Combining observed and predicted data for robot vision in poor visibility

Rustam Stolkin

Submitted for the degree of
Doctor of Philosophy
of the
University of London

Centre for Advanced Instrumentation Systems
University College London

March 2004

Abstract

This thesis addresses the problems of recovering the 3D position and orientation of a vehicle mounted camera relative to a known object and, additionally, tracking the 2D position of that object in camera images, under conditions of extremely poor visibility such as encountered underwater. The human visual system can often make correct interpretations of images that are of such poor quality that they contain insufficient explicit information to do so. It is asserted that such systems must therefore make use of prior knowledge in several forms.

A novel algorithm (the EM/E-MRF algorithm) is presented for the interpretation of scene content and camera position from extremely poor visibility images. The algorithm is capable of tracking camera trajectories over extended image sequences. The algorithm combines observed data (the current image) with predicted data derived from prior knowledge of the object being viewed and an estimate of the camera's motion.

During image segmentation, a predicted image is used to estimate class conditional probability distributions and an Extended-Markov Random Field technique is used to combine observed image data with expectations of that data within a probabilistic framework. Markov dependency is extended to include contributions from corresponding pixels in the predicted image. Interpretations of scene content and camera position are then mutually improved using Expectation-Maximisation.

The resulting algorithm exhibits elements of continuous machine learning. Non-rigid statistical models of object being viewed and background are continuously modified and updated during the analysis of each frame of the video sequence.

Poor visibility image sequences of known objects, filmed along pre-measured trajectories with a calibrated camera have been constructed in order to provide real test data with underlying ground-truth. An industrial robot arm was used to move a camera along a highly repeatable trajectory. Test sequences, (featuring an object of interest in extremely poor visibility generated using dry ice fog), and calibration sequences (featuring calibration targets in good visibility) were filmed along identical trajectories. Camera intrinsics, lens distortion parameters and camera position and orientation could be extracted from the calibration sequences for every frame. This information was used to provide ground-truth for corresponding frames in the poor visibility test sequences.

Using this data, the EM/E-MRF algorithm has been tested on several hundred images, over a range of visibility conditions, camera trajectories, algorithm parameters and observed objects.

Acknowledgements

I thank my supervisors, John Gilby and Alistair Greig, for their guidance, support and hands on help. Thanks are also due to Mark Hodgetts who supervised the first 15 months of the project and my first publication.

I am indebted to my mother, father and older brother, who have all helped to make life easier for me throughout the course of this work, and to dearest Bonny for great understanding, persistent encouragement, constant feeding, special holidays and much more.

This research was undertaken within the Postgraduate Training Partnership established between Sira Ltd. and University College London. Postgraduate Training Partnerships are a joint initiative of the Department of Trade and Industry and the Engineering and Physical Sciences Research Council. They are aimed at fostering closer links between the academic science base, industrial research and industry.

Contents

List of figures

List of symbols

Chapter 3

$\underline{I}$ The set of grey-levels (intensities) for all pixels of an image.

$\underline{C}$ The set of class labels for all pixels of an image.

$\underline{\theta}$ The vector of six co-ordinates representing the position and orientation of the camera for a particular image frame.

$\hat{\underline{\theta}}^n$ The current estimate of $\underline{\theta}$ at the n^{th} iteration of the *EM* algorithm.

$I_{i,j}$ The grey-level (intensity) of an individual pixel (i^{th} from the left and j^{th} from the top) in an image.

$C_{i,j}$ The class label of an individual pixel (i, j) in an image.

$\hat{C}_{i,j}$ The predicted class label of an individual pixel (i, j) in an image.

$\hat{\underline{C}}^{proj}$ The set of class-labels predicted by projecting a predicted image using the current estimate of the camera co-ordinates $\hat{\underline{\theta}}^n$.

$\hat{C}_{i,j}^{proj}$ The class-label of an individual pixel (i, j), predicted by projecting a predicted image using the current estimate of the camera co-ordinates $\hat{\underline{\theta}}^n$.

$U_{i,j}$ The exponential part of a Gibbs distribution.

$J(a,b)$ A function for determining the consistency between possible class labels for two pixels in a neighbourhood.

Z A normalising constant which prevents a Gibbs distribution returning probabilities greater than one.

S_1, S_2 Weighting factors which determine the significance to the prior probability term of the class values of nearest neighbour pixels and predicted pixels respectively.

σ^2 Variance.

μ Mean.

$\sigma^2_{c_{i,j}}$ The variance of the class conditional distribution of pixel intensities that corresponds to the choice of class $C_{i,j}$ that is currently being considered for pixel (i, j).

$\mu_{c_{i,j}}$ The mean of the class conditional distribution of pixel intensities that corresponds to the choice of class $C_{i,j}$ that is currently being considered for pixel (i, j).

(x, y, z) The translation component of a rigid body transformation.

$(\omega_x, \omega_y, \omega_z)$ A vector whose direction defines an axis of rotation and whose magnitude defines the amount of rotation about that axis in radians.

$\mathbf{R}$ A rotation matrix.

r_{mn} A component of a rotation matrix (mth row and nth column).

x_k The "state" vector of a Kalman filter at the kth iteration.

$\mathbf{A}$ The "system model" of a Kalman filter.

w_k The noise model component of a Kalman filter at the kth iteration.

q A quaternion.

$(s, \mathbf{v})$ The *s*calar and **v**ector components of a quaternion, q, such that $q = (s, \mathbf{v}) = s + v_x i + v_y j + v_z k$.

$\bar{q}$ The conjugate of the quaternion q.

u A factor which determines the degree of interpolation between the predicted (via trajectory extrapolation) camera position and the measured (via the EM/E-MRF algorithm) camera position.

d The distance between the end of a cylinder and the intersection between the cylinder axis and the shortest line joining the axis to a ray.

L The length of the projection onto a cylinder axis of the portion of a ray which connects the point of intersection of the ray with the cylinder surface to the point of closest approach between the ray and the cylinder axis.

Chapter 4

$\underline{\mathbf{X}}_{\mathrm{c}}$ The vector representing a 3D point in the camera co-ordinate system.

$\mathrm{X_c, Y_c, Z_c}$ The Cartesian components of $\mathbf{X}$.

$\underline{\boldsymbol{x}}_c$ The vector representing the 2D point which is the projection of $\mathbf{X}$ onto the image plane of the camera.

x_c, y_c The Cartesian components of $\boldsymbol{x}$.

$\mathrm{O_c}$ The origin of the camera co-ordinate system.

$\underline{\mathbf{X}}_{\mathrm{w}}$ The vector representing a 3D point in the world co-ordinate system.

$\mathrm{X_w, Y_w, Z_w}$ The Cartesian components of $\underline{\mathbf{X}}_{\mathrm{w}}$.

$\mathrm{O_w}$ The origin of the world co-ordinate system.

u, v The co-ordinates of $\boldsymbol{x}$ in terms of number of pixels vertically and horizontally from the top, left hand corner of the image.

$\underline{\mathrm{x}}_i$ The 3×1 homogeneous vector representing the point (u, v) in units of pixels.

u_0, v_0 The co-ordinates of the principal point in terms of number of pixels vertically and horizontally from the top, left hand corner of the image.

k_u, k_v The number of pixels per unit length in the u and v directions.

f The focal length of the camera.

α An intrinsic camera parameter which relates x_c to u. Note: $\alpha = fk_u$.

β An intrinsic camera parameter which relates y_c to v. Note: $\beta = fk_v$.

γ A parameter describing the skewness between the u and v axes.

$\mathbf{C}$ The 3×3 calibration matrix of intrinsic camera parameters.

$\mathbf{R}$ A rotation matrix.

$\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3$ The three column vectors of $\mathbf{R}$.

$\mathbf{T}$ A 3×1 translation vector.

$\underline{\mathbf{X}}_{\mathbf{t}}$ The homogeneous vector representing a 3D point in a target co-ordinate system.

X_t, Y_t	The Cartesian components of $\underline{\mathbf{X}}_\mathbf{t}$.
$\mathbf{H}$	The 3×3 matrix representing an homography.
$\mathbf{h}_1$, $\mathbf{h}_2$, $\mathbf{h}_3$	The three column vectors of $\mathbf{H}$.
h_{mn}	The element of $\mathbf{H}$ located on the m^{th} row and the n^{th} column.
$\mathbf{E}$	The matrix of extrinsic camera parameters.
w	The arbitrary scaling factor associated with a homogeneous position vector.
B_{mn}	The element located on the m^{th} row and the n^{th} column of the matrix resulting from the product $\mathbf{C}^{-T}\mathbf{C}^{-1}$.
$\mathbf{b}$	A vector containing the independent elements of $\mathbf{C}^{-T}\mathbf{C}^{-1}$.
$\mathbf{v}_{ij}$	A vector combining elements of $\mathbf{h}_i$ and $\mathbf{h}_j$, such that it is possible to express $\mathbf{h}_i^T\mathbf{C}^{-T}\mathbf{C}^{-1}\mathbf{h}_j$ in the form $\mathbf{v}_{ij}^T\mathbf{b}$.
$\mathbf{Vb}$	A stacking of many examples of $\mathbf{v}_{ij}^T\mathbf{b}$.
$(\hat{u},\hat{v})$	The result of radially distorting the image co-ordinates (u,v).
r	The radial distance of a pixel from the principal point.
k_1, k_2	Radial distortion parameters (quadratic and quartic respectively).
$\mathbf{X}_{target_{ts}}, \mathbf{x}_{image_{ts}}$	A target point in world co-ordinates and its observed image in pixelated camera co-ordinates. For a spot, s, on a target, t.
$\hat{\mathbf{x}}_{image_{ts}}$	The expected image of $\mathbf{x}_{image_{ts}}$ given the current estimates of the camera parameters.

Chapter 5

S_1, S_2	Weighting factors which determine the significance to the prior probability term of the class values of nearest neighbour pixels and predicted pixels respectively.
u	A factor which determines the degree of interpolation between the predicted (via trajectory extrapolation) camera position and the measured (via the EM/E-MRF algorithm) camera position.

1 Introduction

1.1 Overview

This thesis addresses the problem of vision-based navigation in conditions of extremely poor visibility, such as encountered by remote operated vehicles (ROVs) in underwater environments. A motivation for this work is the visual inspection of submerged components of off-shore oil rig structures.

Most robot vision systems are designed for good visibility conditions and typically rely on extracting detailed features, such as edges, lines or corners, from observed images. This kind of feature extraction is unfeasible in conditions of extremely poor visibility as is demonstrated in the following section (section 1.2).

In contrast, the human visual system can often understand the content of images that are of such poor quality that conventional computer vision algorithms fail. It might be argued (Ullman [1996]) that such images do not actually contain enough explicit information to enable correct interpretation. It therefore seems likely that such systems (e.g. human) must make use of prior knowledge in several forms.

This thesis presents the Expectation Maximisation/Extended-Markov Random Field (EM/E-MRF) algorithm, for the interpretation of scene content and camera position from poor quality images. This algorithm combines observed data (the current image) with predicted data derived from prior knowledge of the object being viewed and an estimate of the camera's motion.

An Extended-Markov Random Field technique (See section 3.3) is used to combine observed image data with expectations of that data during image segmentation, within a probabilistic framework. Interpretations of scene content and camera position are then mutually improved using Expectation-Maximisation. The resulting algorithm exhibits elements of continuous machine learning.

To validate these ideas, it was necessary to construct poor visibility image sequences with known ground truth. This must include known models of the object being viewed, a known model of the camera's intrinsic calibration parameters (focal length, principal point position and pixel aspect ratio), a known model of lens distortion and a known camera position and orientation for each frame of each image sequence.

An industrial robot arm was used to move a camera along a highly repeatable trajectory. Test sequences, (featuring an object of interest in extremely poor visibility generated using dry ice fog), and calibration sequences (featuring calibration targets in good visibility) were filmed along identical trajectories. Camera intrinsics, lens distortion parameters and position and orientation could be extracted from the calibration sequence for every frame. This information was used to provide ground truth for the corresponding poor visibility test sequences.

Image sequences with known ground truth were constructed with various different known objects, different degrees of poor visibility and various different camera trajectories. The EM/E-MRF algorithm was tested on these image sequences and the camera position estimates, output by the vision system, were compared with the pre-measured ground truth. The performance of the algorithm has been examined in response to various different conditions. Sources of error and limitations of the algorithm have been high-lighted and suggestions have been made as to how this work might be extended in the future.

1.2 Machine vision in poor visibility

The vast majority of machine vision systems are designed to perform in good visibility through a clear medium which is assumed not to interfere with the relationship between world and image. Unfortunately, poor visibility is inescapable. Outdoor applications are subject to the vagaries of the weather and the atmosphere including haze, fog, rain, hail and snow. Even indoor environments will not provide perfect visibility because of inadequate lighting, shadow, clutter and occlusion. Underwater (and other poor visibility) applications suffer from a variety of forms of image degradation including:

- Radial lens distortion (barrelling).
- Non-uniform lighting (lighting intensity varies with position in image).
- Dynamic lighting (lights move with vehicle, lighting conditions vary with time).
- Camera saturation.
- Shadow.
- Occlusion.
- Attenuation.
- Reflection and back-scattering.
- Blur (both focal blur and motion blur).
- Discrepancies between real objects and their models.

Hardly any reported vision systems are designed to cope with very poor visibility. Occasionally papers appear in the computer vision and robotics literature (see section 2.4) which deal with underwater scenarios and these often claim robustness in poor visibility. However, these invariably still rely on extracting conventional features (typically edges) using conventional techniques. In contrast, this thesis addresses the problems of image sequences for which visibility is so poor that conventional feature detection is impractical.

In order to illustrate what is meant by "poor visibility" in this thesis and in order to demonstrate the difficulties of applying conventional computer vision approaches to these conditions, figures 1.1-1.4 present a selection of poor visibility images and their corresponding edge detected versions (using the Canny edge detection method). It is apparent that attempting to locate structures by robustly identifying relevant edges under such conditions would pose challenging problems.

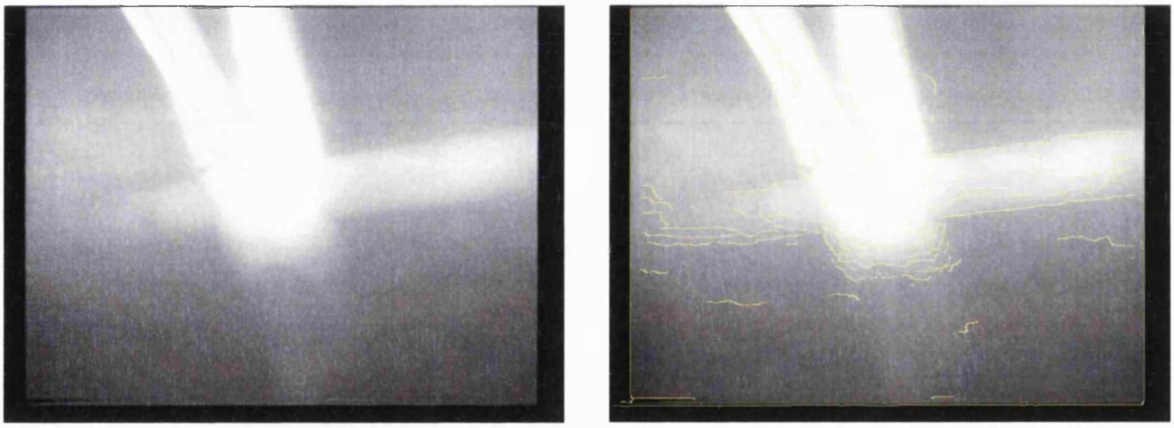


Figure 1.1 **Real, poor visibility image, frame grabbed from video footage featuring a scale model of an off-shore structure, filmed underwater at night from an ROV. The only illumination is from spotlights mounted on the vehicle. Left is original image and right is the result of edge detection.**

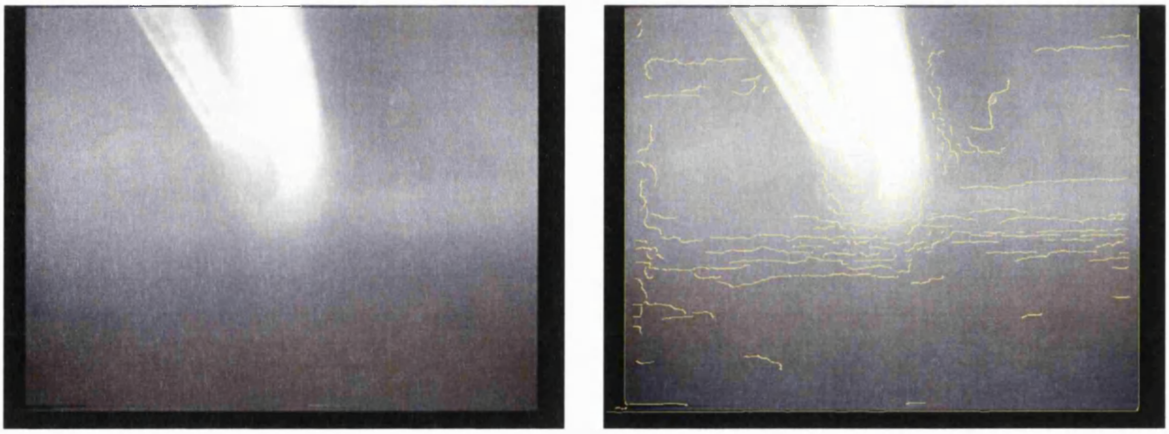


Figure 1.2 **Real, poor visibility image, frame grabbed from video footage featuring a scale model of an off-shore structure, filmed underwater at night from an ROV. The only illumination is from spotlights mounted on the vehicle. Left is original image and right is the result of edge detection.**

