the expected predefined moment after the event has taken place. Imposing larger time delays or phase lags seriously degrades human performance.

Harris and Wolpert (1998) [64] used a minimum variance criterion to generate trajectories for arm motion during reaching tasks, and we re-iterate that the performance times measured here are dominated by free-space motion. (Indeed the form of $D(s)$ used assumes this.) However, in our work minimum variance is achieved using a simple adaptive gain controller *without* an internal state, because our system has feedback, whereas that of Harris and Wolpert runs open-loop.

6.6 Introduction: Bias Experiments

This section details the second set of operator experiments performed using the system developed for this thesis. The purpose of these experiments is to address the concern expressed in the introductory chapter of this thesis on whether operators can tolerate visual positional biases when performing telemanipulation. Biases may be introduced in teleoperation systems which used predictive displays, where a model of the remote site and the operator's current slave demands are used to infer the workcell locations. This is in constrast to the system developed in this thesis where a sensor based visual pose recovery system is used to infer the current object locations.

When using predictive displays errors may be present in the forward kinematics of the robot which propagate to errors in the Euclidean location of the end-effector (eg. Das [77]). In addition, sensors used to obtain the relative Euclidean transformations of the manipulated objects may be mis-calibrated also resulting in biases in the pose. Kim [51] for example, used a computer vision system to localise stationary objects. A third possible source of bias, just as potent though perhaps easier to correct, is the lack of fidelity between the virtual and real world object. For the purpose of this chapter these errors are lumped together and modelled as time-invariant biases.

The experimental procedure detailed in the previous section was followed and the experimental conditions examined were the levels of bias in both the translational and rotational components of the peg pose.

6.6.1 Effects of Variation in Bias

Translational Bias

The times to complete the peg insertion task were measured under values of translational bias added between the pose recovery stage and the display of the information to the operator. The display was updated at a fixed frame rate of 25 Hz, substantially higher than the frequency of 5.6 Hz given in [11] as the minimum needed for fatigue-free operation.

In the first set of experiments, translational biases from the discrete set of values of 0mm, 5mm and 10mm were added in each of the X, Y, Z directions of a coordinate frame centred on the hole. The X and Y axes lay along the sides of the plate in Figure 6.6, and

the Z axis pointed upwards. Points were chosen at random from the $3 \times 3 \times 3$ lattice, to mitigate the effects of fatigue and familiarity. The raw mean completion times are plotted for the various bias values in Figure 6.9.

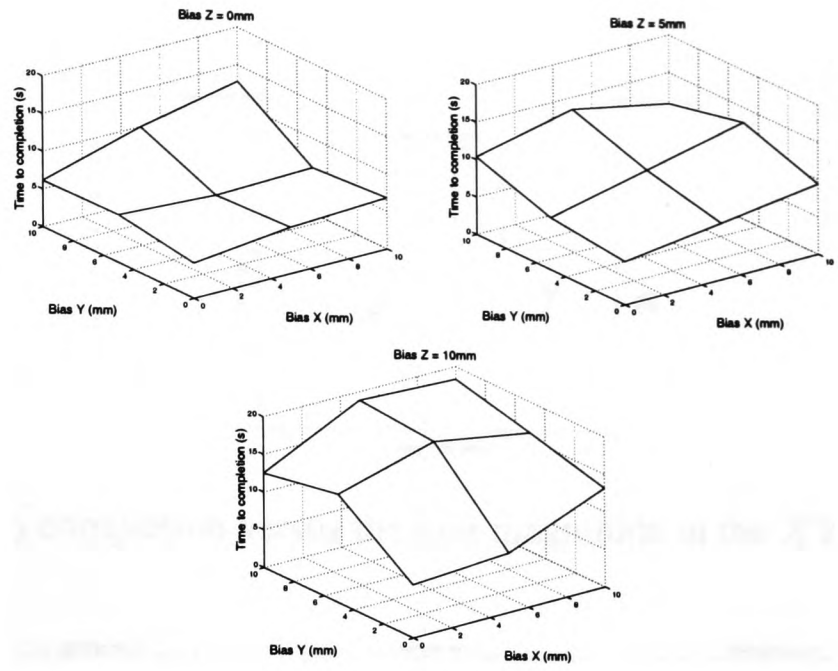


Figure 6.9: Time to completion versus different biases in the horizontal X, Y-plane for the three values of Z bias.

The graphs are nearly symmetric about the line $X = Y$, and in Figure 6.10 we show the time to completion as a function of the magnitude of the bias in X and Y, plotted separately for the 3 values of Z bias.

6.6.2 Variation in rotational bias

The experiments involving orientational bias followed the same methodology. Bias was applied with randomly chosen discretized values about three axes, one aligned with the peg's major axis, and two orthogonal to it. In all cases the axis of rotation went through the tip of the peg so that adding rotation did not displace the tip. The angles rotated around the major axis were $0°$, $5°$ and $10°$. As in the last section, symmetry in completion times was found for rotations around the remaining two axes, so these parameters were combined into a magnitude parameter which was randomly partitioned to the two rotations. These magnitude values were $0°$, $2°$, $4°$, $6°$, $8°$, $10°$, $12°$, $14°$, respectively. The results are shown in Figure 6.11. We note that there is little dependence on changes in bias about the major axis.

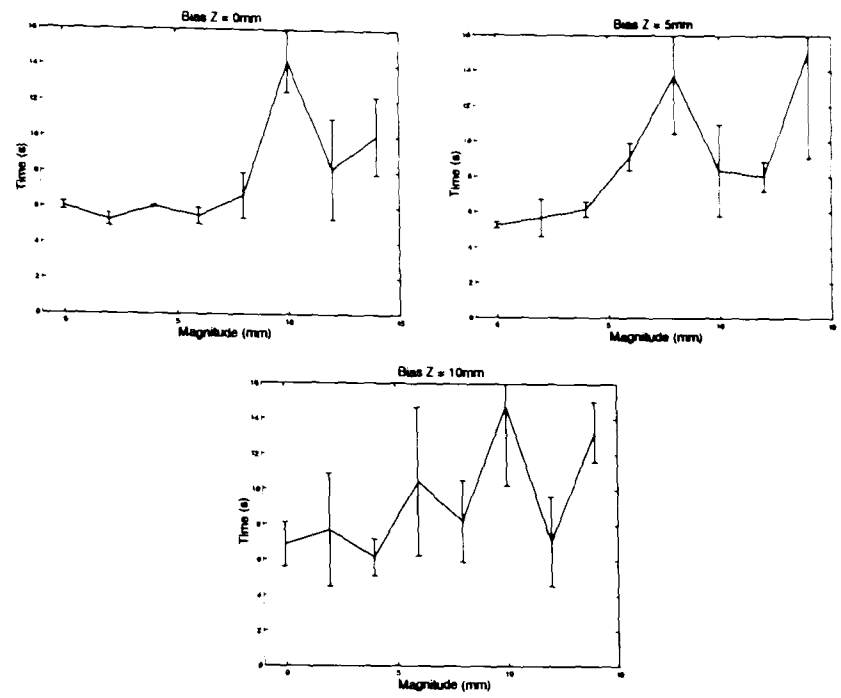


Figure 6.10: Time to completion versus the bias magnitude in the XY-plane for Z bias of 0, 5 and 10 mm.

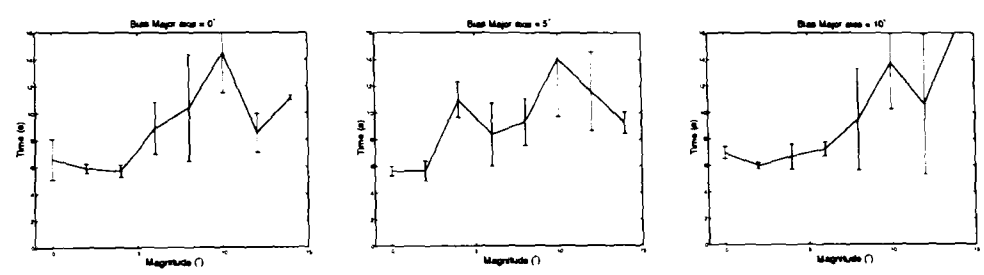


Figure 6.11: Time to complete versus different orientation bias.

6.7 Conclusions: Bias Experiments

This section shows that small biases are easily tolerated in teleoperation but above a certain level the times to completion and operator fatigue increase rapidly. This finding is consistent with those of Klatzy *et al.* [55] who noted that even for non-routine manipulation tasks the importance of attentive foveal visual feedback is much reduced in spatial configurations where force feedback gives reliable spatial information. The peg in hole task requires little visual feedback after contact is made, rather unlike threading a needle. This ready partitioning of the two contact strategies results in visual biases being counteracted by force feedback — the operator knows from experience that a two phased approach is needed for positioning the peg in the hole, an initial visual positioning and then a more refined force positioning strategy. With large biases contact is made between peg and *plate* rather than between peg and hole and the operator is effectively blind both haptically and visually.

The need for, or rather the ability of, the operator to adopt a two phase approach to contact is likely due to intrinsic noise in human motor actuation. Harris and Wolpert [64] proposed that the variance of the motor noise is proportional to the strength of the control signal and used this single physiological assumption to predict the empirical Fitt's law. The consequence of the noise is that we expect our actions to be a little off course and automatically compensate with subsequent positioning strategies. It is as though the small biases are indistinguishable from the already present noise in the muscular-actuation signal.

The results of this section show that sensor based systems rather than fixed model based systems as used in predictive displays were a model of the workcell is driven from the operators control input to infer the current object locations is more optimal for teleoperation.

7

Conclusions

This concluding chapter summarises the thesis, details its conclusions and outlines directions for future work.

7.1 Summary

The main aim of the thesis was to

- explore, improve and in some way optimise the extraction of information from fixed cameras to facilitate teleoperation.

In the introduction this objective, after a brief review of the existing literature, was sub-divided into two rather narrower focused sub-objectives:

- to develop a visual pose[1] recovery system which can recover the pose of workcell objects from the video feeds of the remote workcell.

- explore the visual informational needs of an operator performing a manipulation task, to better the design of a pose recovery system.

[1] position and orientation

The chapter-by-chapter conclusions chart the progress made in the thesis to achieve a first iteration of these objectives.

- In chapter 3 it was discussed how a virtual viewing system could be added to the Oxford force-reflection teleoperation system, respecting the overall architectural and system issues. It continued by discussing the notion of proprioceptive alignment, particularly of visual and tactile frames centered on the head and hand. No method of visual presentation appears ideally suited, and we chose a compromise that provided reasonable alignment but without encumbering the operator.

- Chapter 4 detailed a method of camera calibration using a robot manipulator arm. It was not assumed that the robot could generate perfectly accurate world positions and a possible observational model was proposed. This observational model was incorporated in a camera calibration algorithm to obtain the MLE of the camera parameters. Experiments were performed to support the usefulness of this extra positional information to obtain accurate camera calibration.

- Chapter 5 made a first attempt at developing a visual object pose tracker building on the method of model-based tracking first introduced by Harris but extended to multiple cameras and afforced with robust methods. It performed satisfactorily for object motions of up to 1Hz bandwidth.

- Chapter 6 detailed two sets of experiments conducted on operators using the equipment developed in this thesis. Experiment i. investigated alternative models of the operator's performance during a visually-guided insertion task and develop an argument to test whether the performance is consistent with the operator's visuo-motor control loop maintaining a state model of the process as in a Kalman filter. We concluded that the results did not support this model. A further model was thus presented which modeled the operator as an adaptive gain controller. The results did support this model.

  This finding in the context of visual pose recovery reveals that unlike force where, Goertz remarked "the frequency range that a person can feel is much higher than the frequency range of the human motor system", we find the frequency range a needed

by person a performing manipulation by viewing is the same as that of the human motor system. This symmetry simplifies the design of the tracker.