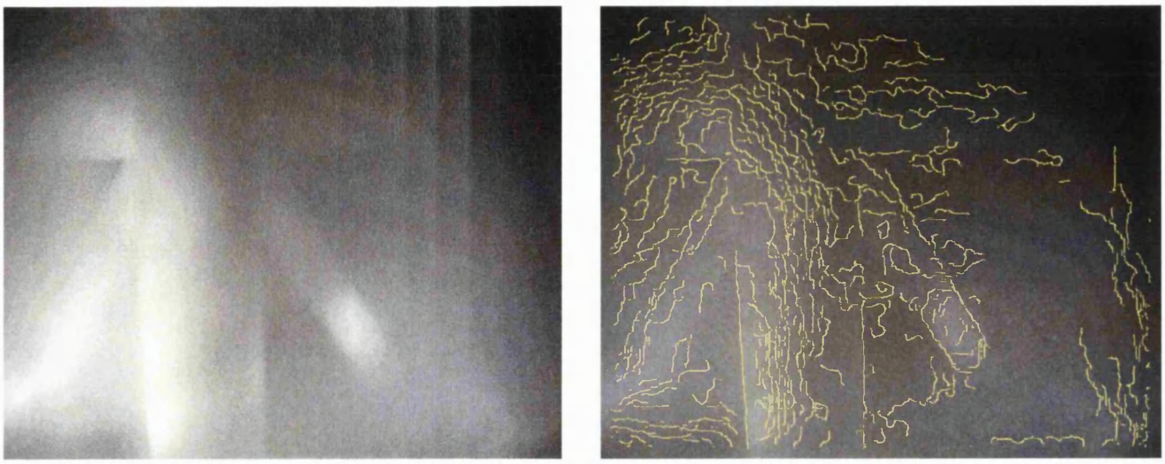


Figure 1.3 **Image from a poor visibility sequence filmed in the laboratory. The creation of these test sequences is described in chapter 4. The image features a model of an oil rig-like structure. Poor visibility conditions are created using dry ice fog. Illumination is from focussed beam spotlights mounted on and moving with the camera. Left is original image and right is the result of edge detection.**

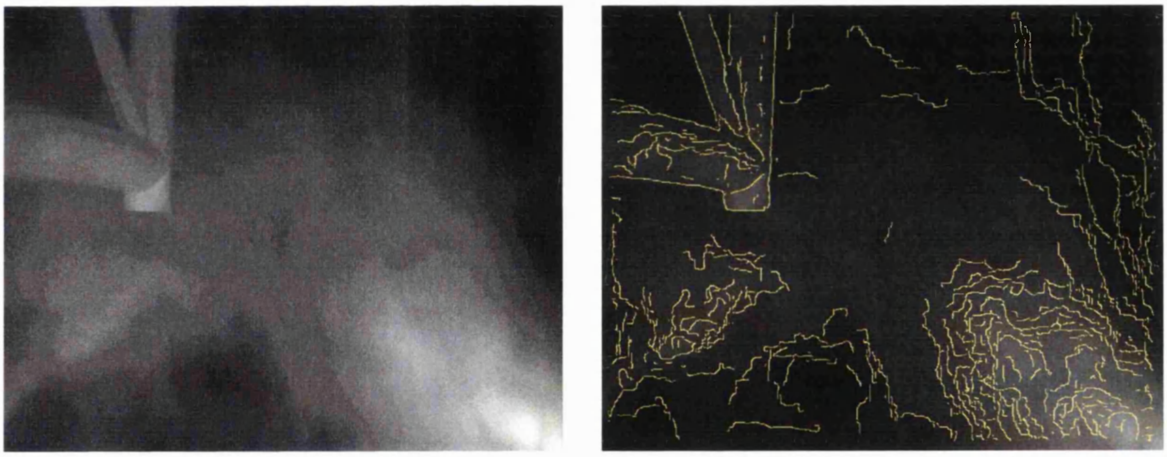


Figure 1.4 **Image from a poor visibility sequence filmed in the laboratory. The creation of these test sequences is described in chapter 4. The image features a model of an oil rig-like structure. Poor visibility conditions are created using dry ice fog. Illumination is from focussed beam spotlights mounted on and moving with the camera. Left is original image and right is the result of edge detection. Note, this image is used extensively in chapter 5 to demonstrate the EM/E-MRF algorithm.**

1.3 Note on image quality

It should be noted that the quality of images presented in this thesis may differ somewhat from that of the original digital data since the resolution of digital photographs may be finer than that of the printer.

Where frames taken from poor visibility image sequences are shown, these are usually linearly contrast stretched to aid the reader.

It may be noticed that the objects in some images appear to be upside down. This is due to the orientation in which the camera was attached to the robot arm during the filming of the image sequences (see chapter 4). The images are presented in their original form, as they were downloaded from the digital video cassettes, and have not been inverted in order to make objects appear the right way up.

1.4 Layout of this thesis

Chapter 2 reviews literature on various topics which are relevant to the work described herein. Areas of research that are examined include published algorithms for model based pose estimation and tracking, image segmentation techniques including Markov Random Fields, use of the Expectation Maximisation algorithm, research into computer vision in poor visibility conditions, the use of known ground truth in various forms for validating vision algorithms and the creation of image sequences with known ground truth, camera calibration methods and work that directly preceded the research described in this thesis.

Chapter 3 describes the EM/E-MRF robot vision algorithm in detail. The structure and motivation for the algorithm are presented in an intuitive fashion, deriving progressively from fundamental requirements of a machine vision system. It is demonstrated how the vision algorithm becomes equivalent to a form of Expectation Maximisation (EM) algorithm when iterated. The algorithm is justified mathematically by rooting it in probability theory, both as an expression of the EM algorithm and also from the point of view of Bayesian discrimination. The algorithm is summarised conveniently in a flow diagram. It may help the reader to view these parts in conjunction with section 5.2 in which the various stages of the algorithm are illustrated. This chapter also describes the practical details of the parameterisation of camera position and orientation, how these poses are interpolated and extrapolated and how the objects being viewed are measured and modelled.

Chapter 4 describes extensive practical work, carried out in the laboratory, to create poor visibility image sequences with known ground truth for the purpose of testing and validating the EM/E-MRF algorithm. The accuracy of the measured ground truth and calibration data is estimated and assessed in various ways and suggestions are made for improving the experimental procedure.

Chapter 5 presents the results of testing the EM/E-MRF vision algorithm on the poor visibility image sequences described in chapter 4. The algorithm is first tested on a single frame, examining the performance when subjected to different kinds of starting error. The algorithm is then tested on extended image sequences and the variation in performance is examined with respect to good and bad visibility conditions, smooth trajectories and those containing abrupt direction changes, different kinds of observed object and variation in important parameters of the algorithm relating to weightings in the use of observed and predicted data in several forms.

Chapter 6 contains a discussion of the results of chapter 5. The limitations of the EM/E-MRF algorithm in its present form are discussed as well as issues pertaining to the implementation of the algorithm at real time frame rates. Suggestions are made for how the work described in this thesis might be extended. These include possible improvements to the vision algorithm, improvements to the practical procedures for generating test sequences with known ground truth and suggestions for advances in the way that the performance of the vision algorithm can be tested, analysed and presented. The thesis is summarised and a list of those aspects of the work thought to constitute original contributions is provided.

2 Literature review

2.1 Model based pose estimation and tracking

Perhaps the most fundamental problem in robot vision is that of how to endow machines with the humanlike capabilities of being able to recognise known objects, distinguish these objects from some scene background and determine the location and orientation of the objects relative to the camera (or similarly the position and orientation of the camera relative to the known objects). These issues have been considered by many researchers throughout the brief history of robot vision development.

The above ideas are central to this thesis in which the EM/E-MRF algorithm is proposed as a way of distinguishing and tracking known objects in conditions of extremely poor visibility. A discussion of relevant literature in this field is important since it will emerge that the approach proposed in this thesis is distinct from those previously explored. Vision systems reported in the literature are almost exclusively designed for good visibility conditions in which conventional features, such as edges, corners and lines, are readily extractable, however the visibility conditions tackled in this thesis are so poor that conventional feature extraction is problematic (see section 1.2).

Besl and Jain [1985] is an early, theoretical and somewhat philosophical paper which attempts to formalise a clear statement and definition of the 3D object recognition problem. It suggests that vision systems should possess models of known objects, be able to determine the location (2D) of a known object in an image, and be able to determine the 3D position and orientation of the object in space.

Faugeras and Hebert [1986b] propose a method for recognising and locating objects. The scheme is based on range data from a laser range finder, which is used to construct a representation of the observed object in terms of “linear primitives” such as points, lines and planes. This representation can then be compared with representations of objects stored in memory.

Lowe [1992] presents a well-known procedure for 3D to 2D registration. An initial estimate of camera pose (relative to the object being viewed) is used to project a 3D model of the object into the image plane. Correspondences are assumed between model features and extracted image features which lie close to these model features. A probabilistic approach is used to select the best matches. Non-linear optimisation is then used to determine the rigid body transformation that best maps the model onto the image. This approach relies on a good initial pose estimate since if this differs too much from the true position then occlusion may hide important model features whilst features not visible in the image are brought into view. This may make it impossible for proper feature correspondences to be established.

Besl and McKay [1992] describe the iterative closest point (ICP) method for registering a model to a 3D data set. Each iteration of the ICP algorithm consists of two steps. Firstly, correspondence is assumed between model points and the closest data point. Secondly, a displacement is found which minimises the distance between corresponding pairs. It can be shown that the procedure converges to a minimum of positional error.

Wunsch and Hirzinger [1996] introduce a method for improved 3D to 2D registration performance. The algorithm combines the iterative closest point method with a 3D to 2D correspondence operator. The algorithm is an improvement over that of Lowe [1992], because it will converge even for large initial displacements.

TINA (Lacey et al. [2001]) is a set of tools for tackling image understanding problems. The initial focus of the project was the development of a 3D model matching system. This uses edges and depth maps extracted from pairs of binocular stereo images together with corresponding camera calibration information. Statistical matching of 3D scene descriptions to a stored wire-frame model enables the location of the model within the scene to be identified.

Once algorithms exist that enable a model to be registered to an image with the corresponding extraction of camera position, it is a natural extension to apply this process to an entire image sequence. The result is a tracking algorithm, which can distinguish objects moving along some trajectory relative to the camera and/or determine the camera trajectory relative to the observed object. Most significantly, extra information is now available since there is normally some relationship between consecutive images. If the trajectory can be modelled, predictions can be made about what might be expected to appear in the next image in the sequence. This a-priori information can improve the robustness of the registration algorithms and gives initial estimates for the registration process at each frame.

Early systems for 3D model-based motion tracking include that reported by Gennery [1982]. This system tracked Sobel edges within a five pixel range of predicted edges. The prediction involved velocity extrapolation and filtering. In earlier work, Gennery [1981] also addressed the issue of probabilistic evaluation of feature matches to a model. Verghese et al. [1988][1990] proposed a system for tracking 3D objects, based on the assumption that features are spatio-temporally dense (moving less than one pixel from frame to frame).

Harris [1992] describes the system known as RAPiD (Real-time Attitude and Position Determination). This is a model-based 3D tracking algorithm for a known object executing arbitrary motion and viewed by a standard video camera. The system matches high contrast edges from the image to markings, folds or edges projected from the 3D object model based on an initial position estimate derived from a Kalman filter. The set of measured displacements of these edges from those predicted is used to refine the estimate of model pose.

Drummond and Cipolla [1999][2000] present a three-dimensional model-based tracking system, incorporated into a visual servoing (camera on robot arm) system. The system uses a CAD model of the object to be tracked and matches this to the observed image in order to recover position and orientation at every frame. The approach is similar to that of the RAPiD system (Harris [1992]) and uses an estimate of motion trajectory to predict the object position in the next frame. This position estimate is then refined by measuring the displacements between projected model features and observed image features.

Christmas, Kittler and Petrou [1996] describe a system for tracking the pose of a camera relative to some 3D object for which a model exists. An initial (perhaps inaccurate) estimate of camera pose is used to project a 3D CAD model of the object into the image plane. A probabilistic 2D-2D matching algorithm is then used to determine correspondences between the observed image features and projected model features. These correspondences provide labels for the image features. A better estimate of camera position and orientation can then be computed. An iterative scheme is suggested in which successive refinements of camera pose are used as initial estimates for the following iterations of the algorithm. The authors suggest an application to navigation of an underwater vehicle observing an oil rig structure, however the images used in the work were filmed in air in a laboratory. The authors acknowledge that the poor visibility conditions often encountered in real underwater applications might make this approach unworkable.

Additional reported work on model based visual tracking includes Ginhoux et al. [2001], Kosaka et al. [1995], Braud et al. [1994], Jurie [1997]. Much of this work follows similar approaches to that already described or combines various similar techniques, employing models of the objects being tracked, predictions of position and fitting of the models to extracted features in some optimal way.

An alternative to using stored 3D object models is to use a small number of stored images of the object (see Ullman [1996]). Ullman suggests interpolating between stored images to synthesise predicted images from different viewpoints. These can then be compared with the observed image, varying the synthesised viewpoint until a match is found. A logical extension to this idea would be to use two observed images, from an observed sequence, to synthesise an interpolated image which is matched against a single stored reference image. For matching, Ullman discusses the use of a variety of possible features including points, edges, blob centres and contours.

The approach proposed in this thesis differs significantly from all of the systems described above. All the tracking algorithms so far mentioned, rely on the extraction of high contrast features from the image, most typically edge detection. The aim of this thesis is to tackle image sequences in which visibility conditions are so bad that edge detection based methods are not feasible. Rather than detecting edges to which a model is then fitted, the EM/E-MRF algorithm segments the image pixel by pixel using a probabilistic MRF based approach aided by prediction. The model is then fitted directly to the segmented image. No edge or other conventional feature detection methods are used. Ullman [1996] also briefly discusses directly matching between two images. The approach consists of evaluating the differences in grey-level between corresponding pixels in the two aligned images, whereas the EM/E-MRF algorithm finds the camera position for which a projected segmented image best fits the segmented observed image.

Some alternative approaches to tracking also deserve mention. It is possible to track a moving object simply on the basis of a moving coherent region in an image sequence. Such tracking methods are not applicable for the purposes of this thesis because they do not provide 3D information on the position of the camera relative to the object being tracked. Isard and Blake [1996] and [1998] report the use of the Condensation (Conditional Density Estimation) algorithm to track continually deforming curved boundaries of various moving objects against a cluttered background. The Condensation algorithm tracks a discretely sampled probability distribution of various alternative hypotheses from image to image. The tracking approach relies on being able to extract high contrast edges around the object being tracked in each image. Even though the algorithms are tested on objects against a cluttered background, object edges are clearly visible (though the clutter provides additional spurious edges). Such tracking systems are not appropriate for the extreme case considered in this thesis where visibility is so poor that object boundaries are often not defined by extractable edges. The tracking system is also inappropriate in that it is purely 2D and does not yield the 3D position and orientation of the camera.

Zisserman et al. [1999], Fitzgibbon et al. [1998] and Torr et al. [1999] report methods for extracting the 3D camera trajectory and camera Intrinsic parameters from image sequences. The advantage of the method is that neither a pre-calibrated camera nor a special calibration target or object are necessary. The method relies on matching a large number of high resolution features (corners and lines) between successive images with the constraint that the scene is rigid. The method is not suitable for scenes in which such features are sparse. This work does not attempt to recognise or locate a particular object of interest in the image. The intended application is the introduction of fictitious objects into video sequences for the entertainment industry.

Most of the tracking systems discussed so far deal with rigid bodies, since that is the scope of the work reported in this thesis. However, work is also reported which deals with deformable or articulated objects, particularly the case of human bodies. Sometimes these are modelled as kinematic chains of linked rigid components. Hilton et al. [2000] presents a technique for automatically building recognisable, moving 3D models of individual people. A set of images of a person from different viewpoints is captured. A standard 3D "generic humanoid model" is then transformed to approximate the individual's shape and anatomical structure by fitting it to the captured images. Ioffe and Forsyth [1999] describe a method to find sparsely clad people in static images. People are modelled as an assembly of nine cylindrical segments. Deutscher et al. [2000] and [2001] address human motion capture, modelling the body as an articulated set of truncated cones. They compare the use of Kalman filtering, Condensation and "annealed particle filtering" for tracking an articulated body with up to 34 degrees of freedom.

2.2 Segmentation and Markov Random Fields

Segmentation is the process of partitioning an image into a set of non-intersecting regions, such that each region is homogeneous but the union of no two adjacent regions is homogeneous. This thesis is concerned with binary or bi-level segmentation which is equivalent to classifying all pixels of an image into two classes, namely “object” and “background”.

Segmentation is a fundamental low-level vision task and forms the first essential step in many complex vision systems. This thesis presents a model based tracking algorithm which involves firstly segmenting each image in a sequence and secondly fitting a model of the object being tracked to the segmented image in order to extract the camera position and orientation relative to that object. Clearly, in such systems, the quality of the final output will depend largely on the quality of the initial segmentation process.

Hundreds of segmentation techniques are present in the literature, but no existing method works well on all kinds of images and each kind of image or imaging situation will yield best results with a different technique. For a comprehensive review of many different kinds of segmentation techniques see Pal and Pal [1993]. Other reviews include Fu and Mui [1981] and Haralick and Shapiro [1985].

The scope of this thesis is limited to the case of monochrome grey-scale images only. Pal and Pal describe several categories of segmentation techniques for these images including grey-level thresholding, relaxation, Markov Random Field (MRF) approaches, neural networks, edge detection of region boundaries and methods based on fuzzy set theory.

Thresholding is a simple and popular technique for image segmentation. A grey-level value (the threshold) is chosen and pixels are classified according to whether they are brighter or dimmer than this level. If only one threshold is used for the entire image then it is called global thresholding whereas schemes that involve partitioning the image into several sub-regions with a separate threshold defined for each are known as local thresholding or adaptive thresholding. For general surveys of various thresholding techniques see Sahoo et al. [1988] and Kittler et al. [1984].

Kittler and Illingworth [1985] derive a minimum error threshold under the assumption that the grey-levels of both object and background pixels are normally distributed. The pixel intensities of the image are described by a histogram giving the frequency of occurrence of each grey-level in the image. This histogram is viewed as an estimate of the probability density function for pixel grey-levels. Kittler and Illingworth model this density function as a mixture of two separate distributions, for "object" and "background" pixels respectively. These distributions are modelled as normal distributions. Since the means and variances of the two components of this mixture are unknown, Kittler and Illingworth present a method of best fitting the mixture of two Gaussians to the original histogram. The point of intersection of the object and background distributions then provides an optimal threshold value.

The EM/E-MRF algorithm presented in this thesis uses thresholding as the first stage of the segmentation process. The initial segmented image produced by simple thresholding is then iteratively refined using an Extended-Markov Random Field (E-MRF) method. As in Kittler and Illingworth's work, the image histogram is modelled as a mixture of two normal distributions, one representing object pixels and the other representing background. This work is distinct for two reasons. Firstly, instead of best fitting the normal distributions to the image histogram (in the manner of Kittler), a predicted (and segmented) image is projected using a camera position estimate and an object model. This is used to predict which regions of the observed image represent object and which background. Means and variances can then be calculated over these predicted regions and these values are used as estimates of the means and variances of object and background normal distributions. Secondly, in

Kittler and Illingworth's work the two normal distributions (and hence the segmentation thresholding value) remain fixed, however in the EM/E-MRF algorithm these distributions are continually relearned, both over successive iterations of the algorithm on a single image and also from image to image over a sequence.

Adaptive thresholding techniques are described by Chow and Kaneko [1972] and extended by Nakagawa and Rosenfeld [1979]. Each image is divided up into regions. An optimal threshold is determined for each region and this is interpolated between regions in order to determine an individual threshold for each pixel. The EM/E-MRF algorithm, described in this thesis, does not make use of adaptive thresholding. Incorporating adaptive thresholding into the algorithm would provide a more general image model and might improve robustness. This is discussed in chapter 6 as a possible extension for future work.

A more complex segmentation technique (for a surveillance camera application) is described by Grimson et al. [1998]. A number of observed images from a fixed camera are used to build up statistical image data of an observed scene for every individual image pixel. This historical data is then compared against new images in order to track foreign objects which have recently moved into the field of view (e.g. people, cars etc). Image rgb-levels are modelled by Gaussian mixture models, with a separate model for each pixel. A subset of the most common Gaussians-those with the highest weightings-are assumed to represent "background". Any observed pixel value which does not lie within two standard deviations of at least one of the background distribution means, is classified as "object".

This method is adaptive and robust, however it is intended for a fixed camera observing a relatively static scene. In contrast, the EM/E-MRF algorithm creates new statistical image models for each new image, since a moving camera, moving and focussed beam light sources, and changing visibility conditions mean that historical image models may not be applicable to new images. Two valuable aspects of Grimson's method are that the image model distributions are multi-modal and can vary with position in the image. Similar ideas are explored as possible further work in section 6.5.1.

Since the 1970s, there has been increasing interest in the use of Markov Random Fields (MRFs) as models to aid in the restoration and segmentation of digital images. MRFs are particularly useful in the case of very noisy or degraded images (e.g. in poor visibility) since they can make up for deficiencies in observed information (fluctuations in intensity, colour, texture and shape in observed images) by adding a-priori knowledge to the image interpretation process in the form of models of spatial interaction between neighbouring pixels. Hence, the classification of a particular pixel is based, not only on the grey-level of that pixel, but also on the classification of neighbouring pixels. Simplistically, pixels are more likely to belong to the "object" class if their nearest neighbours are also members of the "object" class and similarly for background pixels. Landmark papers include Besag [1974], Besag [1986], and Geman and Geman [1984]. Historically, the mathematical concepts originate in the statistical mechanics and mathematics literature with Gibbs [1902], Markov [1906] (in Russian) and Ising [1925] (in German).

One key problem is that of determining the values of the probability distribution of classifications for each pixel based on those of its neighbourhood. Besag [1974] and [1986], Geman and Geman [1984] and Derin [1985] and [1986] all make use of Gibbs distributions for characterising MRFs. These distributions were first used by Ising [1925] (in the statistical mechanics literature) to model molecular interactions. For the purposes of image segmentation, Gibbs distributions offer a simple way to assign a numerical value to the probability of any particular pixel classification that is dependent on the classifications of other pixels in the neighbourhood.

Once a suitable neighbourhood size has been specified and a model (e.g. Gibbs distribution) has been assumed to enable the computation of probabilities, the optimum segmentation problem becomes that of classifying every pixel in the image such that the probability associated with no pixel can be increased by altering the classification of either that pixel or any of its neighbours. The space of all possible permutations of pixel classification is too large to be searched exhaustively. Several iterative algorithms have been suggested for the solution of this problem. These are surveyed and their performance compared in Dubes et al. [1990]. Two of the most popular methods are known as "simulated annealing" and "iterated conditional modes".

Simulated annealing (Geman and Geman [1984]) belongs to the class of stochastic relaxation algorithms. Simulated annealing is theoretically guaranteed to find a globally optimal labelling, however it is relatively computationally expensive and slow. Dubes et al. [1990] report simulated annealing as failing on some real problems due to computational burden.

The Iterated Conditional Modes (ICM) algorithm (Besag [1986]) is not guaranteed to find the probabilistically optimum set of pixel labels, being vulnerable to convergence on local maxima. It is, however, several orders of magnitude faster than simulated annealing and therefore much more suitable for real time applications. Interestingly, despite the theoretical sub-optimality, Dubes et al. [1990] find the ICM algorithm to be more robust than simulated annealing. They also note that the probabilistically optimal labelling solution does not always correspond to the "best" image segmentation. The ICM algorithm was chosen to solve the Extended Markov Random Field (E-MRF) problem described in this thesis.

The work of Bouthemy and Lalande [1988] and [1989] is especially relevant to this thesis. Bouthemy and Lalande are concerned with the interpretation of murky underwater image sequences for robot navigation. Crucially, they extend the notion of Markov dependency to include, not only contributions from a given pixel's neighbourhood in the observed image, but also a contribution from the corresponding pixel in the previous frame of the image sequence. Thus Markov dependency becomes both spatial and temporal. In the Extended-Markov Random Field (E-MRF) used in this thesis, Markov dependency is again extended but, here, any given pixel's neighbourhood includes the corresponding pixel from a *predicted* image based on a model of the object being tracked and an estimate of the current camera position based on a learned model of the camera trajectory.

Other approaches to segmentation include those based on neural networks and fuzzy set theory. Neural networks are massively connected networks of elementary processors, some of which are claimed to resemble information processing in biological neurones. Many kinds of network architecture have been reported in the neural network literature. Good introductory texts include Bishop [1995], Ripley [1996] and Tarassenko [1998]. An obvious question in the neural network approach is what to use as input features. In general a (non-trivial) network (e.g. perceptron) will have several inputs and so a grey-level alone is insufficient information for a network based system that is designed to provide a classification for any particular pixel in an image. Hall et al. [1992] use, for each pixel, the intensities from three different Magnetic Resonance (MRI) images, as the three feature inputs of a neural network which then outputs a classification for that pixel. Ghosh et al. [1991] employ a neural network to segment an image according to a Markov Random Field model. The inputs to the network are the pixel values of a local neighbourhood and the output is the optimal classification for the pixel corresponding to that neighbourhood. A proposed hardware implementation of the network offers the potential of a high speed solution to MRF approaches which are normally computationally expensive.

The impetus behind the introduction of fuzzy set theory was to provide a means of defining categories which are inherently imprecise. This is achieved by means of membership functions such that a particular object can be a member of multiple sets simultaneously but with varying degrees of membership of each. The notion of membership functions of fuzzy categories is readily applied to image segmentation which attempts to divide up an image into several homogeneous regions. A membership function can be associated with each region and pixels assigned according to their degree of membership. Keller and Carpenter [1990] apply a similar approach to produce fuzzy versions of three segmentation schemes, namely fuzzy clustering, fuzzy region growing and fuzzy relaxation. The performance of these schemes is then compared to that of their crisp (non-fuzzy) counterparts. Pal et al. [1980] and [1987] assign fuzzy brightness levels to each pixel. They then define an "image fuzziness" value, based on fuzzy measures of distance between the grey-level image and its nearest binary (two-tone) version. An optimum segmentation thresholding value is then determined so as to minimise the corresponding image fuzziness value.

2.3 Expectation Maximisation

The Expectation-Maximisation (EM) algorithm was first reported by Dempster et al. [1977] as an iterative solution to problems where the observations can be viewed as incomplete data. The EM algorithm has since become increasingly popular in the literature, more recent examples including Neal and Hinton [1993], Bishop [1995], Ripley [1996], Cootes and Taylor [1997], North and Blake [1997], Grimson et al. [2000].

The EM algorithm is often referred to in the context of gaussian mixture models however it has far wider application. It is a general iterative approach to problems involving a hidden or latent variable (Blake [2000]).

Neal and Hinton [1993] express the algorithm in terms of calculating an expected distribution (E-step) for unobserved variables (in our case pixel class) in terms of observations (in our case pixel grey-level) and a current estimate of parameters (in our case camera position). The Maximisation or M-step then re-estimates the parameters to be those with maximum likelihood. It can be shown that with each iteration the true likelihood improves or at least remains constant until a maximum is reached.

The EM algorithm has previously been used to solve complicated image segmentation problems. Grimson et al. [2000] and Wells et al. [1996] incorporate segmentation of medical images by MRF within an EM feedback loop whilst refining an estimate of certain parameters of the scanning equipment (gain field or bias field). In this case the E-step consists of calculating pixel class (unobserved variables) based on the observed variable (pixel intensity) and a current estimate of the bias field (the hidden parameter). The M-step consists of re-estimating the bias field based on the new estimate of pixel class. The EM/E-MRF algorithm, described in this thesis, also incorporates MRF segmentation within the EM algorithm whilst refining an estimate of a hidden parameter, in this case camera position.

2.4 Poor visibility

The vast majority of current vision systems are designed to perform in good visibility through a clear medium which is assumed not to interfere with the relationship between world and image. Hardly any reported vision systems are designed to cope with very poor visibility. Occasionally papers appear in the computer vision and robotics literature which deal with underwater scenarios and these often claim robustness in poor visibility. However, these invariably still rely on extracting conventional features (typically edges) using conventional techniques.

Several authors have discussed the effects of poor visibility on images and vision systems. Barun and Ivanov [1999] use the theory of radiative transfer to investigate the optical effects of turbid media such as aerosol atmosphere, sea or oceanic water. They address the visibility problems of driving in poor visibility, including such topics as the visibility of retro-reflective markers of heavy trucks, ultimate visibility range of a car driver in a foggy environment and how many anti-fog headlamps one should use on a car. The paper does not deal with computer vision in the sense of automated image analysis, segmentation or object recognition but rather is restricted to a physical analysis of the optical effects of turbid media.

Narasimhan and Nayer [2002] investigate ways of actually making use of poor visibility effects in order to recover three-dimensional structure of a scene. In haze or fog, the visibility, colour and brightness of objects will diminish with distance from the observer. Hence, using one or two images taken under poor weather conditions it is possible to determine range information about objects in the scene. The work also investigates the chromatic effects of atmospheric scattering. Algorithms are developed for computing fog or haze colour, extracting depth information and recovering “clear day” scene colours. The work does not deal with feature extraction or object recognition under these conditions. The nature of the image degradation addressed does not eliminate image features (e.g. edges). Rather, the image is dimmed and colours (but not structure) are distorted. The degradation is also uniform over each of the images.

The work of Watkins et al. [2000] describes a system that improves the vision of pilots on runways in fog. The system utilises hyperstereo vision (a binocular system with baseline separation wider than the human inter-ocular spacing). A camera and laser are fitted to each wing of the aircraft. Each camera is synchronised with the laser on the opposite wing. The cameras alternately capture images with illumination from the opposite laser. The backscatter radiation pattern has a decreasing gradient away from the side where the illumination source is located and by comparing the images from each camera it is possible to subtract the backscattered radiation pattern from each image. The cameras are fitted with narrow bandpass filters which only permit light of the same frequency as the lasers to be detected. This is useful for minimizing the effects of scatter from solar and other light sources. They also propose the use of special retro-reflectors on the runway. The reflections of laser radiation from the reflectors provide a-priori information which enables deblurring to be performed. The work is also relevant to this thesis in that a test image sequence was created by using a fog chamber to generate poor visibility conditions. The work does not consider computer vision algorithms for object recognition or tracking but is restricted to providing a human pilot with enhanced images using stereo goggles and a display.

Foresti [2001] describes a vision based system to enable an autonomous underwater vehicle to navigate by following pipelines on the sea bed. The work is relevant to this thesis in that it involves vision based navigation in an underwater environment and also in that a known 3D model of the environment is used to provide a-priori information to the system. Foresti claims robustness in poor visibility, however his notion of poor visibility is not the same as that of this thesis, where poor visibility is taken to mean that conventional features (e.g. edges) are not useable. Foresti's system relies on extracting edges from the observed image and then fitting a known model of the pipeline to the edges in order to recover the position and orientation of the camera. When poor visibility obscures the edges of the pipeline, Foresti suggests reliance on the on-board inertial motion sensors of the vehicle.

Rokita [1997], Kaneda [1991] and Nishita [1987] deal with the modelling of poor visibility conditions for computer graphics applications including flight and driving simulators. Various kinds of poor visibility are considered including visual effects caused by ground fog, haze, clouds and raindrops on a windscreen. This work is relevant in that it highlights the range and complexity of the degradation processes present in real image sequences thus confirming the importance of using real images over synthetically rendered images when testing computer vision algorithms.

2.5 Ground truth

In order to quantify the performance of vision based tracking algorithms, it is necessary to test the algorithms using appropriate image sequences for which ground-truth data is available. This ground-truth data can then be compared with the outputs of the vision algorithms, enabling the computation of errors. Ground-truth data might include the true positions and orientations of the camera (or a tracked object) at each frame, calibration data for the camera and models of the viewed scene.

It is easy to construct artificial image sequences with known ground-truth using computer graphics packages (e.g. "POV-ray for Windows", http://www.povray.org). However, although testing computer vision algorithms on synthetic scenes allows comparison of performance, it gives only a limited idea of how the algorithms will perform on real scenes. Artificial scenes, generated using computer graphics software, do not completely reproduce the detailed variation of objects, the multitude of complex lighting conditions and modes of image degradation encountered in the real world and the only true test of computer vision algorithms remains their performance on real data. To this end, a number of researchers have attempted to create real video sequences with pre-measured ground-truth.

Drummond and Cipolla [1999][2000] describe algorithms for tracking an object by fitting a CAD model to an observed image. They use a robotic "camera-in-hand" system (camera attached to a robot arm) to test the algorithms. The arm is set to manoeuvre the camera into a specified position and orientation relative to the object being observed. The process is repeated with the arm starting from a variety of different randomly selected positions. The final positions (which ideally should all be identical) are read from the robot controller. Differences in final position yield r.m.s. translational and rotational errors. The process is repeated with the object being observed rotated by 15 degrees each time. The final positions should ideally now lie on a circle. Deviation from a true circle is used to assess accuracy.

In both these experiments the differences in final position are measured *relative to each other*. The true ground-truth position of the camera relative to the object being observed is not known and so no objective assessment of the system's positional accuracy is possible. The r.m.s. differences quoted assess variation in end result (noise) but there is no way of knowing if these noisy perturbations are about the true camera location or whether there is some underlying structured deviation from the true position.

Wunsch and Hirzinger [1996] also describe an algorithm for registering a model to an object in an image. The algorithm yields the position and orientation of the object relative to the camera. Wunsch and Hirzinger describe an experiment to assess the accuracy of their algorithm in which a robot arm is used to position a camera in known positions relative to the object being viewed. They report known ground-truth camera positions as being accurate to 0.5mm and 1.0 degrees. It is not clear whether or not these positions were extracted from the robot controller and, if so, how the position of the camera optical centre was measured relative to the terminal link of the robot.

The work is significant in that an attempt has been made to capture images with known ground-truth camera positions. The work is limited in that only still images from fixed positions have been captured. In contrast, the work described in this thesis generated moving image sequences with ground-truth camera positions determined along entire trajectories.

Agapito, Hayman and Reid [2001] generate ground-truth image sequences using their "Yorick" stereo head/eye platform. Ground-truth data for the orientation and zoom of the camera at each frame is extracted from the motor encoders of the platform. The work is limited to providing motion with only two degrees of freedom (both rotational). Angles of elevation and pan are known but the translational position of the camera remains unknown. It is not clear how the orientation information extracted from the encoders is synchronised to match individual frames in the sequence. This may be an inherent functionality of the Yorick system.

Otte and Nagel [1994] and [1995] have created both real and synthetic image sequences with known ground-truth for the assessment of optical flow algorithms. The real sequences involved using a robot arm to "fly" a camera past simple scenes. Known velocities are generated by translating the camera at known speeds. The work is significant in that the authors actually measured the ground truth motion field for a real video sequence and have made the sequence and the motion field publicly available. Unfortunately, the focus of work on optic flow centres on extracting motion fields (related to velocities) rather than absolute positions. The work is also limited to the case of pure translation only. Although motion fields are measured, camera positions do not appear to have been measured.

McCane et al. [2001] present a benchmarking suite of image sequences for the purpose of evaluating optical flow algorithms. Their technique allows the measurement of ground-truth motion fields for sequences involving general motion of a (hand-held) camera about a scene. The work is limited in that the scene may only contain planar polyhedral objects in front of a planar background. Only the background may intersect the image edges and the polyhedra must be un-occluded. Furthermore, all visible faces of the polyhedra must have at least four vertices, and the set of visible polyhedra faces may not change over the image sequence. The technique involves the hand-labelling of matching features in every single image of the sequence. This laborious process prohibits the use of sequences longer than a few frames. Again, while motion fields are measured for the sequence, the absolute position and orientation of the camera remain unknown.

Gracias and Santos-Victor [2001] address the problem of estimating the 3D trajectory of an underwater autonomous vehicle from a set of images of the seabed taken by an onboard camera. They present algorithms for visual pose estimation using video mosaicing. They describe the use of an image sequence with available ground-truth in order to assess the performance of the algorithms and quantify error. A sequence of images of the sea bed, captured by a surface-driven ROV is used to generate a mosaic of the sea bed which is assumed to be planar. A trajectory and camera model are specified and new images corresponding to views from the camera on this trajectory are synthesized from the mosaic. This data set cannot truly be said to be a "real" image sequence. The images are synthesized albeit based on other images which are real. The object being viewed is, in this case, the sea bed. This is constrained to be perfectly planar. Images of solid objects viewed against a background are not available using this method.