Experiment ii. investigated the operators performance under increasing static bias which may be present if a simulation of the remote site had been used to update the pose of the workcell objects. We concluded that operators can only tolerate biases of

becomes virtually impossible. This is the motivation some 5mm before teleop

is not based on geometrical priors and uses sensors to for using an approach wl

measure the object's pos

7.2 Future Work

Although each chapter has a large section dedicated to future work centered on the work conducted in that particular chapter, important research directions which has arisen due to the work as a whole will now be outlined:

- Chapter 4 detailed a method of autonomous camera calibration using the slave-manipulator arm, we feel a more thorough exposition of some of the real-time issues would greatly serve the computer vision community. The key areas of research, in our opinion, are i) to develop a recursive implementation of our proposed batch method, perhaps contrasting the results from an Extended Kalman Filter, which is known to be sensitive to initial conditions and the Unscented Filter proposed by Jullier and Uhlmann [131]. We view the unscented filter as a key component of computer vision systems utilising recursion; it is not plagued by the effects of nonlinearities as with the EKF, and it is computationally tractable, unlike particle filters. Another key area of research is ii) a functional definition of camera calibration accuracy perhaps drawing on the design of experiments literature [132]. A functional definition would greatly facilitate the derivation of a camera calibration path planning routine and the active vision community in the whole.

- Chapter 5 laid a framework for robust pose recovery to track manipulated objects of bandwidth of 2.5Hz within an extended Kalman filter framework. An EKF is an attractive framework because of its ability to filter pose at frame rate. However this

approach is only viable, because of the extended Kalman filters inability to track multiple state hypotheses, when there is sufficient prior information and image information to disambiguate each tracked object every frame. We argue there is, after all only three object point correspondences are required to recover pose. We have laid out a framework to disambiguate this data, or increase robustness, by using feature saliency.

An issue which we would have liked to explore is that of choosing features not only to achieve robustness but also to achieve accuracy. In a multi-camera system it makes sense to distribute control points equally amongst the images rather than using the same control point in the same image over and over again but why is this so? Perhaps a control point placement algorithm could be devised to automatically generate and place control points on the objects to increase accuracy. A good starting point would be the work of Davison [133]. Davison constructed an active vision robotic simultaneous localisation and mapping (SLAM) system which automatically selected which feature in the world to locate, measure and then incorporate in the overall map to improve positional information. It must however be stated that mapping systems are only currently concerned with locating landmark features, serially and which are robustly matched in the image.

- Chapter 6 detailed experiments conducted to measure the optimal bandwidth of a filter for telemanipulation where optimality was an inverse measure of time to completion for a manipulation task. But telemanipulation is just one task an operator may perform using a teleoperator. An extension to our studies could include experiments to obtain the optimal filter parameters for a variety of other teleoperation tasks. For example, when idle[2], the operator does not wish to see objects jitter on the screen, and the required filter bandwidth would be a lot lower than that needed for optimal manipulation. Although the measurement of the optimal parameters for each task might be relatively straight forward, designing a system to deduce what task the operator wishes to perform, ie. the inference or labelling of operator intent, might be a little more challenging. Even more so when one consideres that manipulation may

[2]maybe the word task is a little too strong

be broken down into four subtasks or phases (Klatzky *et al.* [55]) with the weighting of information from vision and force and hence the characteristics of the visual filter being different for each phase.

7.3 Outlook

Teleoperation systems allow operators to perform non-routine, delicate tasks in hazardous environments which robots, due to their limited intelligence and adaptability, would not be able to perform autonomously at the present time. Teleoperation systems also offer an intimation into the structure of the basic manipulation modules present in primate brains, which we as humans routinely use without a second thought but as yet have eluded a computational model.

This thesis has laid down a framework, principally to improve teleoperation systems but additionally to improve our understanding of the informational needs of the operator. This is of benefit to fulfill the immediate demand for better teleoperators and also maybe to fulfill the long term demand of autonomous systems, completely removing the need for operators. As we have alluded to in the introduction of this thesis, whether this is a good thing remains to be seen.

A

Coordinate Frames and Homogeneous Transformations

The following appendix puts forward the notation and conventions used in this thesis for coordinate frames and their respective interframe transformations/projections.

A.1 Homogeneous Transformation Notation

A.1.1 Cartesian Coordinate Frames

For visual teleoperation systems the coordinate frames of importance are:

1. Slave manipulator base frame.
2. Master manipulator base frame.
3. Camera frames.
4. Object frames.

In this thesis the adopted convention in expressing the transformation through these coordinate frames (though not the notation) is consistent with Craig [72]. The term $\mathtt{D}_{\mathtt{i}\rightarrow\mathtt{j}}$ describes the transformation of a body-fixed frame $O_i x_i y_i z_i$ to a new location $O_j x_j y_j z_j$.

Geometric entities are expressed relative to a coordinate frame, eg. the point whose coordinates are in the ith coordinate frame is denoted $\mathbf{p}^i$.

A.1.2 Homogeneous Euclidean Transformations

In general the 4x4 homogeneous matrix allows coordinate transformations to take place between homogeneous points defined by projective coordinate frames, however all our coordinates frames are Cartesian and hence the point 4x4 homogeneous transformation matrix takes on the special form of

$$\mathtt{D}_{\mathtt{i}\to\mathtt{j}} = \begin{bmatrix} \mathtt{R}_{\mathtt{i}\to\mathtt{j}} & \mathtt{t}_{\mathtt{i}\to\mathtt{j}} \\ \mathtt{0}^{\mathtt{T}} & 1 \end{bmatrix}$$

Coordinate Frame Transformations

Coordinate frames can be attached to any rigid body to denote the pose of the object with respect to a coordinate frame. A chain of displacements can be concatenated together to form a kinematic chain. Eg. the frame attached to a manipulator's end-effector expressed in the manipulator's base frame by

$$\mathtt{D}_{\mathtt{B}\to\mathtt{E}} = \mathtt{D}_{\mathtt{B}\to\mathtt{0}}\mathtt{D}_{\mathtt{0}\to\mathtt{1}} \cdots \mathtt{D}_{\mathtt{6}\to\mathtt{E}}$$