The work of Watkins et al. [2000] is significant in that a test image sequence is created in bad visibility. Dry ice (solid CO_2) and liquid nitrogen are used to fill a chamber with fog. Images are filmed through the fog to simulate conditions experienced by a pilot landing a plane in bad visibility. Ground truth camera data and positions were not calculated.

2.6 Work that directly precedes this research

The research described in this thesis was originally intended as an extension of work begun in Fairweather [1997a], Fairweather et al. [1997b] and Hodgetts et al. [1999]. This work introduces the Extended-Markov Random Field (E-MRF) in which

Markov dependency is extended such that the local neighbourhood surrounding any particular pixel also includes a contribution from the corresponding pixel in a predicted image. The predicted image is projected using a 3D model of the object being tracked and an estimate of camera position based on a Kalman filtered model of the camera trajectory. The method is tested on a variety of degraded images and the performance of the E-MRF is demonstrated to be superior to that of a conventional MRF (Geman and Geman [1984], Besag [1986] and Dubes et al. [1990]) and also superior to a spatio-temporal version of the MRF (Bouthemy and Lalande [1988] and [1989]) when segmenting poor visibility, underwater images for which a model is available.

Fairweather presents a tracking algorithm for determining the position of a remote operated vehicle (ROV) relative to an observed underwater oil-rig-like structure. Each frame in the video sequence is first segmented using the E-MRF technique. The segmented image is then edge detected and straight lines are best fitted to the edges. These straight lines are assumed to correspond to the boundaries of cylinders from which the oil rig structure is composed. Range information is computed by comparing the diameter of cylinders in the observed image with the known diameter of cylinders in a 3D model of the oil rig. Orientation relative to the oil rig is determined by comparing the angle between cylinders in the observed image with the true angle known from the model.

Fairweather's algorithm is limited in several respects. Firstly, it is assumed that the camera is looking directly at a node (intersection of 3 cylinders). Besides these assumed conditions being inapplicable in most real scenarios, this assumption reduces the degrees of freedom of motion that can be accommodated from six down to four (namely range and orientation with respect to the node). Secondly, the highly task specific method of determining range and orientation means that the algorithm cannot be applied to any other kind of observed object other than oil rig-like nodal intersections of three cylinders.

Fairweather was primarily concerned with proof of principle of E-MRF segmentation technique and, to this end, certain steps in his vision system were performed by hand. Both predicted images (produced by CAD software) and observed images from a video sequence were "trimmed" by hand such that portions of each would overlap to produce a good match. Most importantly, in order to determine statistics of the observed image, including the means and variances of the "object" and "background" portions of the image, each image was first "hand segmented". This means, in effect, that the vision system could only function if it was already given its ideal output as one of its inputs.

Fairweather's vision system, as presented, seems to rely on accurate predicted images coupled with a large weighting in favour of predicted information. Many of the outputs appear to have received little influence from the actual observed image.

The EM/E-MRF vision algorithm presented in this thesis differs from Fairweather's system in a number of important respects. A novel contribution of this work is the use of a predicted image to compute image statistics. The predicted image is used to divide the observed image into an initial estimate of "object" and "background" regions. Now pixel intensity values can be summed over these regions to compute means and variances for each region and hence class conditional normal distributions. This is in contrast to Fairweather's use of "hand-segmentation" to create these statistics for each image.

Fairweather's system, of fitting straight lines to the edge detected segmented image in order to extract cylinders, was found difficult to implement and re-produce.

Instead, the EM/E-MRF system fits the model of the object being viewed directly to the segmented image by means of non-linear optimisation of the correlation between segmented image and an image predicted from the current camera position estimate. This results in a far more general vision algorithm which can be applied to the tracking of any object for which a model is available, providing that the object's structure is sufficiently complex to provide unique views from which each position can be determined.

The other major advance on Fairweather's work is the use of iterative feedback by means of an Expectation Maximisation style mutual refinement of unobserved data and parameters. This permits the system outputs of both camera position and pixel classification to be simultaneously optimised over several iterations for each frame in the sequence. In contrast, Fairweather's entire system would be equivalent to a single such refining iteration.

2.7 Camera calibration

The work described in this thesis involves camera calibration for two reasons. Firstly, camera calibration makes it possible to measure ground-truth camera positions for test image sequences (see chapter 4). Secondly, the EM/E-MRF vision system relies on accurate knowledge of intrinsic camera calibration parameters (focal length, principal point location, pixel aspect ratio) as well as an accurate radial lens distortion model. Cameras used to film test sequences therefore need to be calibrated before the vision algorithm can be tested on those sequences.

The camera calibration approach used is based largely on that of Zhang [1998]. Many other camera calibration techniques have been proposed, both within the photogrammetry community (see for example Brown [1971]) and more recently in the computer vision literature (see for example Faugeras [1986a], Tsai [1987], Weng [1992], Maybank [1992], Faugeras [1992], Wei [1993]).

Photogrammetric calibration techniques rely on capturing images of known calibration objects or targets. These typically consist of three exactly orthogonal planes containing sets of features (e.g. corners of a grid of squares) or a single plane which undergoes a precisely known translation (e.g. placed on a bench for which the height can be precisely varied). These methods produce accurate and reliable calibration but require expensive or elaborate equipment.

Brown [1971] uses a series of plumb lines. By assessing the deviation from straight of the images of the lines, a lens distortion function can be found which varies with object distance. Weng [1992] uses the corners of a grid of black squares as his calibration features. The calibration target is mounted on a stand that can be raised or lowered in precise increments with a micrometric screw. Tsai [1987] uses a similar arrangement to calibrate video cameras and model radial lens distortion.

In contrast, "self-calibration" techniques, developed in the computer vision community, do not require any special calibration objects or targets. Instead they make use of geometrical constraints provided by corresponding points detected between multiple images of a static scene. This approach is very flexible but is much less robust, reliable and accurate than the photogrammetric methods.

Faugeras [1986a] suggests finding the camera parameters and epipolar geometry of a stereo pair of cameras using, firstly, a known set of 3D co-ordinates for a set of reference points in the images and, secondly, without a known set of 3D co-ordinates but just using pairs of matched correspondence points between images. Faugeras [1992] describes a system for calibrating a single camera which is moving. The system relies only on point matches between different images from the sequence. It is not necessary to know the motion of the camera. Maybank [1992] uses the epipolar transformations between several images from a single camera in different positions to provide constraints on the camera calibration parameters. Wei [1993] notes that it is possible to compute parameters defining image projection without explicitly finding the physical parameters of the camera itself. He calls these "intermediate" parameters and the process is known as "implicit" calibration.

The technique used in this work, based on the theory of Zhang [1998], lies somewhere between the two extremes of photogrammetry and self-calibration. As with self-calibration techniques, the camera captures calibration information from its observed environment whilst moving along its trajectory. In this case, however, the observed environment has a known structure since calibration targets have been placed in view of the camera throughout its motion. Unlike some of the photogrammetry apparatus, these targets are simple and inexpensive, being printed on a conventional laser printer and mounted on MDF fibre board (Zhang reports accurate results using only a book cover as his planar surface).

Once the intrinsic camera parameters are calculated (from a few views of a calibration target), the camera position and orientation can be computed at every frame in an image sequence, provided that at least one calibration target is in view at any time. This allows the complete camera trajectory for the image sequence to be re-constructed.

3 The EM / E-MRF algorithm

3.1 Overview of this chapter

This chapter explains in detail the vision algorithm which is the focus of this thesis. The Extended-Markov Random Field segmentation technique is described and it is shown how this can be incorporated into an Expectation Maximisation (EM) iterative feedback scheme. It is shown how this scheme is arrived at intuitively from the fundamental requirements of the vision system and the algorithm is also justified mathematically by expressing it formally in terms of probability theory.

Sections 3.9 and 3.10 explain some practical details including the parameterisation chosen to encode camera positions and orientations, how these poses are interpolated and extrapolated (in order to combine two pose hypotheses or predict a future pose), how the objects being viewed are measured and modelled and how these models are used to project predicted images.

3.2 Requirements of the vision system

The fundamental purpose of a vision based navigation (or tracking) system is to estimate the position and orientation of a camera (relative to some observed object) at regular intervals in time. Additionally (possibly as a by-product of locating the camera) it is useful for the system to interpret which part (2D) of the image represents the object being observed (i.e. segmentation). The system takes, as inputs, the grey level values of each pixel from the current image in a video sequence. The corresponding camera co-ordinates are output for each frame in addition to a segmented version of the observed image. This simple description is conveniently illustrated in figure 3.1.

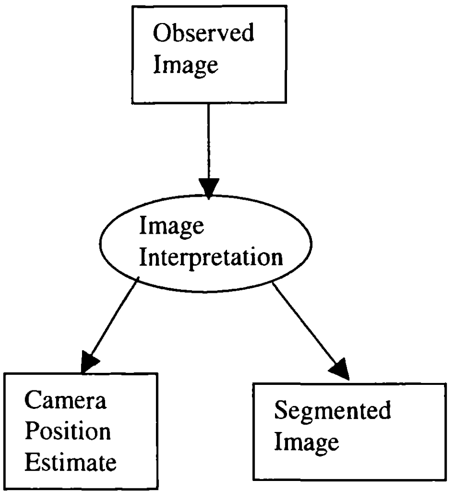


Figure 3.1 **Fundamental requirements of the vision system**

This thesis is particularly concerned with situations in which extremely poor visibility conditions are encountered. Hence it can be assumed that the information (grey-level pixel values) inputs to the system are severely degraded. Conventional approaches based on extracting features (e.g. edges) purely from the image grey-levels are unsuitable for this level of noise (see section 1.2) and it is therefore necessary to utilise some additional information. Since the recent history of the camera trajectory is known, it should be possible to estimate the camera position at the frame in question. The additional information contained in this position estimate might then be exploited by the vision system, aiding in the interpretation of the observed image. This system is illustrated in figure 3.2. One way of exploiting the estimated camera position information, in addition to a known camera model and a known model of the object being viewed, is to project a predicted image which can then be compared with the observed image.

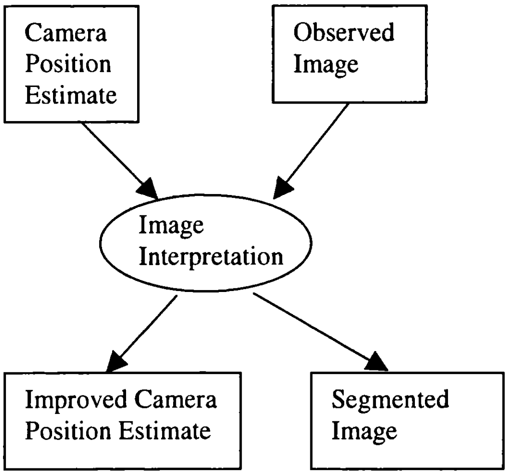


Figure 3.2 Utilising prior knowledge of the camera trajectory

Since an estimated position is now a system input, and an improved estimate of position is also a system output, an obvious feedback scheme suggests itself. The improved position estimate (system output) can now be fed back into the input of the system and this process can be iterated, hopefully converging on an optimal solution. This iterative process is illustrated in figure 3.3.

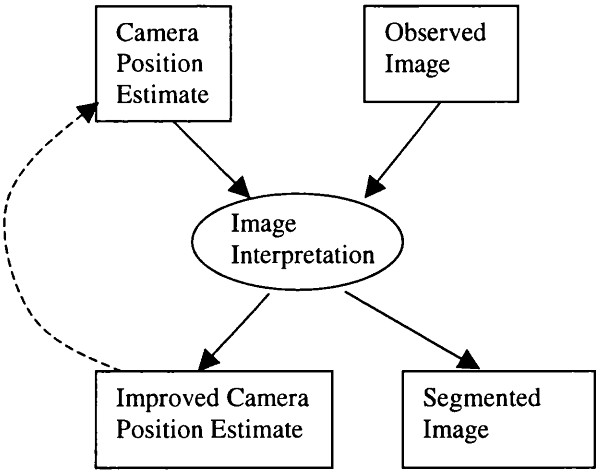


Figure 3.3 Iterative feedback scheme

The next section (3.3) describes an iterative procedure known as the Expectation Maximisation (EM) algorithm. It will become apparent that, under certain conditions, the iterative process of figure 3.3 can be seen as an example of the EM algorithm.

3.3 Expectation Maximisation

The Expectation Maximisation (EM) algorithm is often presented in the context of optimising Gaussian mixture models (e.g. Neal and Hinton [1993], Bishop [1995], Cootes and Taylor [1997]). It is, however, a very general iterative scheme (Blake [2000]) for solving problems which involve a set of observed variables, a set of "unobserved", "hidden" or "latent" variables, and a set of parameters which may be either probabilistically or explicitly coupled to both the unobserved and observed variables. In order to remain consistent with analysis undertaken later in this chapter (where $\underline{I}$ is used to represent pixel $\underline{I}$ntensities, $\underline{C}$ represents the set of pixel $\underline{C}$lass labels and $\underline{\theta}$ represents camera position and orientation co-ordinates), let the vector of observed variables be $\underline{I}$, the vector of unobserved variables be $\underline{C}$ and the vector of coupled parameters be $\underline{\theta}$. During the n^{th} iteration of the EM algorithm, the current estimate of $\underline{\theta}$ is written as $\hat{\underline{\theta}}^n$.

The EM algorithm consists of two steps, which are iterated alternately until convergence. The "E-step" (expectation) consists of computing the joint probability distribution for the observed and unobserved variables ($\underline{I}$ and $\underline{C}$) that is *expected*, given the observed variables, $\underline{I}$, and the current estimate of the parameters, $\underline{\theta}$. In other words, compute:

$$\mathcal{E}\left[p(\underline{I},\underline{C}\,|\,\underline{\theta})\,\middle|\,\underline{I},\hat{\underline{\theta}}^n\right]$$ **Equation 3.1**

The "M-step" (maximisation) consists of re-estimating $\underline{\theta}$ to be that which *maximises* this expected probability. In other words, a new estimate of $\underline{\theta}$ (which now becomes $\underline{\hat{\theta}}^{n+1}$) is chosen to be that which would be most likely to result in the currently observed values of the variables $\underline{I}$ and the inferred values of the hidden variables $\underline{C}$. Often a "log-likelihood" function is used (Dempster et al. [1977]) which is equal to the logarithm of the expected probability expression of equation 3.1. Taking the logarithm of a probability term can be a convenient way of simplifying expressions, since these terms are often exponential (e.g. normal distribution) and the logarithm function is monotonic. In this case $\underline{\hat{\theta}}^{n+1}$ is chosen to maximise:

$$\mathcal{E}\left[\log_e\left\{p\left(\underline{I},\underline{C}\,|\,\underline{\hat{\theta}}^{n+1}\right)\right\}\Big|\,\underline{I},\underline{\hat{\theta}}^{n}\right] \qquad \textbf{Equation 3.2}$$

The new estimate $\underline{\hat{\theta}}^{n+1}$ is now fed back into the E-step and the process is iterated until convergence. It can be shown (Neal and Hinton [1993]) that each such iteration improves the true likelihood, or leaves it unchanged if a local maximum has already been reached. The resulting iterative scheme is illustrated by figure 3.4.

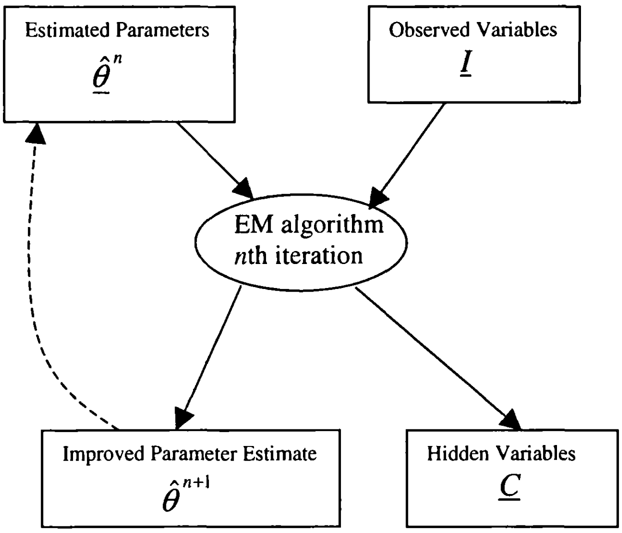


Figure 3.4 An illustration of the EM algorithm

The reason for this section's digression into EM theory will now be apparent, since the EM algorithm flow chart of figure 3.4 shares obvious similarities with the visual tracking system flow chart of figure 3.3. Under certain conditions, as follows, the vision scheme does indeed become a paradigm for the EM algorithm.

It is possible to regard the set of pixel grey levels from an observed image as corresponding to the "observed set of variables" of the EM inputs. Likewise, the set of "parameters" of the EM scheme are taken to be the camera position and orientation co-ordinates of the vision system. The true values of the "unobserved variables" are taken to be the ideal set of class labels (classes being either "object" or "background") for pixels in the image. Thus, inferring the values of the set of unobserved variables is equivalent to the process of image segmentation.

In order for the two schemes to become truly equivalent, the "image interpretation" process of figure 3.3 should involve choosing a new estimate of the camera position parameters which maximises an expected log-likelihood function, the expected form of which is based on both the observed pixel grey level values and also the current estimate of the camera co-ordinates. The following section describes how a modified notion of Markov dependency, the Extended-Markov Random Field (E-MRF), can be used to produce just such a log-likelihood function. Not only does the E-MRF enable estimated camera positional information to be included in the image interpretation process, but the use of spatial Markov processes as an image model provides robustness against severe noise (poor visibility being the theme of this thesis). It will be shown that the resulting EM architecture becomes equivalent to an intuitive iterative scheme whereby, firstly, observed images are segmented (via E-MRF utilising predicted camera co-ordinates) and, secondly, an improved estimate of the camera parameters is extracted from the segmented image, these two stages being iterated until convergence.