Homogeneous Point Transformations

We defined a point $\mathbf{p}^i$ as the homogeneous point expressed in the "i"th coordinate frame. The point can be expressed in other coordinate frames by the application of the homogeneous transformation matrix $\mathtt{D}_{\mathtt{i}\to\mathtt{j}}$. The point in the "j"th coordinate frame is transformed into the point expressed in the "i"th coordinate frame by

$$\mathbf{p}^{\mathtt{i}} = \mathtt{D}_{\mathtt{i}\to\mathtt{j}}\mathbf{p}^{\mathtt{j}} \quad .$$

A valid interpretation of $\mathtt{D}_{\mathtt{i}\to\mathtt{j}}$ is that it "converts" the coordinates of a point known in the jth coordinate system into the ith coordinate system. As an aside the columns of the rotation component of $\mathtt{D}_{\mathtt{i}\to\mathtt{j}}$ are just the vector coordinates of the axes of the jth frame, namely $x^{\mathtt{j}}$, $y^{\mathtt{j}}$, $z^{\mathtt{j}}$ in the ith coordinate system.

A.1.3 Homogeneous Projections

We are less restrictive when it comes to classes of projective transformations we simply defined the 3x4 projective matrix operating on homogeneous points to be

$$\mathtt{P}_{\mathtt{i}\rightarrow\mathtt{j}} = \begin{bmatrix} P_{00} & P_{01} & P_{02} & P_{03} \\ P_{10} & P_{11} & P_{12} & P_{13} \\ P_{20} & P_{21} & P_{22} & P_{23} \end{bmatrix}$$

A.2 Camera Frames and Models

The ideal pin-hole camera model is a good representation of the geometry of a CCD camera. Central to the pin-hold model is that of the 3x4 projective transformation matrix P. The coordinate frames associated with cameras are shown in the next section, followed by the form and decomposition of the projective matrix P into projective camera parameters.

A.2.1 Camera Coordinate Frames

Cameras generally have three coordinate frames. Firstly the camera coordinate frame which is a 3D frame aligned with the camera's optic center and positioned as in Figure A.1. The second is the 2D image coordinate frame whose coordinates are given in pixels and is aligned relative to the camera coordinate frame as in Figure A.1. The third virtual image is just a notational convenience lying at unit focal length away from the optic center and again aligned with the camera coordinate frame.

The 3x4 homogeneous projection matrix (defined in Section A.1.3), which projects 3D homogeneous points to 2D homogeneous points, is generally defined from a world coordinate frame.

$$\mathtt{x}^{\text{image}} = \mathtt{P}_{\text{image}\rightarrow\text{world}}\mathtt{X}^{\text{world}} \quad \text{and} \quad \mathtt{x}^{\text{ideal image}} = \mathtt{P}_{\text{ideal image}\rightarrow\text{world}}\mathtt{X}^{\text{world}} \quad .$$

The intermediate 3D position expressed in the cameras coordinate is often used in any visibility stage of computer vision algorithms such as analysing object profile points. This Euclidean transformation forming the camera's extrinsic parameters.

$$\mathtt{X}^{\text{cm}} = \mathtt{D}_{\text{cm}\rightarrow\text{wo}}\mathtt{X}^{\text{wo}}$$

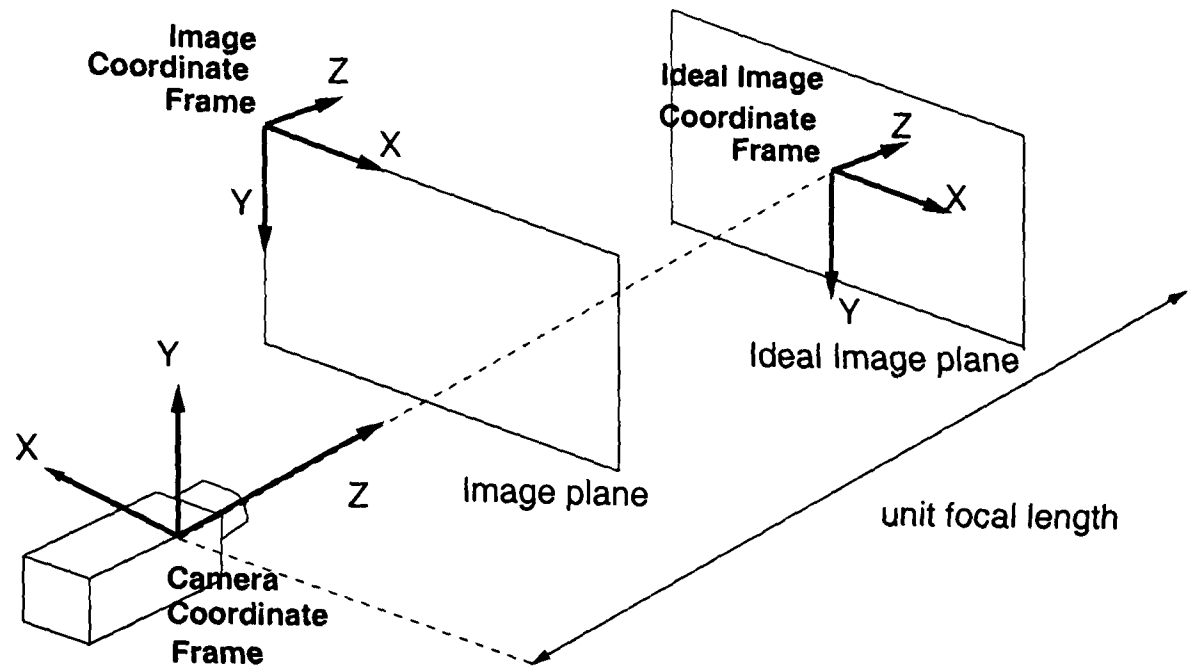


Figure A.1: Each camera generally has three coordinate frames. The camera, image and ideal image coordinate frame. Note the image and ideal image coordinate frames are only 2D the z axis is shown just for reference.

and

$$\mathtt{D}_{\mathrm{cm}\rightarrow\mathrm{wo}} = \begin{bmatrix} \mathtt{R}_{\mathrm{cm}\rightarrow\mathrm{wo}} & \mathtt{t}_{\mathrm{cm}\rightarrow\mathrm{wo}} \\ \mathtt{0}^{\top} & 1 \end{bmatrix}$$

it is often common to encode the rotation in its angle-axis equivalent form.