3.4 Markov Random Fields

It will be seen in later sections of this chapter that, in order to evaluate the expression of equation 3.2 or, alternatively, to determine a probabilistically optimal segmentation for an image deriving from Bayes' law, it will be necessary to determine values for the prior probability $P(\underline{C})$, where $\underline{C}$ denotes a particular arrangement of class labels for the pixels of an image. For the purposes of the images that are considered in this thesis, class labels can be either "object" or "background".

Markov Random Field image models enable $P(\underline{C})$ to be evaluated by assuming a spatial dependency between the classes of neighbouring pixels. They are particularly useful for interpreting very noisy or degraded images (e.g. in poor visibility), since they can make up for deficiencies in observed information (fluctuations in intensity, texture and shape in observed images) by adding a-priori information to the image interpretation process in the form of models of spatial interaction between neighbouring pixels. Simplistically, a pixel is more likely to belong to a particular class if its neighbours also belong to that class.

A random field is a collection of random variables arranged on a lattice (Zhang et al. [2000]). A digital image can be considered as a random field. In particular this thesis is concerned with binary, segmented images in which pixels can take either of two discrete values, namely "object" or "background". In principle the random field can be characterised by its probability distribution and the optimum classification for any particular pixel will be that which maximises the overall probability for the image. In other words, when segmenting an image containing N

pixels, for the i^{th} pixel we seek a class label, C_i, which maximises the joint probability:

$$P(C_i) \equiv P(C_1, C_2, \ldots . C_i \ldots .. C_N) \qquad \textbf{Equation 3.3}$$

Unfortunately, this implies that such a probability distribution must explicitly characterise the joint statistics of every pixel. In a binary image, this would consist of 2^N permutations with N being the total number of pixels in the image. This is an impossibly massive space to search, every time a pixel needs to be classified.

This combinatorial explosion is avoided by treating the image as a *Markov* Random Field (MRF), the fundamental notion associated with Markovianity being that of conditional independence (Zhang et al. [2000]). Conditional independence means that the probability distribution that describes a particular element of the random field can be de-coupled from the values of the other elements in the field beyond some local neighbourhood. For a simple, one dimensional example, consider the (temporal) Markov chain in which each variable (element) represents the weather on a particular day. In this case, de-coupling might mean that the probability of rain tomorrow is related to whether or not it is raining today, but is not related to whether or not it rained yesterday or on days prior to yesterday. This concept is readily extended to the two dimensional case of a digital image. It is now possible to de-couple the classification of a particular pixel from the classifications of other pixels in the image, instead restricting the probability of classification to being related only to the classifications of the pixels in some small neighbourhood local to the pixel in question. For the pixel at image location (i, j):

$$P(C_{i,j}) \equiv P(C_{i,j}, C_{i+m, j+n_{(m,n \in k)}}) \qquad \textbf{Equation 3.4}$$

where *k* denotes a small local neighbourhood around the pixel (*i*, *j*). Here, the neighbourhood is considered to include the eight pixels which immediately border the pixel in question (see figure 3.5).

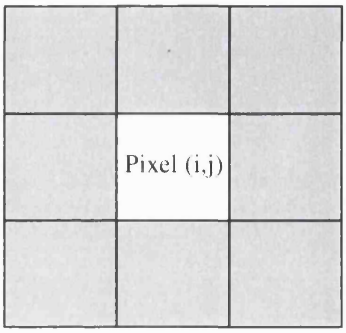


Figure 3.5 Conventional Markovian neighbourhood

In order to evaluate this expression for specific permutations of neighbourhood class labels, the Markov Random Field is characterised by a Gibbs distribution of the form:

$$P\left(C_{i,j}\right) = \frac{e^{-U_{i,j}}}{Z} \qquad \textbf{Equation 3.5}$$

where *Z* is included as a normalising constant to prevent equation 3.5 returning probabilities greater than one. The exponential part of this equation is defined as:

$$U_{i,j} = \sum_{m,n\in k} J(C_{i,j}, C_{i+m,j+n}) \qquad \textbf{Equation 3.6}$$

where *J* is a function defined as:

$$J(a,b) = \begin{cases} -1 & \text{if } a = b \\ 0 & \text{if } a \neq b \end{cases} \qquad \textbf{Equation 3.7}$$

Equations 3.4 to 3.7 describe a conventional MRF image model in which pixel class labels are considered to be spatially dependent. However, in accordance with equation 3.2, it is desirable to incorporate prior knowledge into the image model by making use of the estimated camera co-ordinates $\hat{\underline{\theta}}^{n}$. This is achieved by means of a known model of the object being viewed and a known model of the camera intrinsic parameters. Using the estimated camera co-ordinates, $\hat{\underline{\theta}}^{n}$, a predicted binary (segmented) image of the object being viewed is created by projecting the object model through the camera model. Markov dependency is now extended so that the Markovian neighbourhood includes, not only the nearest neighbour pixels to the pixel being classified, but also the corresponding pixel in the predicted image (see figure 3.6).

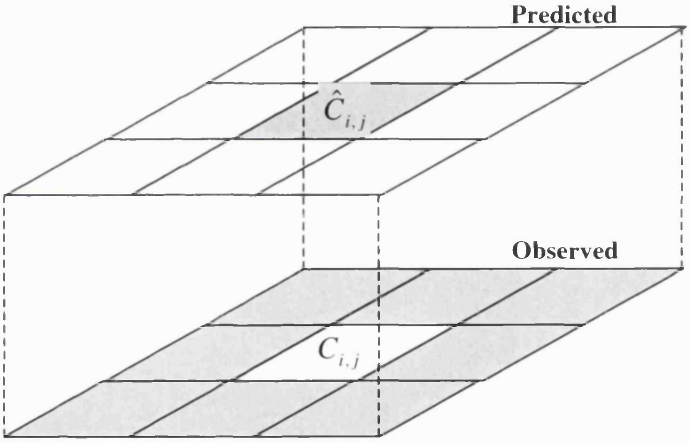


Figure 3.6 Extended-Markovian neighbourhood

Now:

$$P(C_{i,j}) \equiv P(C_{i,j}, C_{i+m,j+n_{(m,n\in k)}}, \hat{C}_{i,j}) = \frac{e^{-U_{i,j}}}{Z} \qquad \textbf{Equation 3.8}$$

where $\hat{C}_{i,j}$ denotes the *predicted* class label of the pixel (*i*, *j*), i.e. the value of the corresponding pixel in the predicted image. The exponential part of the Gibbs distribution now consists of weighted components:

$$U_{i,j} = \sum_{m,n\in k} S_1[J(C_{i,j}, C_{i+m,j+n})] + S_2[J(C_{i,j}, \hat{C}_{i,j})] \qquad \textbf{Equation 3.9}$$

where S_1 and S_2 are weighting constants which adjust the relative significance of information derived from the observed image versus information derived from the predicted image.

Thus, the Extended-Markov Random Field model provides a convenient means of determining the prior probability distribution for any particular pixel class label. In the following sections this model will be exploited in order to evaluate the log-likelihood function of the Expectation Maximisation algorithm.

3.5 E-step

From equation 3.2, the E-step consists in determining the *expected* value of the logarithm of the joint distribution $p\left(\underline{I},\underline{C}\,|\,\underline{\hat{\theta}}^{n+1}\right)$ *given* the observed set of pixel grey levels, $\underline{I}$, and the current estimate of camera parameters $\underline{\hat{\theta}}^{n}$. Maximising this likelihood function can be seen as mutually optimising, over the entire image, the corresponding likelihood function for individual pixels:

$$\mathcal{E}\left[\log_e\left\{p\left(I_{i,j},C_{i,j}\,|\,\underline{\hat{\theta}}^{n+1}\right)\right\}|\,\underline{I},\underline{\hat{\theta}}^{n}\right] \qquad \textbf{Equation 3.10}$$

This is the expected value of $\log_e\{p(I_{i,j}\cap C_{i,j})\}$, given $\underline{\hat{\theta}}^{n+1}$ where:

$$p(I_{i,j}\cap C_{i,j}) = p(C_{i,j})\times p(I_{i,j}\,|\,C_{i,j}) \qquad \textbf{Equation 3.11}$$

The prior probability $p(C_{i,j})$ can be evaluated by making use of the Gibbs distribution of equation 3.9. It is not necessary to consider the entire set of pixel classes, $\underline{C}$, because of the assumption of Markovian conditional independence.

The class conditional distributions, $p(I_{i,j} \mid C_{i,j})$, are estimated using a novel technique, developed during this work, which also makes use of prior knowledge and prediction. A conventional approach would be to estimate these distributions offline, based on values averaged over some training set of images for which the "true" class labels are known. This may not be appropriate if, for example, lighting conditions change radically over the image sequence as might be expected in an underwater environment where the light source is mounted on a moving vehicle. Different models may be necessary for different images. The approach taken here is to allow the vision system to re-learn new class-conditional models for each image frame and during each EM iteration by making the approximation:

$$p(I_{i,j} \mid C_{i,j}) \approx p(I_{i,j} \mid \hat{C}_{i,j}) \qquad \textbf{Equation 3.12}$$

where $\hat{C}_{i,j}$ denotes the *predicted* class label of the pixel (*i*, *j*). In other words, the predicted image (found by projecting the object model based on estimated camera co-ordinates $\underline{\hat{\theta}}^n$) is used to define provisional (predicted) class labels, $\underline{\hat{C}}$, for the observed image, from which class conditional grey-level histograms, means and variances can be computed. The validity of this approximation is obviously dependent on how closely $\underline{\hat{\theta}}^n$ approximates the true camera co-ordinates $\underline{\theta}$. (For examples of the system failing due to overly poor approximations, $\underline{\hat{\theta}}$, see sections 5.33 and 5.35).

The class conditional distributions are next approximated to Normal distributions. This approximation is justifiable in that the true class conditional histograms are often uni-modal and bell shaped (see figure 5.3, section 5.2.2). However, future work (see section 6.5.1) will propose ways of modelling both multi-modality in the distributions and also variation of the distributions with position in the image. The Gaussian model is particularly useful since it is of exponential form. The prior probabilities (equation 3.8) are also of exponential form and so it is easy to arrive at the log-likelihood function required by equation 3.10. The overall likelihood for a particular classification of a particular pixel is:

$$p(C_{i,j}) \times p(I_{i,j} \mid C_{i,j}) = \frac{e^{-U_{i,j}}}{Z} \times \frac{1}{\sigma_{c_{i,j}}\sqrt{2\pi}} \exp\left\{-\left(I_{i,j} - \mu_{c_{i,j}}\right)^2 / 2\sigma^2_{c_{i,j}}\right\}$$

Equation 3.13

Where $\sigma^2_{c_{i,j}}$ and $\mu_{c_{i,j}}$ are the variance and mean of the class conditional distribution of pixel intensities that corresponds to the choice of $C_{i,j}$ that is currently being considered for pixel (*i*, *j*). This results in the *negative* log-likelihood function:

$$\sum_{m,n \in k} S_1[J(C_{i,j}, C_{i+m,j+n})] + S_2[J(C_{i,j}, \hat{C}_{i,j})] + \tfrac{1}{2}\log_e\left(\sigma^2_{c_{i,j}}\right) + \frac{\left(I_{i,j} - \mu_{c_{i,j}}\right)^2}{2\sigma^2_{c_{i,j}}}$$

Equation 3.14

Note that certain constants, including the Z of equation 3.5, can be ignored since it is only necessary to compare the *relative* likelihoods of alternative pixel classification choices.

There is no obvious way of choosing values for the constants S_1 and S_2. In chapter 5, results will be demonstrated using different values in different visibility conditions. In good visibility, it is desirable to rely on observed information while taking comparatively little notice of error prone predictions derived from extrapolating the previous trajectory. Hence S_1 will be large and S_2 comparatively small. Conversely, given the absence of observed information in bad visibility conditions, it is necessary to make greater use of predicted information. In this case, much larger values of S_2 must be used. Further work (see chapter 6) may investigate methods by which these values can be automatically adjusted in response to varying visibility conditions.

3.6 M-step

The purpose of the M-step is to choose a new estimate of camera position, $\underline{\hat{\theta}}^{n+1}$, which maximises the overall log-likelihood of the image. This is equivalent to jointly *minimising* the expression of equation 3.14 simultaneously over all pixels in the image.

The optimal set of class labels, $\underline{C}$, should represent the binary image of the object being viewed, formed by projecting the object model through a camera placed at the true camera co-ordinates $\underline{\theta}$. The optimal choice of values for $\underline{\hat{\theta}}^{n+1}$ should thus be geometrically coupled with the optimal choice of class labels, $\underline{C}^n$, determined during each iteration. Choosing new values for $\underline{\hat{\theta}}^{n+1}$ with maximum likelihood, can thus be achieved by, firstly, choosing values of $\underline{C}^n$ which maximise the likelihood function (this corresponds to optimally segmenting the image) and, secondly, choosing values of $\underline{\hat{\theta}}^{n+1}$ in order to best fit the segmented image.

The space of all possible image interpretations contains many variables since it is necessary to consider all possible class label permutations over all pixels in the image. It is therefore not possible to search this space (of size 2^N where N is the number of pixels in the image) exhaustively in order to locate its global minimum (minimum *negative* likelihood). Various methods for finding mimima in MRF problems were discussed in section 2.2. Dubes et al. [1990] find the Iterated Conditional Modes (ICM) method, proposed by Besag [1986] to be both faster and more robust than Simulated Annealing (SA), proposed by Geman and Geman [1984], even though the ICM method is not guaranteed to find a global minimum. The ICM method was used during the research described in this thesis and is summarised as follows:

1) Initialise values for $\underline{C}^n$ by choosing class labels that maximise the class conditional distributions, $p(I \mid C) \approx p(I \mid \hat{C})$, for each pixel.

2) For each pixel (i, j) in the image:

   Update the class label, $C_{i,j}$, to be that which minimises the negative log-likelihood function of equation 3.14. (Note that this operation was performed on each pixel, line by line, as opposed to randomly choosing pixels to update).

3) Iterate step 2) until there is no further change of pixel class labels.

The second stage of the M-step involves finding the set of camera co-ordinates, $\hat{\underline{\theta}}^{n+1}$, which best fits the set of class labels, $\underline{C}^n$, that were determined by the ICM algorithm. The fitting is done by optimising the correlation between the ICM class labels, $\underline{C}^n$, and those predicted by projecting a predicted image using the current estimate of the camera co-ordinates $\hat{\underline{\theta}}^{n+1}$. Denoting the set of projected class labels as $\hat{\underline{C}}^{proj}$ gives a correlation based "goodness of fit" function:

$$\frac{\sum_{all\_i,j} \left(\mu_{C^n} - C^n_{i,j}\right)\left(\mu_{\hat{C}^{proj}} - \hat{C}^{proj}_{i,j}\right)}{\sqrt{\sum_{all\_i,j} \left(\mu_{C^n} - C^n_{i,j}\right)^2 \sum_{all\_i,j} \left(\mu_{\hat{C}^{proj}} - \hat{C}^{proj}_{i,j}\right)^2}}$$

Equation 3.15

where $C_{i,j} = \begin{cases} 1 \text{ if class is "object"} \\ 0 \text{ if class is "background"} \end{cases}$

and:

$$\mu_{C^n} = \frac{1}{N} \sum_{all\_i,j} C_{i,j}^n$$ **Equation 3.16**

$$\mu_{\hat{C}^{proj}} = \frac{1}{N} \sum_{all\_i,j} \hat{C}_{i,j}^{proj}$$ **Equation 3.17**

where N is the total number of pixels in the image.

The new estimate of camera position, $\underline{\hat{\theta}}^{n+1}$, is found by non-linear optimisation of the goodness of fit function over the six dimensional space of camera co-ordinates (three degrees of rotational freedom and three degrees of translational freedom). Many possible optimisation algorithms can be used. During this work, Powell's method was used for convenience, since it was also used in other aspects of the work (see chapter 4) and was readily available. This work has only been concerned with proof of principle and no attempt has yet been made to implement the EM/E-MRF algorithm in real time. Powell's method does take a long time to converge and it is possible (see section 6.5.1) that an alternative method might profitably sacrifice quality of fit for speed, especially since further refinement of position can be performed in successive EM iterations.

3.7 Equivalent Bayesian analysis

It is possible to arrive at the same likelihood function (equation 3.14) in a more intuitive fashion by treating the problem as one of segmentation according to maximum likelihood derived from Bayes' law. Given an observed image, $\underline{I}$, we wish to segment each pixel (i, j) by choosing a class label, $C_{i,j}$, which maximises the a-posteriori probability $p(C_{i,j} \mid I_{i,j})$. From Bayes' law, we have:

$$p(C_{i,j} \mid I_{i,j}) \propto p(I_{i,j} \mid C_{i,j}) \times p(C_{i,j})$$ **Equation 3.18**

As before, the prior probability, $p(C_{i,j})$, is modelled by the Extended-Markov Random Field and the class conditional probabilities, $p(I_{i,j} \mid C_{i,j})$, are predicted and continuously re-learned via equation 3.12. The Iterated Conditional Modes algorithm is then applied to choose class labels which maximise expression 3.18 over all pixels of the image. The model is then best fitted to the segmented image, yielding an improved estimate of the camera co-ordinates.

The EM algorithm is thus equivalent to an intuitive, two step process as shown in figure 3.7. An initial position estimate is used to help segment the image. The object model is then fitted to the segmented image to produce an improved position estimate. This is fed back into the segmentation process and the two stages are iterated until convergence. The algorithm is briefly summarised in the following section.

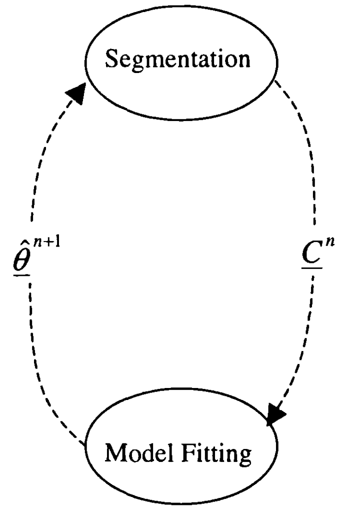


Figure 3.7 Equivalent intuitive two step process

3.8 Summary of the EM/E-MRF algorithm

This algorithm (figure 3.8) estimates the current camera position from the recent vehicle motion using a predictive filter. A predicted (and segmented) image is then generated by projecting a 3D model of the object being viewed onto an image plane at the estimated camera position. The predicted image is used to help interpret a relatively poor visibility observed image by means of an Extended-Markov Random Field (*E-MRF*) segmentation technique. The resulting segmented image is compared with the object model to provide a new estimate of the camera position. This improved position estimate can be fed back into the start of the algorithm resulting in an iterative scheme which has been shown to be a variant of the Expectation-Maximisation (*EM*) algorithm.

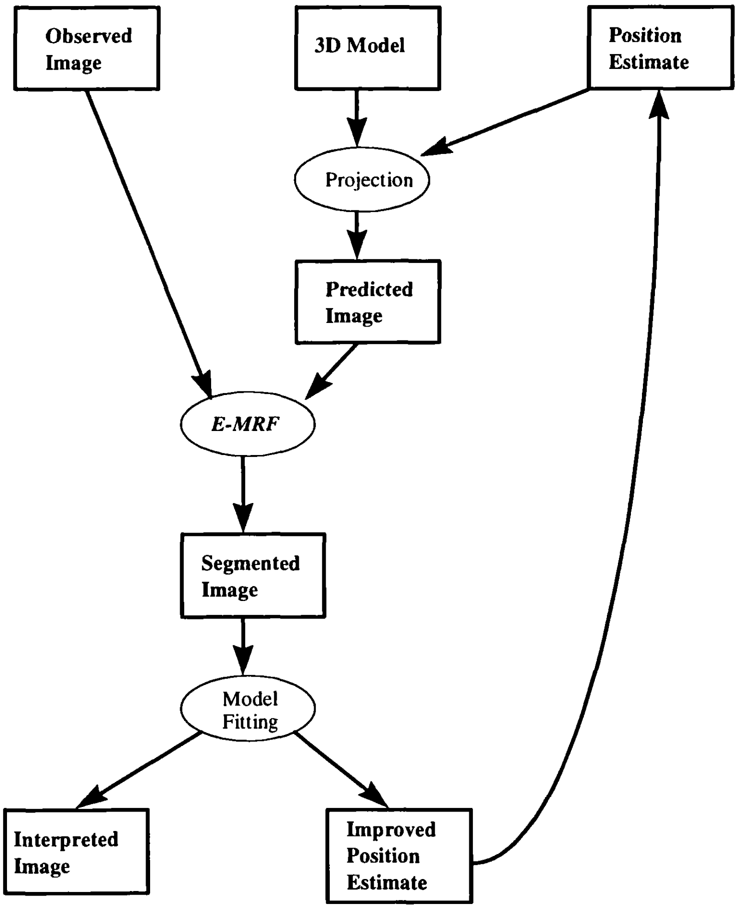


Figure 3.8 The EM/E-MRF algorithm

The algorithm combines predicted data with observed data in several important ways:

- A predicted image is used to estimate class conditional probability density functions.
- The predicted class of each pixel is introduced within an extended MRF model, enabling image segmentation to be both data and expectation driven.
- The estimate of camera position, as measured by the vision system, can be combined with the position predicted by extrapolating the recent trajectory of the camera.

3.9 Camera position parameterisation and prediction

3.9.1 Minimum parameterisation for rigid body rotations

The EM/E-MRF algorithm is designed to track the six degree of freedom motion of a camera. It is simple to parameterise the translational position of the camera, relative to the origin of a world co-ordinate system, in terms of three co-ordinates which represent the translation of the camera along each of the three orthogonal, cartesian *x*, *y*, *z* axes of the system. Unfortunately, it is not so simple to parameterise the rotations about these axes since they are not (kinematically) independent i.e. rotation about one Cartesian axis can be produced by combining rotations about the other two axes.

It is often convenient to describe rotation using a 3×3 rotation matrix. Since this matrix contains nine numbers, it provides excess degrees of freedom beyond the three required for rigid body rotation. Both in the EM/E-MRF algorithm (when best fitting the object model to the segmented image), and also during the calibration work described in the next chapter, it is necessary to perform non-linear optimisations on rotations, incrementing each rotational component by small amounts. It is desirable to avoid the complication of performing these optimisations under additional constraints (e.g. constraining a 3×3 matrix to remain a true rotation matrix while varying each of its elements) and so a minimum parameterisation is used which describes rotation uniquely using three numbers.

Rigid body transformations are defined by a vector of the form $(x, y, z, \omega_x, \omega_y, \omega_z)$ where (x, y, z) conventionally defines the translation component and $(\omega_x, \omega_y, \omega_z)$ is a vector whose direction defines an axis of rotation and whose magnitude defines the amount of rotation about that axis in radians. Since rotation matrices are still useful for operations such as projection of predicted images, it is necessary to be able to convert between the two notations.

It can be shown (see Paul [1981] and Watt [1992]) that, for a rotation of θ radians about an axis (of unit magnitude) $\underline{\hat{n}} = (n_1 i + n_2 j + n_3 k)$, the rotation matrix **R** is given by:

$$\mathbf{R} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} = \cos\theta \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} + (1-\cos\theta) \begin{pmatrix} n_1^2 & n_1 n_2 & n_1 n_3 \\ n_2 n_1 & n_2^2 & n_2 n_3 \\ n_3 n_1 & n_3 n_2 & n_3^2 \end{pmatrix} + \sin\theta \begin{pmatrix} 0 & -n_3 & n_2 \\ n_3 & 0 & -n_1 \\ -n_2 & n_1 & 0 \end{pmatrix}$$

Equation 3.19

In this case:

$$\theta = |\underline{\omega}| = \sqrt{\omega_x^2 + \omega_y^2 + \omega_z^2}$$ **Equation 3.20**

and $\underline{\hat{n}} = \underline{\hat{\omega}}$ **Equation 3.21**

where $\underline{\omega} = \omega_x i + \omega_y j + \omega_z k$ **Equation 3.22**

In order to retrieve the minimum parameterisation $(\omega_x, \omega_y, \omega_z)$ from the corresponding rotation matrix, **R**, the procedure is as follows. Summing the diagonal terms of the rotation matrix gives:

$$\cos\theta = \frac{1}{2}(r_{11} + r_{22} + r_{33} - 1) \qquad \textbf{Equation 3.23}$$

Differencing the off-diagonal terms:

$$r_{32} - r_{23} = 2n_1 \sin\theta \qquad \textbf{Equation 3.24}$$

$$r_{13} - r_{31} = 2n_2 \sin\theta \qquad \textbf{Equation 3.25}$$

$$r_{21} - r_{12} = 2n_3 \sin\theta \qquad \textbf{Equation 3.26}$$

Squaring and adding equations 3.24, 3.25 and 3.26 gives:

$$\sin\theta = \pm\frac{1}{2}\sqrt{(r_{32} - r_{23})^2 + (r_{13} - r_{31})^2 + (r_{21} - r_{12})^2} \qquad \textbf{Equation 3.27}$$

Taking the positive square root produces a positive value for $\sin\theta$, ensuring that$0 \le \theta \le \pi$. Note that this gives a unique direction to the axis of rotation. There are now two possibilities. If $\theta \le \frac{\pi}{2}$, then, from equations 3.24-3.26:

$$n_1 = \frac{r_{32} - r_{23}}{2\sin\theta} \qquad \textbf{Equation 3.28}$$

$$n_2 = \frac{r_{13} - r_{31}}{2\sin\theta} \qquad \textbf{Equation 3.29}$$

$$n_3 = \frac{r_{21} - r_{12}}{2\sin\theta} \qquad \textbf{Equation 3.30}$$

If, however, $\frac{\pi}{2} \le \theta \le \pi$, the diagonal terms of the rotation matrix are used:

$$r_{11} = n_1^2(1 - \cos\theta) + \cos\theta \qquad \textbf{Equation 3.31}$$

$$r_{22} = n_2^2(1 - \cos\theta) + \cos\theta \qquad \textbf{Equation 3.32}$$

$$r_{33} = n_3^2(1 - \cos\theta) + \cos\theta \qquad \textbf{Equation 3.33}$$

Giving

$$n_1^2 = \frac{r_{11} - \cos\theta}{1-\cos\theta}$$ **Equation 3.34**

$$n_2^2 = \frac{r_{22} - \cos\theta}{1-\cos\theta}$$ **Equation 3.35**

$$n_3^2 = \frac{r_{33} - \cos\theta}{1-\cos\theta}$$ **Equation 3.36**

One must take care, when square rooting the above expressions, to obtain the correct signs. From equations 3.24, since $\sin\theta$ is always taken as positive, ω_x must have the same sign as $(r_{32} - r_{23})$. This gives:

$$n_1 = \operatorname{sgn}(r_{32} - r_{23})\sqrt{\frac{r_{11} - \cos\theta}{1-\cos\theta}}$$ **Equation 3.37**

$$n_2 = \operatorname{sgn}(r_{13} - r_{31})\sqrt{\frac{r_{22} - \cos\theta}{1-\cos\theta}}$$ **Equation 3.38**

$$n_3 = \operatorname{sgn}(r_{21} - r_{12})\sqrt{\frac{r_{33} - \cos\theta}{1-\cos\theta}}$$ **Equation 3.39**

where $$\operatorname{sgn}(r_{mn} - r_{nm}) = \begin{cases} +1 & \text{if } (r_{mn} - r_{nm}) \geq 0 \\ -1 & \text{if } (r_{mn} - r_{nm}) < 0 \end{cases}$$ **Equation 3.40**

In practice, only the component of $\underline{\hat{n}}$ with the largest value is taken from equations 3.37-3.39. The other two components are then found by summing the off-diagonal terms of the rotation matrix:

$$r_{21} + r_{12} = 2n_1 n_2 (1-\cos\theta)$$ **Equation 3.41**

$$r_{32} + r_{23} = 2n_2 n_3 (1-\cos\theta)$$ **Equation 3.42**

$$r_{13} + r_{31} = 2n_3 n_1 (1-\cos\theta)$$ **Equation 3.43**

The vector $\underline{\omega}$ is easily obtained by scaling $\underline{\hat{n}}$ by the magnitude θ.

3.9.2 Quaternions for interpolating angular displacements

While tracking over a video sequence, two different estimates of camera position and orientation are available at each frame. One of these is the "initial estimate" based on extrapolating the camera trajectory from the previous frame position. The other is the output of the EM/E-MRF vision system. These can be regarded as two independent measurements (though arguably not completely independent) which should be combined or "averaged" according to some optimum weighting (depending on the level of confidence associated with each measurement) to give a position estimate which makes best use of both sources of information. A commonly used technique for combining information from two measurement sources during tracking is the Kalman filter (Welch and Bishop [2002], Kalman [1960]) in which the updated Kalman gain provides the probabilistically optimum weighting for combining two sources of information.

In conventional implementations of the Kalman filter, the state (in this case camera pose) is updated (extrapolated to predict the next position) by multiplying it by a matrix-the "system model". This is commonly written as:

$$x_k = \mathbf{A}x_{k-1} + w_{k-1} \qquad \textbf{Equation 3.44}$$

where x_k is the "state" (here position) and w is a noise model. For position tracking, the matrix **A** often contains a set of linear kinematic equations, typically constant acceleration models.

Unfortunately, when tracking a rigid body moving with six degrees of freedom of motion, there is no obvious choice for the matrix **A** because the three degrees of rotational freedom are not independent. In addition, proper implementation of the Kalman filter requires estimates of the variances associated with each of the two position estimates. For the application described in this thesis, these variances are difficult to measure or predict since they will vary with the visibility conditions encountered and with the different kinds of possible camera motion.

Instead, for interpolating between the observed and estimated camera orientations and for extrapolating the camera trajectory to predict new orientations, rotations were encoded in quaternion space (see Watt [1992]). Quaternions extend the concept of a complex number to include three imaginary units:

$$\text{quaternion } q = (s, \mathbf{v}) = s + v_x i + v_y j + v_z k \qquad \textbf{Equation 3.45}$$

(s stands for $s$calar component and $\mathbf{v}$ stands for $\mathbf{v}$ector component)

where $$i^2 = j^2 = k^2 = -1 \qquad \textbf{Equation 3.46}$$

$$ij = k \qquad \textbf{Equation 3.47}$$

$$ji = -k \qquad \textbf{Equation 3.48}$$

with the cyclic permutation:

$$i \to j \to k \to i \qquad \textbf{Equation 3.49}$$

Quaternions form a closed group under the multiplication operator defined as:

$$q_1 q_2 = (s_1, \mathbf{v}_1)(s_2, \mathbf{v}_2) = (s_1 s_2 - \mathbf{v}_1 . \mathbf{v}_2 ,\ \ s_1 \mathbf{v}_2 + s_2 \mathbf{v}_1 + \mathbf{v}_1 \times \mathbf{v}_2) \qquad \textbf{Equation 3.50}$$

Quaternions are useful for representing rotations since a subgroup of the quaternion group is closely related to the group of rotation matrices. It can be shown (Watt [1992]) that the act of rotating a vector **r** by an angular displacement θ about an axis **n**, is equivalent to performing the operation:

$$qp\bar{q} \qquad \textbf{Equation 3.51}$$

where q is a quaternion encoding the rotation:

$$q = \left(\cos\left(\frac{\theta}{2}\right),\ \sin\left(\frac{\theta}{2}\right) \mathbf{n} \right) \qquad \textbf{Equation 3.52}$$

and p is another quaternion representing the vector to be rotated:

$$p = (0, \mathbf{r})$$ **Equation 3.53**

and $\bar{q}$ is the conjugate of q defined such that:

$$q\bar{q} = s^2 + |\mathbf{v}|^2 = |q|^2$$ **Equation 3.54**

Rotations map onto quaternions of unit magnitude so that:

$$q\bar{q} = 1 \qquad \text{and} \qquad \bar{q} = q^{-1}$$ **Equation 3.55 and 3.56**

and the entire group of rotations maps onto the surface of a four dimensional hypersphere in quaternion space. Since any two angular displacements lie on this surface, the angular displacement that interpolates between them must also lie on this surface. In order to ensure a sensible, smooth interpolation between two angular displacements, it is necessary to employ spherical linear interpolation, moving along an arc of the geodesic that passes through the hyperspherical locations of the mappings into quaternion space of the two displacements. Figure 3.9 illustrates the case of interpolating between two angular displacements, represented by the quaternions q_1 and q_2. The interpolated quaternion is shown as q_{int} where:

$$q_1 \cdot q_2 = \cos\Omega$$ **Equation 3.57**

u represents the degree of interpolation between q_1 and q_2, i.e:

$$0 \le u \le 1$$ **Equation 3.58**

It can be shown (Watt [1992]) that the correct interpolation is given by:

$$q_{\text{int}} = q_1 \frac{\sin(1-u)\Omega}{\sin\Omega} + q_2 \frac{\sin\Omega u}{\sin\Omega}$$ **Equation 3.59**

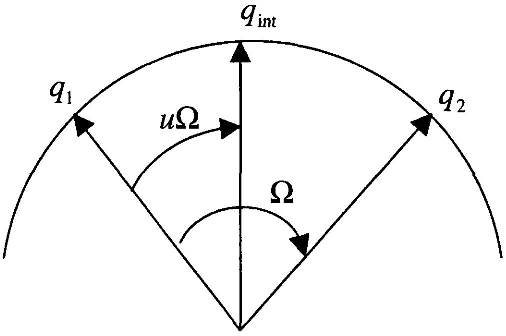


Figure 3.9 Spherical linear interpolation

When combining measured (by the EM/E-MRF vision system) and predicted camera pose estimates, a value of u is chosen that reflects the confidence associated with each source of data. In good visibility, the interpolation is weighted heavily in favour of the vision based measurement whereas in increasingly poor visibility u is chosen to weight increasingly in favour of the predicted camera orientation, based on trajectory extrapolation. In this respect, u acts very much like a Kalman gain.

In order to extrapolate a trajectory for predicting the camera orientation at a frame, the orientations of the two previous frames are assigned to the quaternions q_1 and q_2, and the interpolation factor u is set to the value of 2. This is equivalent to a constant velocity model which is a reasonable assumption given a slow camera motion and high frame rate. The translational components of the camera position are similarly predicted using a constant velocity model.

3.10 Measurement and modelling of viewed objects

In order to test the EM/E-MRF algorithm, real video sequences were filmed (see chapters 4 and 5) which contain various different objects, including a rectangular steel block and a scale model oil-rig-like structure. In order for the EM/E-MRF algorithm to project predicted images, it was necessary to build computer models of these objects. For a discussion of projective geometry, the camera model and camera calibration, lens distortion and the co-ordinate frames used see chapter 4.

The “oil-rig” object is composed of cylinders. Each of these cylinders can be defined by the co-ordinates of each end (ends of cylinder axis) and a radius. The spatial co-ordinates and radii of all the cylinders were measured on a co-ordinate measuring machine (CMM) and the co-ordinates were converted to those of the world co-ordinate frame (chosen as that of the base calibration target, see chapter 4).

For every pixel in each image to be predicted, the vector equation of a ray is found which passes through that pixel, originating at the optical centre of the camera. Each ray is then examined to determine whether or not it intersects any of the rig cylinders.

In order to determine whether or not a ray intersects a cylinder, the shortest distance between the ray and the cylinder axis is first determined. If this distance is greater than the cylinder radius, then intersection does not occur and the corresponding pixel is labelled as "background" (black). If this distance is less than the cylinder radius, then there are two scenarios in which intersection can occur. Firstly (figure 3.10), intersection occurs if the shortest line, connecting the ray to the cylinder axis, intersects the axis between the end points of the cylinder. Secondly intersection occurs if (see figure 3.11) the distance d is shorter than the distance L, where d is the distance between the end of the cylinder and the intersection between the cylinder axis and the shortest line joining the axis to the ray. L is the length of the projection onto the cylinder axis of the portion of the ray which connects the point of intersection of the ray with the cylinder surface to the point of closest approach between the ray and the cylinder axis.