A.2.2 Camera Models

As the pin-hole camera model is a good representation of CCD cameras the world to image transform is conveniently encoded in the 3x4 projection matrix P. This projection matrix is often decomposed into the projective camera parameters for most estimation problems - a minimal set of parameters obviously not having any redundancies.

Projective Camera Parameters

The projection matrix for the world to image projection is composed of 10 camera parameters, 4 intrinsics refering to the focal lengths $(\mathrm{f}_x, \mathrm{f}_y)$ and image center $(\mathrm{u}_0, \mathrm{v}_0)$ and 6 extrinsics representing the displacement of the camera from a Euclidean reference frame (the translation $\mathtt{t}_{\mathrm{cm}\rightarrow\mathrm{wo}}$ and rotation which is encoded in its angle axis form $[\mathrm{aa}_x, \mathrm{aa}_y, \mathrm{aa}_z]$).

The *camera calibration matrix* K for our camera frame conventions is given by

$$\mathtt{K} = \begin{bmatrix} -\mathrm{f}_x & 0 & \mathrm{u}_0 \\ 0 & -\mathrm{f}_y & \mathrm{v}_0 \\ 0 & 0 & 1 \end{bmatrix} \quad ,$$

to express coordinates in the ideal image which is at unit focal length in the x and y directions and is centered in the middle of the image

$$\mathtt{K}^{\mathrm{i}} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

the projective camera parameter 10-vector is

$$\mathbf{p} = [\mathrm{f}_x \ \mathrm{f}_y \ \mathrm{u}_0 \ \mathrm{v}_0 \ \mathrm{aa}_x \ \mathrm{aa}_y \ \mathrm{aa}_z \ \mathrm{t}_x \ \mathrm{t}_y \ \mathrm{t}_z]^\top$$

and the projection matrix expressed in camera parameters is thus

$$\mathtt{P} = \mathtt{K}[\mathtt{R}\ \mathtt{t}]$$

Decomposition into Camera Parameters

The left hand 3×3 submatrix of P equal to KR for a finite projective camera is non-singular. RQ decomposition decomposes a non-singular matrix into an upper-triangular matrix and orthogonal matrix. For code reuse we use the QR decomposition routine from Press *et al.* [105] remembering that the inverse of an upper-triangular matrix is still upper-triangular and the inverse of a rotation matrix is its transpose.

$$(\mathtt{KR})^{-1} = (\mathtt{R}^\top \mathtt{K}^{-1}) \quad .$$

Care must be taken to align K to our conventions of negative focal length values which then forces the alignment of the rotation matrix accordingly. The translation component of the camera parameters is then given by

$$\mathbf{t} = \mathtt{K}^{-1}\mathbf{p}_4$$

Bibliography

[1] T. B. Sheridan. *Telerobotics, Automation, and Human Supervisory Control*. The MIT Press, Cambridge MA, 1992.

[2] R. C. Goertz and R. C. Thompson. Electronically controlled manipulators. *Nucleonics*, pages 46–47, 1954.

[3] G. Hirzinger. Rotex - the first robot in space. In *Proc. of International Conference on Advanced Robotics*, 1993.

[4] J.R. Vadus. International status and utilization of undersea vehicles. In *Proc. of Inter-Ocean Conference*, June 1976.

[5] E.L. Grimson, R Kikinis, A. Jolesz, and P.M. Black. Image guided surgery. *Scientific American*, June 1999.

[6] W.S. Kim. Developments of new force-reflecting control scheme and an application to a teleoperation training simulator. In *Proc IEEE Int Conf on Robotics and Automation, Los Alamitos, CA*, pages pp. 1412–1419, 1992.

[7] R. W. Daniel and P. R. McAree. Fundamental limits of performance for force reflecting teleoperation. *International Journal of Robotics Research*, 17(8):811–830, 1998.

[8] M.V. Noyes and T.B. Sheridan. A novel predicator for telemanipulation through a time delay. In *Proc Annual Conference on Manual Control, Moffett Field, CA, NASA Ames Research Center*, 1984.

[9] G. Hirzinger, J. Heindl, and K. Landzettel. Predictor and knowledge-based telerobotic control concepts. In *Proc IEEE Int Conf on Robotics and Automation, Scottsdale, SA*, pages pp. 1768–1777, May 14-19 1989.

[10] H.W. Stone. Mars pathfinder microrover: A low-cost, low-power spacecraft. In *Proc. AIAA Forum on Advanced Developments in Space Robotics, Madison, WI.*, 1996.

[11] V. Ranadive. Video resolution, frame rate, and gray-scale tradeoffs under limited bandwidth for undersea teleoperation. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, 1979.

[12] B. J. Deghuee. Operator-adjustable frame-rate, resolution and gray-scale tradeofss in fixed bandwidth remote manipulator control. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, 1980.

[13] M. Barth, T. Burkert, C. Eberst, N.O. Stöffler, and G. Färber. Photo-realistic scene prediction of partially unknown environments for the compensation of time delay in telepresence applications. In *Proc IEEE Int Conf on Robotics and Automation, San Francisco, CA*, 2000.

[14] D.A. Lawrence. Stability and transparency in bilateral teleoperation. *IEEE Transactions on Robotics and Automation*, 1993.

[15] M. Posner, M. Nissen, and R. Klein. Visual dominance: An information-processing account of its origins and significance. *Psychological Review*, 83(2):157–171, 1976.

[16] T.C.Jordan. Characteristics of visual and proprioceptive response times in the learning of a motor skill. *Quarterly Journal of Experimental Psychology*, 24:pp. 536–543, 1972.

[17] A.J. Bray. *Recognizing and Tracking Polydrehal Objects*. PhD thesis, University of Sussex, UK, 1990.

[18] D. 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.

[19] D.B. Gennery. Visual tracking of known three-dimensional objects. *International Journal of Computer Vision*, 7(3):243–270, 1992.

[20] C. G. Harris. Tracking with rigid models. In A. Blake and A. Yuille, editors, *Active Vision*. MIT Press, Cambridge, MA, 1992.

[21] D.G. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):441–450, May 1991.

[22] D.G. Lowe. Robust model-based motion tracking through the integration of search and estimation. *International Journal of Computer Vision*, 8(2):113–122, August 1992.

[23] G.D. Sullivan. Visual interpretation of known objects in constrained scenes. *Phil Trans Roy Soc Lond B*, 337:361–370, 1992.

[24] W.J. Wilson, C.C. Williams Hulls, and G.S. Bell. Relative end-effector control using cartesian position based servoing. *IEEE Trans. Robotics and Automation*, 12(5):684–696, October 1996.

[25] R. Cipolla and N.J. Hollinghurst. Human-robot interface by pointing with uncalibrated stereo vision. *Image and Vision Computing*, 16(1):171–178, 1996.

[26] G.D. Hager, S. Hutchinson, and P. Corke. A tutorial introduction to visual servo control. *IEEE Transactions on Robotics and Automation*, 12(5):661–670, 1996.

[27] R.M. Haralick, C.N. Lee, K. Ottenberg, and M. Nölle. Review and analysis of solutions of the three point perspective pose estimation problem. *International Journal of Computer Vision*, 13(3):331–356, 1994.

[28] M.A. Fischler and R.C. Bolles. Random sample concensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. ACM*, 24(6):381–395, 1981.

[29] W. Förstner. Reliability analysis of parameter estimation in linear models and applications to mensuration problems in computer vision. *Computer Vision, Graphics, and Image Processing*, 1987.

[30] R. Horaud, B. Conio, O. Leboulleux, and B. Lacolle. An analytic solution for the perspective 4-point problem. *Computer Vision, Graphics, and Image Processing*, 47:33–44, 1989.

[31] L. Quan and Z. Lan. Linear n-point camera pose determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1999.

[32] D. Huttenlocher and S. Ullman. Object recognition using alignment. In *Proceedings, First International Conference on Computer Vision*, pages 102–111, London, UK, 1987.

[33] D. Dementhon and L.S. Davis. Model-based object pose in 25 lines of code. *International Journal of Computer Vision*, 15:123–141, 1995.

[34] D.B. Gennery. Visual tracking of known objects. *International Journal of Computer Vision*, 7(3):243–270, 1992.

[35] J. Heuring and D.W. Murray. Modeling and copying human head movements. *IEEE Journal of Robotics and Automation*, 1999.

[36] M. Armstrong and A. Zisserman. RoRAPiD: A robust object tracker. Unpublished report, 1997.

[37] D.G. Lowe. Robust model-based motion tracking through the integration of search and estimation. *International Journal of Computer Vision*, 8(2):113–122, August 1992.

[38] P.J. Huber. *Robust Statistics*. John Wiley and Sons, New York, 1981.

[39] E. Marchand, P. Bouthemy, F. Chaumette, and V. Moreau. Robust real-time visual tracking using a 2D-3D model-based approach. In *Proc 7th Int Conf on Computer Vision, Kerkira, Greece, September 1999*, 1999.

[40] T. Drummond and R. Cipolla. Real-time tracking of multiple articulated structures in multiple views. In *Proc 6th European Conf on Computer Vision, Dublin, Ireland, June 2000*, 2000.

[41] Z. Li and H. Wang. Real-time 3d motion tracking with known geometric models. *Real-Time Imaging*, 1999.

[42] R. Horaud, F. Dornaika, B. Lamiroy, and S. Christy. Object pose: the link between weak perspective, paraperspective, and full perspective. *International Journal of Computer Vision*, 22(2):173–189, March 1997.

[43] Isard M. and Blake A. CONDENSATION – conditional density propagation for visual tracking. *International Journal of Computer Vision*, 1998.

[44] D. Forsyth and J. Ponce. *Computer Vision – A modern approach.* unpublished, 2001. Unpublished.

[45] H. Sidenbladh, M. J. Black, and D. J. Fleet. Stochastic tracking of 3D human figures using 2D image motion. In *Proc 6th European Conf on Computer Vision, Dublin, Ireland, June 2000*, 2000.

[46] D. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. In *Proc of the IEEE Conf on Computer Vision and Pattern Recognition*, 2000.

[47] D. W. Schloerb. A quantitative measure of telepresence. *Presence*, 4(1):64–80, 1995.

[48] M.O. Ernst and M.S. Banks. Humans integrate visual and haptic information in a statistically optimal fashion. *Unpublished report*, 2001.

[49] T. B. Sheridan. Space teleoperation through time delay : Review and prognosis. *IEEE Transactions on Robotics and Automation*, 9(5):592–606, October 1993.

[50] W.S. Kim and A. K. Bejczy. Demonstration of a high-fidelity predictive/preview display techniques for telerobotic servicing in space. *IEEE Transactions on Robotics and Automation*, 9(5):pp. 698–702, October 1993.

[51] W. Kim. Virtual reality calibration and preview/predictive displays for telerobotics. *Presence*, 5(2):173–190, Spring 1996.

[52] J.J.Gibson. Adaptation, after-effect and contrast in the perception of curved lines. *Journal of Experimental Psychology*, 16:1–31, 1933.

[53] I. Rock and J. Victor. Vision and touch: An experimentally created conflict between two senses. *Science*, 143:594–596, 1964.

[54] M.O. Ernst, M.S. Banks, and H.H.Bülthoff. Touch can change visual slant perception. *Nature Neuroscience*, 3(1):pp. 69–73, 2000.

[55] R.L. Klatzky, K.A. Purdy, and S.J. Lederman. When is vision useful during manipulatory tasks. *Proc of fifth annual symposium on haptic interfaces for virtual environment and teleoperator systems, Atlanta , Georgia, USA, DSC-Vol. 58, ASME/IMECE*, 21 November 1996.