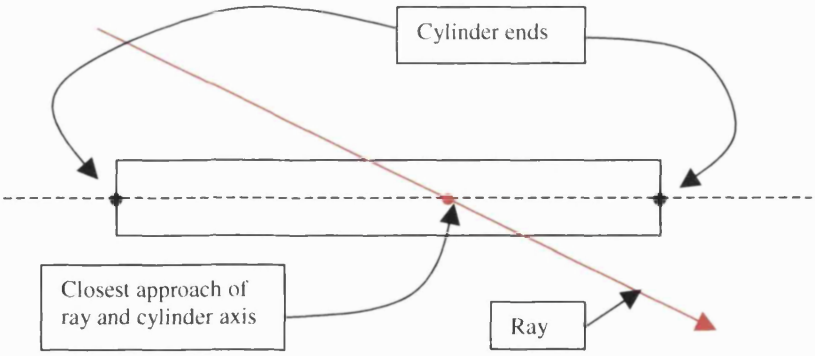


Figure 3.10 **Shortest distance between ray and cylinder axis occurs between cylinder end points. View shown is the projection on a plane parallel to both ray and cylinder axis directions.**

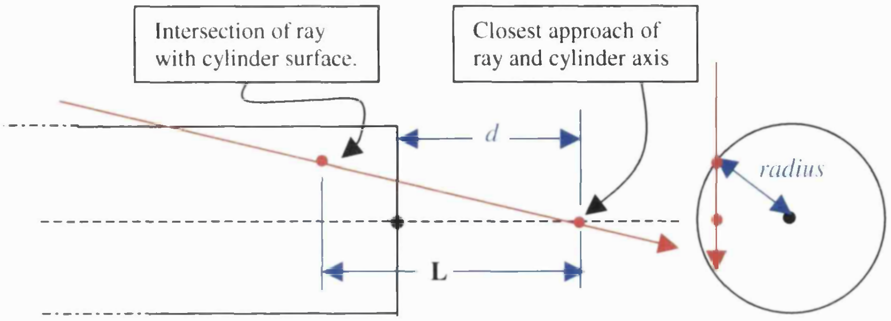

Intersection of ray with cylinder surface.
Closest approach of ray and cylinder axis
d
L
radius

Figure 3.11 Shortest distance between ray and cylinder axis occurs outside cylinder end points. Left diagram is a projection on a plane parallel to both ray and cylinder axis. Right diagram is an end view of the cylinder.

The "block" object was measured with callipers and the co-ordinates of each corner in the world co-ordinate system (see chapter 4) were computed and recorded. In order to project a predicted image of the block, each corner was projected onto the image plane by multiplying its co-ordinates by the camera projection matrix (see chapter 4). The three dimensional corner positions thus give rise to a set of two dimensional projected corners on the image plane. The shape of the projected block is now defined by the convex hull (smallest possible convex polygon) that encloses these points. The hull can be found using a "package wrapping" algorithm (other algorithms are computationally more efficient but unnecessarily so in this limited case of a six sided polygon).

In both cases (oil-rig and block), the projected image must now be radially distorted using the measured (during calibration) radial distortion parameters. Chapter 4 includes a detailed discussion of how the radial distortion parameters are determined and used.

4 Constructing a data set

4.1 Introduction

4.1.1 Purpose of this work

Having proposed algorithms to enable a robotic vehicle to navigate visually, it is necessary to construct appropriate video sequences with which to test and validate these algorithms. It is desirable to create test video sequences filmed along a pre-measured camera trajectory. This known ground-truth can then be compared to the outputs of the vision algorithms in order to quantify their performance.

The purpose of this experimental work is to produce a set of image sequences for which the camera position at every frame has been accurately measured. The image sequences must show a known object, which can be accurately modelled. They must be captured by a camera of known calibration parameters moving along a known trajectory and must be filmed in various conditions of limited visibility.

4.1.2 Why is this work necessary?

Chapter 3 describes the *EM/E-MRF* algorithm for vision based robotic navigation in conditions of poor visibility. Variant algorithms have also been proposed (see section 6.5).

During early work (Stolkin et al. [2000]), the algorithm was partially demonstrated using an image taken from a set obtained by Fairweather et al. (Fairweather [1997a], Fairweather et al. [1997b] and Hodgetts et al. [1999]). No calibration information was known for this image and so the *EM/E-MRF* algorithm was demonstrated crudely by extracting camera ranges as a ratio of the unknown focal length. The performance of the algorithm was assessed qualitatively in that successive iterations could be seen by eye to converge towards the true image interpretation (in terms of a predicted image superimposed over the observed image). It is common in the literature for tracking (e.g. Christmas [1996], Drummond [2000]), model registration (e.g. Lacey [2001], Wunsch [1996]), and segmentation (e.g. Kamber [1992], Wells [1996]) algorithms to be demonstrated and validated "visually" in an ad-hoc manner (i.e. illustrating a visual match between an observed image and a superimposed outline of the algorithm's interpretation of that image). Such tests are a simple and intuitive way to support the validity of novel algorithms, however they have several deficiencies. Problems posed by using un-calibrated image sequences and adhoc visual validation include:

- Ranges cannot be properly extracted (by matching predicted images to observed images) from an image up to a scale factor of focal length without knowing any other camera parameters. If the depth of the object being viewed is significant relative to the range of the object from the camera, then parts of the object that are close to the camera will appear enlarged relative to those that are distant from the camera. The severity of this distortion is also a function of the focal length of the camera; hence predicted images based on a unit focal length and a camera range estimated in "focal length units" will not properly correspond to the observed image even if the range estimate is accurate.

- The position of the camera cannot be properly extracted, making it impossible to model the trajectory of the camera. This means that the predictive filtering aspects of the algorithms cannot be tested, and that it is therefore impossible to test the algorithms on sequences of multiple images.

- It is not possible to quantify the performance of the algorithm without an image sequence for which the "ground-truth" of camera position has been accurately measured at each image for comparison with the outputs of the navigation algorithms.

It is therefore necessary to produce a set of test sequences with a properly calibrated camera moving along an accurately measured trajectory.

4.1.3 Why not use artificial image sequences?

A video sequence is needed with known ground-truth in the following forms:

- Intrinsic camera parameters.
- Lens distortion parameters.
- Camera position and orientation for every frame.
- Known object in the field of view which can be accurately modelled.

It is a relatively simple task to construct an artificial image sequence, which satisfies these requirements, using commonly available computer graphics software e.g. POV-Ray (http://www.povray.org). Furthermore, it would then be possible to generate varying degrees of poor visibility by artificially adding noise to the synthetic images.

In fact, the use of artificial images for testing vision algorithms is common in the literature (e.g. Smith [1997], Otte [1994], Harkness [2000], Mokhtarian [2000]). In general, vision and image processing algorithms seem to perform much better on these artificial (or artificially degraded) images than on real images of real objects filmed with a real camera (see Fairweather [1997a]). Real cameras and real visibility conditions result in many kinds of noise and image distortion. These real conditions are far more complicated than Gaussian noise or "salt and pepper" type speckling and it is not trivial or obvious how to realistically synthesise real world noise in an artificial image (Rokita [1997], Kaneda [1991]). Typically, real sources of image degradation (see figure 4.1) will include:

- Radial lens distortion (barrelling).
- Non-uniform lighting (e.g. ROV mounted spotlights, lighting intensity varies with position in image).
- Dynamic lighting (lights move with vehicle, lighting conditions vary with time).
- Camera saturation.
- Shadow.
- Occlusion.
- Attenuation.
- Back-scattering.
- Blur (both focal blur and motion blur).
- Reflection.
- Discrepancies between real objects and their models.
- The unknown and unplanned e.g. fish, seaweed etc.

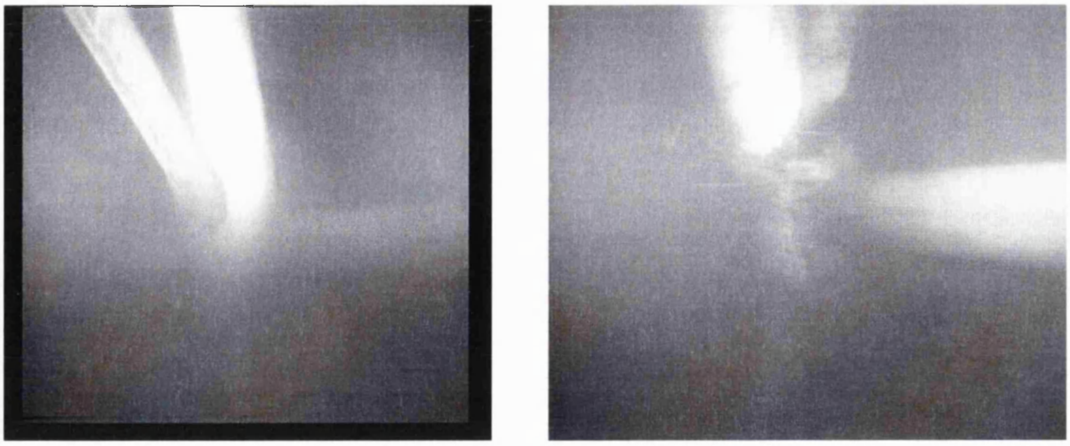


Figure 4.1 **Images of an oil-rig like structure filmed underwater. The only illumination comes from lights mounted on the ROV. Many kinds of image degradation are present.**

It is therefore not sufficient to test vision algorithms on artificial images subjected to simple degradation models, especially when it is claimed that these algorithms are suitable for real world pictures in conditions of extremely poor visibility.

4.1.4 Characteristics of the data set

A set of image sequences, exploring a range of conditions, has been produced. These conditions include:

- A range of 3 different objects of varying complexity.
  - A cuboidal block.
  - A cuboidal block plus hexagonal prism.
  - A scale model of an offshore rig type structure.

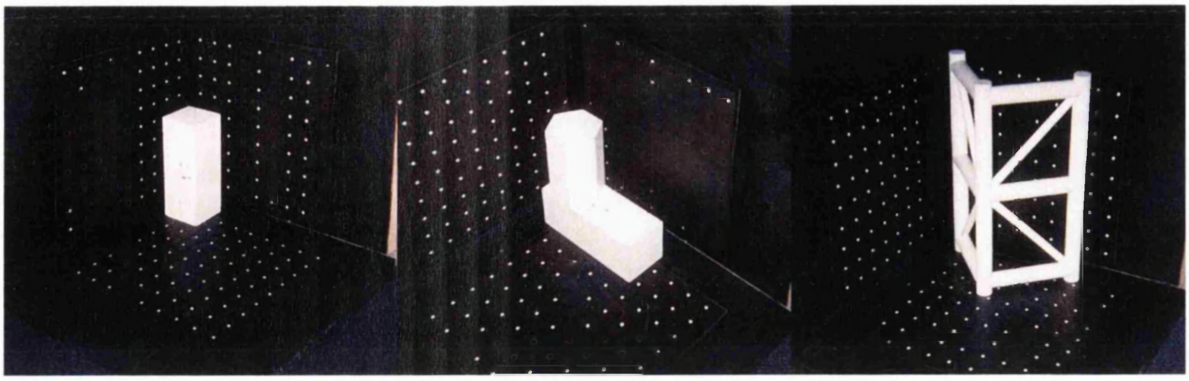


Figure 4.2 Photographs of the three objects to be filmed in the video sequences. Objects are shown in position within the calibration target system. Scale is 30mm between dots.

- A range of trajectories of different complexities:
    - Pure translation.
    - Pure rotation about an axis approximately through the camera (panning).
    - Movement of the camera in a planar, approximately circular motion about the object being viewed.
    - A six degree of freedom motion involving varying speeds and accelerations and sudden direction changes.
- A range of different visibility conditions varying from full visibility to zero visibility. Poor visibility was created using dry ice fog by pouring boiling water on solid CO_2 chips. This produces visibility conditions similar to the genuine underwater video footage collected by Fairweather [1997a].
- Different lighting conditions including:
    - Fixed lighting.
    - Dynamic lighting consisting of spotlights mounted on the camera that moved with the camera. This simulates the lighting conditions encountered by an underwater Remote Operated Vehicle.

Furthermore, for each video sequence filmed it was possible* to extract the following information:

- The “intrinsic” camera parameters were measured including:
  - Focal length.
  - Principal point location.
  - Pixel aspect ratio.
- Radial lens distortion parameters.
- The position and orientation of the camera (“extrinsic” parameters) for every frame in the sequence.
- An accurate computer model of the object being viewed.

*Note: although image sequences filmed along four trajectories of varying complexity were captured (along with additional calibration images) for each object, due to time constraints only the most complicated (the general six degree of freedom trajectory) of these has so far been fully calibrated and analysed. Computer models were created and tested for the block object and the oil-rig object, but so far not for the hexagonal prism object.

4.1.5 Structure of this chapter

Section 4.2 explains in detail how the data was captured, including physical details of the experimental set-up and construction. Section 4.3 explains how this data was analysed to produce calibrated image sequences. Section 4.4 presents the results of this work, including the trajectory extracted during analysis and a discussion of accuracy and sources of error. The calibration technique adopted here and much of the analysis in this chapter is adapted from Zhang [1998].

4.2 Data capture procedure

4.2.1 Summary of data capture procedure

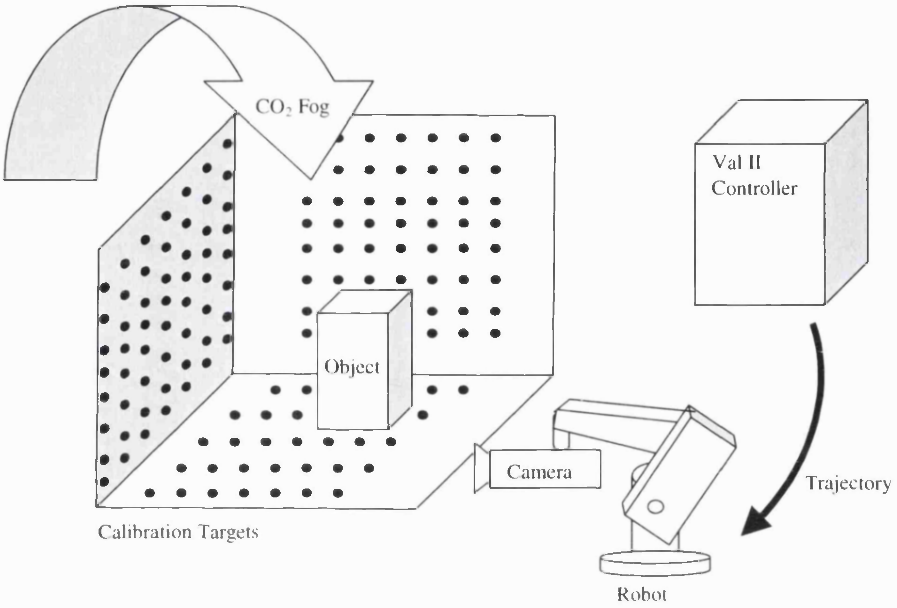


Figure 4.3 Equipment set-up for data capture.

An industrial six axis robot arm (PUMA 560) was used to move a digital cam-corder along a highly repeatable trajectory. “Calibration sequences” were filmed during the motion by placing a set of three calibration targets (square grids of dots) in the field of view. “Test sequences” (bad visibility image sequences) were filmed by:

- Concealing the calibration features on the targets.
- Introducing an object of interest at a known location relative to the targets.
- Introducing dry ice fog to create poor visibility conditions.
- Introducing variable lighting conditions.
- Moving the camera past this scene (object in limited visibility) along the same trajectory as for the calibration sequence.

“Target relations” image sets were also filmed. These involved positioning the camera in order to capture images clearly featuring all three targets together and also pairs of targets. These “target relations” images were used to compute the position and orientation of each target relative to the base target (one of the three targets lying in a horizontal plane, forming the base of the scene and used as a world co-ordinate frame for the scene) and also to provide information about intrinsic camera parameters and lens distortion.

Software was constructed to analyse the calibration sequence:

- Detect, locate, and label calibration features.
- Extract intrinsic camera parameters (focal length, principal point and pixel aspect ratio).
- Extract lens distortion parameters (two numbers defining radial distortion).
- Compute the position and orientation of targets relative to each other.
- Compute the position and orientation of the camera (extrinsic parameters) at every frame in the video sequence (with respect to a world co-ordinate frame defined to lie in the base target).

Positions and orientations for the camera at each frame in the calibration sequence were used to provide ground truth for the corresponding (synchronous) frames in the poor visibility test sequence.

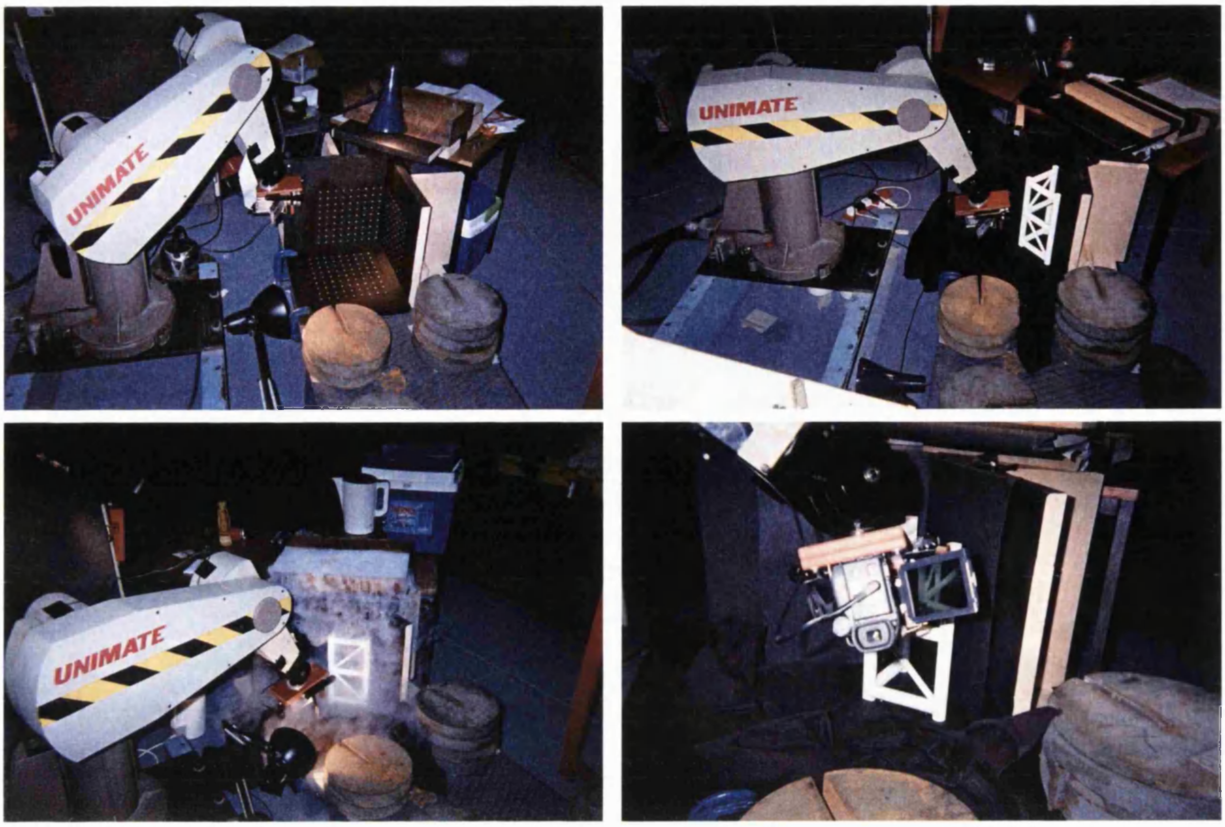


Figure 4.4 **Filming calibration sequence (top left), good visibility sequence (right) and poor visibility sequence (bottom left).**

4.2.2 Synchronising image sequences

The strategy, for the creation of video sequences with known ground truth, relies on extracting camera positions for each frame in a calibration sequence and then using these positions as ground truth for the corresponding frames in a test sequence (video sequence of interest). The success of this strategy depends on how well the two sequences can be synchronised.