[56] I. Rock. *The Nature of Perceptual Adaptation.* Basic Books, New York, 1966.

[57] C.S. Harris. Perceptual adaptation to inverted, reversed, and displaced vision. *Psychological Review*, 72:419–444, 1965.

[58] R.B. Welch. *Perceptual Modification: Adapting to Altered Sensory Environments.* Academic, New York, 1978.

[59] G.M. Stratton. Vision without inversion of the retinal image. *Psychological Review*, 4:341–360, 463–481, 1897.

[60] H. von Helmholtz. *Handbuch der Physiologischen Optik*, volume III. Leopold Voss, Leipzig, 1867.

[61] J.R. Flanagan and A.M. Wing. The rôle of internal models in motion planning and control : Evidence from grip force adjustments during movements of hand-held loads. *Journal of Neuroscience*, 17:1519–1528, 1997.

[62] J.R. Flanagan, D.M. Wolpert, and R.S. Johansson. Sensorimotor prediction and memory in object manipulation. *Canadian Journal of Experimental Psychology*, 55(2):89–97, 2001.

[63] R. C. Miall and D.M. Wolpert. Forward models for physilogical motor control. *Neural Networks*, 9:pp. 1265–1279, 1996.

[64] C.M.Harris and D.M.Wolpert. Signal-dependent noise determines motor planning. *Nature*, 394:pp. 780–784, 20 August 1998.

[65] M. Massimino and T. Sheridan. Teleoperator performance with varying force and visual feedback. *Human Factors*, 36(1):145–157, 1994.

[66] J. Hill. Two measures of performance in a peg-in-hole manipulation task with force feedback. In *13th Annual Conference on Manual Control*, Cambridge, MA: MIT, 1977.

[67] T. L. Brooks. Superman: a system for supervisory manipulation and the study of human computer interaction. Master's thesis, MIT, Cambridge MA, 1979.

[68] J. Vertut and P. Coiffet. *Teleoperation and Robotics Vol 3A – Evolution and Development*. Kogan Page, London, 1985.

[69] J. Vertut and P. Coiffet. *Teleoperation and Robotics Vol 3B – Applications and Technology*. Kogan Page, London, 1985.

[70] P. Fischer, R. W. Daniel, and K. V. Siva. Specification and design of input devices for teleoperation. In *Proceedings of IEEE International Conference on Robotics and Automation, Cincinnati*, 1990.

[71] P. Fischer. *A novel input device for force control of robots*. PhD thesis, Department of Engineering Science, University of Oxford, 1993.

[72] J.J. Craig. *Introduction to robotics, mechanics and control*. Addison-Wesley, 2nd edition, 1989.

[73] P.J. Fischer and R.W. Daniel. Real time kinematics for a 6 dof telerobotic joystick. In *Proc. Symposium on Theory and Practice of Robots and Manipulators, Udiné*, September 1-4 1992.

[74] ATI industrial automation. http://www.ati-ia.com.

[75] QNX software systems ltd. http://www.qnx.com.

[76] D.P. Jang, J.H. Ku, , M.B. Shin, Y.H. Choi, and S.I. Kim. Objective validation of the effectiveness of virtual reality psychotherapy. *CyberPsychology and Behaviour*, 3(3), 2000.

[77] H. Das. *Kinematic Control and Visual Display of Redundant Teleoperators*. PhD thesis, MIT, 1989.

[78] Kama W.N and DuMars R.C. Remote viewing: a comparision of direct viewing, 2D and 3D television. Technical report, Wright Patterson AFB, 1964.

[79] Chubb G.P. A comparision of performance in operating the CRL-8 master-slave manipulator under monocular and binocular viewing conditions. Technical report, Wright Patterson AFB, 1964.

[80] S. Levy, T. Munzner, M Phillips, et al. Geomview: 3D visualization software. Technical report, Software Development Group, Geometry Center, University of Minnesota, 1996. http://www.geom.umn.edu/software/geomview.

[81] P. Hintjens. Libero: FSM programming tool. Technical report, iMatix Corporation, 1998. http://www.imatix.com/html/libero/index.htm.

[82] The Mathworks, Inc. Stateflow. http://www.mathworks.com.

[83] TargetJr image understanding software. http://www.esat.kuleuven.ac.be/ targetjr.

[84] G.D. Hager, W-C. Chang, and A.S. Morse. Robot hand-eye coordination based on stereo vision. *IEEE Control System Magazine*, 15:30–39, 1995.

[85] R. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(4):pp. 323–344, August 1987.

[86] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521623049, 2000.

[87] S.J. Maybank and O. Faugeras. A theory of self-calibration of a moving camera. *International Journal of Computer Vision*, 8(2):123–151, 1992.

[88] W. Triggs. Autocalibration and the absolute quadric. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition, Puerto Rico.*, pages pp. 609–614. IEEE Computer Society Press, 1997.

[89] A. Heyden and K. Åström. Euclidean reconstruction from constrain intrinsic parameters. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition, Puerto Rico.* IEEE Computer Society Press, 1996.

[90] M. Pollefeys, R. Koch, and L. Gool. Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. In *Proc. ICCV98*, page pp., 1998.

[91] L. de Agapito, E. Hayman, and I. Reid. Self-calibration of a rotating camera with varying intrinsic parameters. In *Proc 9th British Machine Vision Conf, Southampton*, pages 105–114, 1998.

[92] W. Triggs. Autocalibration from planar scenes. In *Proc 5th European Conf on Computer Vision, Freiburg, Germany, June 1998.* Springer-Verlag, 1998.

[93] R.I. Hartley. Self-calibration from multiple views with a rotating camera. In *Proc 3rd European Conf on Computer Vision, Stockholm*, volume 1, pages 471–478, 1994.

[94] A. Zisserman, P. A. Beardsley, and I. D. Reid. Metric calibration of a stereo rig. In *Proc IEEE Workshop on Representations of Visual Scenes, Boston*, pages 93–100. IEEE Computer Society Press, 1995.

[95] P.F. McLauchlan and D.W. Murray. Active camera calibration for a Head-Eye platform using the variable State-Dimension filter. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(1):15–22, 1996.

[96] E. Grossmann and J.S. Victor. The precision of 3d reconstruction from uncalibrated views. In *Proc 9th British Machine Vision Conf, Southampton*, 1998.

[97] J.-M. Lavest, M. Viala, and M. Dhome. Do we really need an accurate calibration pattersn to achieve a reliable camera calibration? In *Proc 5th European Conf on Computer Vision, Freiburg, Germany, June 1998*, pages 158–174, Berlin, 1998. Springer.

[98] C.M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.

[99] J.M. Hollerbach and C.W. Wampler. The calibration index and taxonomy for robot kinematic calibration methods. *International Journal of Robotics Research*, 15, 1996.

[100] Stäubli S.A. Puma 500 mark 3 equipment manual, 1992.

[101] K.L. Conrad, P.S. Shiakolas, and T.C. Yih. Robotic calibration issues: Accuracy, repeatability and calibration. *Proceedings of the 8th Mediterranean Conference on Control and Automation (MED2000), Rio, Patras, Greece*, 2000.

[102] R.W.Daniel. Manipulator encoder errors. Private correspondence.

[103] K. Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly Applied Mathematics*, 1944.

[104] D.W. Marquardt. An algorithm for least squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11:pp. 431–441, 1963.

[105] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes in C.* Cambridge University Press, 1988.

[106] S.B. Pollard, T.P. Pridmore, J. Porrill, J.E.W. Mayhew, and J.P. Frisby. Geometrical modelling from multiple stereo views. *International Journal of Robotics Research*, 8(4):3–32, 1989.

[107] D.W. Murray, D.A. Castelow, and B.F. Buxton. From image sequences to recognized moving polyhedral objects. *International Journal of Computer Vision*, 3(3):181–208, 1989.

[108] C. Harris and C. Stennett. RAPiD – a video rate object tracker. In *Proc 1st British Machine Vision Conf, Oxford*, pages 73–78, 1990.

[109] J.J. Heuring. *Merging Computer Telepresence and Computer Vision.* PhD thesis, Department of Engineering Science, University of Oxford, 1998.

[110] M. Armstrong and A. Zisserman. Robust object tracking. In *Asian Conference on Computer Vision, Volume I*, pages 58–61, 1995.

[111] N. Maitland and C. Harris. A video based tracker for use in computer aided surgery. In *Proc 5th British Machine Vision Conf, York*, pages 609–681, 1994.

[112] A.C.F Colchester, J. Zhao, K.S. 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.

[113] Rousseeuw P.J. Least median of squares regression. *J. Amer. Statis. Assoc.*, 1984.

[114] Rousseeuw P.J. and Leroy A.M. *Robust Regression and Outlier Detection.* New York: Wiley, 1987.

[115] P.H.S. Torr and D.W. Murray. Outlier detection and motion segmentation. In *Proc SPIE Sensor Fusion VI*, pages 432–443, Boston, September 1993.

[116] R.J. Evans. Filtering of pose estimates generated by the RAPiD tracker in applications. In *British Machine Vision Conference, Oxford*, pages 79–84, 1990.

[117] Faugeras O.D. and Hébert M. The representation, recognition, and locating of 3D objects. *International Journal of Robotics Research*, 1986.

[118] J.K. Mills. Manipulator transition to and from contact tasks: a discontinuous control approach. In *Proc IEEE Int Conf on Robotics and Automation, Cincinnati OH, 1990*, pages 440–446, Los Alamitos CA, 1993. IEEE Computer Society Press.

[119] O. Khatib and J. Burdick. Motion and force control of robot manipulators. In *Proc IEEE Int Conf on Robotics and Automatic, San Francisco, 1986*, pages 1381–1386, 1986.

[120] Y.F. Li and R.W. Daniel. Robot tip velocity measurement for direct end effector position control. In *Proc IEEE Int Conf on Intelligent Robots and Systems, Raleigh NC, 1992*, pages 998–1103. IEEE Computer Society Press, 1992.

[121] Y.F. Li. A sensor-based robot transition control strategy. *The International Journal of Robotics Research*, 15(2):pp. 128–136, April 1996.

[122] R. McAree and R.W. Daniel. Stabilizing impacts in force reflecting teleoperation using distance to impact estimates. Preprint, January 1999.

[123] T. L. Brooks. Telerobot reponse requirements. a position paper on control requirements for the FTS telerobot. *Report No. STC/ROB/90-93*, 1990.

[124] R.W. Daniel. Controls for teleoperation. *Proceedings of the Workshop on Human and Machine Haptics*, 1997.

[125] P.M.Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Experimental Psychology*, 47(6):pp. 381–391, June 1954.

[126] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, Inc, NewYork, 1991.

[127] R.G. Brown and P.Y.C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering with MATLAB*. John Wiley and Sons, 1996.

[128] G.A. Calvert, M.J. Brammer, and S.D. Iversen. Crossmodal identification. *Trends in Cognitive Science*, 2:247–253, 1998.

[129] H. McGurk and J. MacDonald. Hearing lips and seeing voices. *Nature*, 264:746–748, 1976.

[130] D.M. Wolpert, R. C. Miall, and M. Kawato. Internal models in the cerebellum. *Trends in Cognitive Sciences*, 2(9):pp. 338–347, September 1998.

[131] S.J. Jullier and J.K. Uhlmann. A new extension of the kalman filter to nonlinear systems. In *Proc. of AeroSense: The 11th International Symposium on Aeroapace/Defence Sensing, Simulation and Controls, Orlando, Florida*, 1997.

[132] G.W. Oehlert. *A First Course in Design and Analysis of Experiments.* Freeman, W.H., 2000.

[133] A.J. Davison and D.W. Murray. Mobile robot localization using active vision. In *Proc 5th European Conference on Computer Vision, Freiburg, Germany, 1998*, Lecture Notes in Computer Science Vol 1407, pages 809–825, Berlin, 1998. Springer Verlag.