For the purpose of synchronisation, an extra calibration feature (a white spot) was introduced to the scene (located in an extreme corner of one of the targets). The robot trajectories were programmed such that this extra feature was always visible at the beginning and end of each trajectory/video sequence. During poor visibility (with fogging) sequences, the dry ice fog was not introduced until after this extra feature had been clearly filmed. The synchronisation procedure was then as follows:

- Choose an image at beginning or end of poor visibility test sequence in which the extra "synchronisation spot" can be clearly observed.
- Superimpose successive frames from the calibration sequence (e.g. by image differencing) until an accurate match is found. Label these images as the calibration synchronisation image and the test sequence synchronisation image respectively.
- The camera position for the test sequence image, a certain number of frames away from the test sequence synchronisation image, is now taken to be the position extracted from the calibration image that is the same number of frames away from the calibration synchronisation image.

A detailed analysis of the synchronisation error is presented later (section 4.4.3). Most sequences could be synchronised to within ± 1 pixel when comparing synchronisation spots. At 25 frames per second, synchronisation in terms of temporal error should be at worst ± 0.02 seconds. If multiple test and calibration sequences are filmed, there is a high probability of finding an accurately matching test/calibration sequence pair.

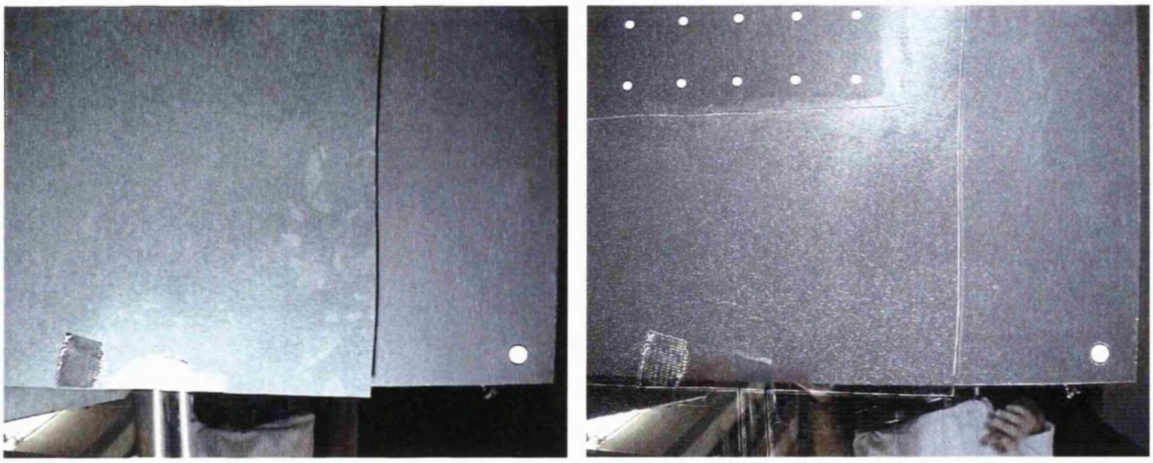


Figure 4.5 **The "synchronisation spot" shown at the beginning of a calibration sequence (right) and a test sequence (left).**

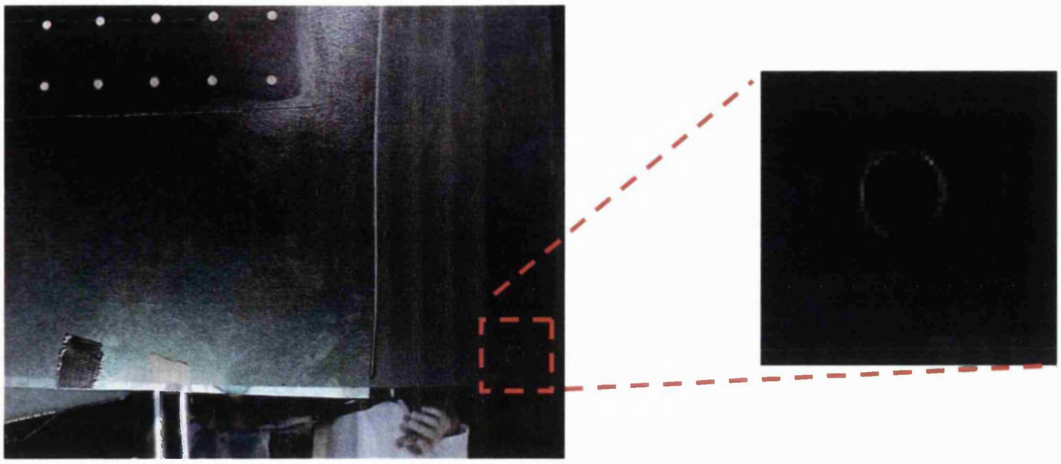


Figure 4.6 **Images from figure 4.5, superimposed using image differencing. The area of the calibration spot is black indicating a good (to within ± pixel) match between the two sequences.**

4.2.3 Why not extract positions from the robot control system?

There are several reasons why it is not practical to extract camera positions for each frame of an image sequence from the robot control system.

Firstly, industrial robots are highly *repeatable* but not *accurate*. Any position obtained from the control system would be significantly erroneous (Greig [1996]). Additionally, positions are needed that are measured relative to the object being observed (or a co-ordinate frame common to both object and camera) rather than from the robot's arbitrary co-ordinate system origin.

Furthermore, even if the robot controller could output a list of points, there would be no obvious way of matching these points to individual frames in the video sequence (i.e. synchronising the robot position measurements with information from the camera).

What is needed is the position of the camera (optical centre) which is *not* the same as the position of the robot terminal link. Camera calibration methods would therefore, in any case, have to be used to compute the position and orientation of the camera relative to the terminal link of the robot.

4.2.4 Calibration target strategy

The construction of the calibration targets is described in the following section (4.2.5). The purpose of the calibration targets is to provide a sufficient number of appropriate features in each frame to allow the computation of camera position and orientation for that frame. In addition, it should be possible to compute camera intrinsic parameters and lens distortion parameters from these features.

The calibration target structure must also provide a world co-ordinate system. It must be possible to accurately and repeatably locate the objects being observed at known world co-ordinates within this system.

During this work, three calibration targets were used. Each target consisted of a square, 9×9 grid of white circular spots on a matt black background. The three targets were arranged approximately orthogonally. An arrangement was chosen such that at least one target would be in the field of view of the camera throughout its motion during each video sequence.

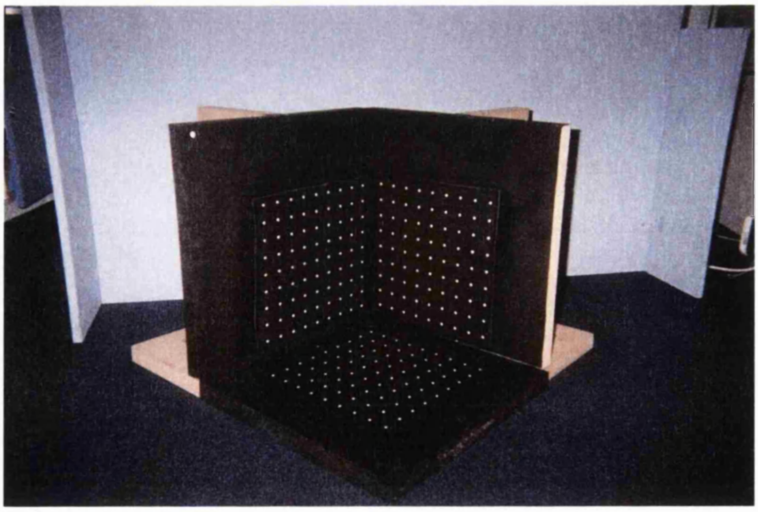


Figure 4.7 The three calibration targets.

For each target, each spot was labelled according to a pair of cartesian axes set in that target i.e. each target had spots ranging from (1,1) to (9,9). The world co-ordinate system was taken to be that of the spots in the base target. The units of the world co-ordinate system are thus "spot spaces" (of 30mm). Thus the calibration process produced camera positions and orientations relative to the co-ordinate system of the base target. Objects being viewed by the camera were located (see section 4.2.6) on the base target at known positions relative to this co-ordinate system.

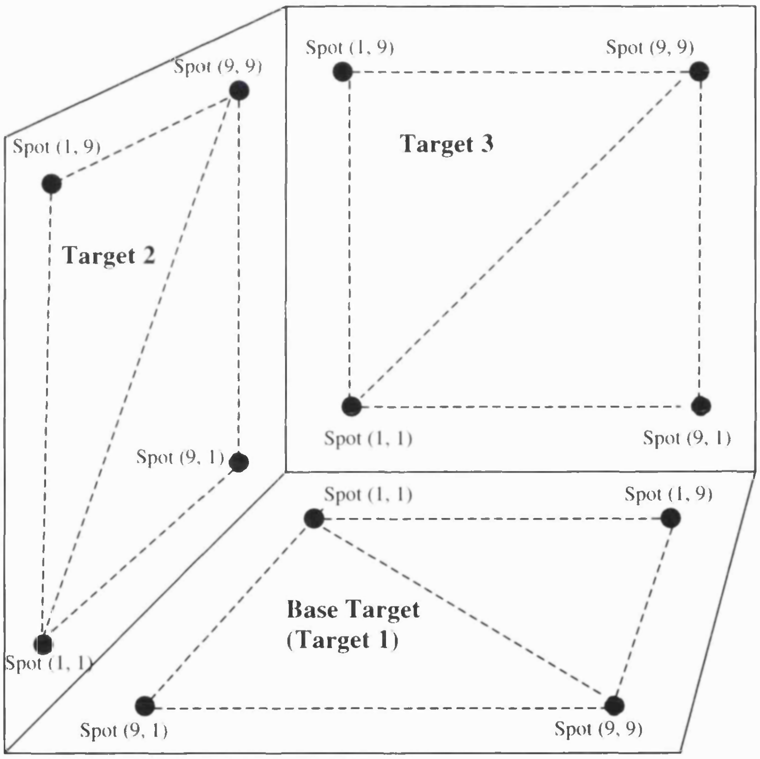


Figure 4.8 **Labelling system for spots in each target. The labels of each spot refer to that spot's position within a co-ordinate frame set in the target to which the spot belongs.**

It should be possible (Zhang [1998]) to extract the position of a pre-calibrated camera from a single view of a single calibration target. The reason that the arrangement of three targets was used is that, due to the complicated motion of the camera, the base target was not always in view. During the data analysis process (see section 4.3) it was possible to compute the position and orientation of each additional target relative to the base target. This meant that camera positions could be extracted from views of any target. These positions (relative to the target in view) were then combined with known relationships between each target to yield camera positions relative to the world co-ordinate system fixed in the base target.

4.2.5 Construction of calibration targets

- The target features were printed on thin card using a conventional office laser printer. Each target featured a square 9×9 grid of white circular spots of 4mm diameter and 30mm spacing, against a black background.

- The printed spot spacings were measured by hand to check for distortion in the printing process. A small distortion was noted in one direction of approximately 0.5mm over 8 spot spaces (240mm). This was considered too small to be significant. Larger errors of this kind could be easily corrected during the calibration process.

- After printing, each target was sprayed with a matt varnish in order to reduce reflection from the black background sections and make these sections appear more consistently dark in images.

- Each printed target was then spray mounted onto 30mm thick medium density fibre-board (*MDF*) to ensure that targets remained rigidly flat and planar.

- Any remaining visible surfaces of *MDF* were painted matt black.

- Two targets were fitted with *MDF* bases to make them stand vertically and one target was left to lie flat, forming a base to the scene being viewed (see figures 4.7 and 4.8).

- The targets were clamped in position on a steel deck.

4.2.6 Locating objects in the scene

The calibration strategy relies on locating the camera (for every frame in the video sequence) relative to a co-ordinate system attached to the base target. Clearly, this is only meaningful if the objects to be viewed can be accurately and repeatably located at a known position and orientation with respect to this base target co-ordinate system.

The objects being filmed were constructed such that they would sit stably on a flat surface under their own weight. The problem of locating the objects thus became two dimensional. A simple jig was incorporated into the base target to ensure precise, repeatable location of objects within the scene.

Two straight steel strips were bonded to the base target using a cyano-acrylic adhesive. Objects were then repeatably located in unique positions and orientations by butting them up against the straight edges (see figure 4.9). The edges were attached at known distances from the grids of calibration spots (see figure 4.10).

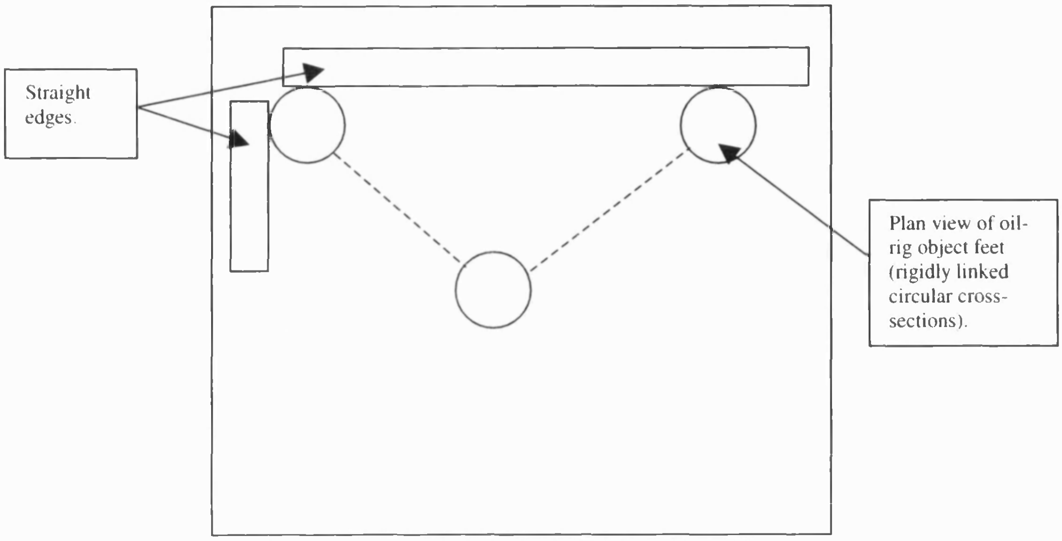

Straight edges.
Plan view of oil-rig object feet (rigidly linked circular cross-sections).

Figure 4.9 **Plan view of the base target, showing how two straight edges can uniquely locate the three circular feet of the oil-rig object.**

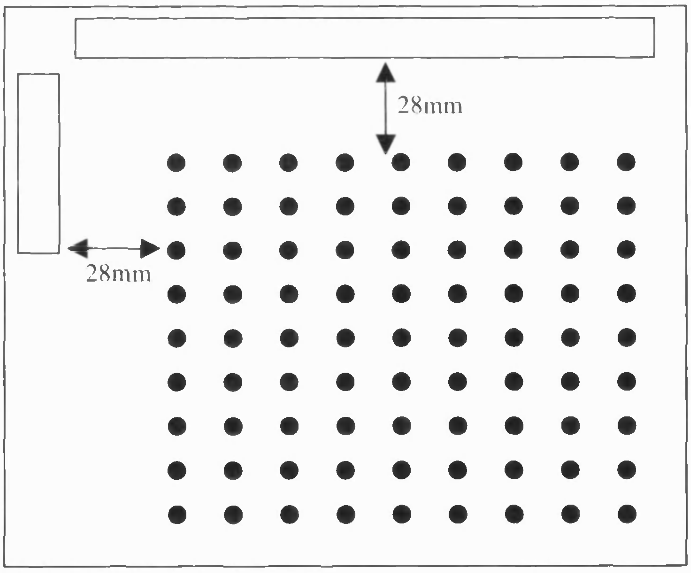


Figure 4.10 **Plan view of base target showing steel strips and their position relative to the grid of calibration spots. The 28mm gap is measured from the edges of the spots (not the centre).**

4.2.7 Attaching the camera to the robot

A rig (see figure 4.11) was built that enabled the cam-corder to be rigidly clamped to the terminal link of the PUMA robot. The major components were manufactured from machined Tuffnel since this material is light, strong and rigid. The rig system consisted of two plates. The upper plate was bolted to the robot and the lower plate was attached to the camera using both a circular clamp around the camera barrel and also a U shaped clamp that fitted around the body of the camera. The upper and lower plates were then bolted together securing the camera rigidly to the robot. An overhanging lip afforded a degree of protection to the camera in instances of collision.

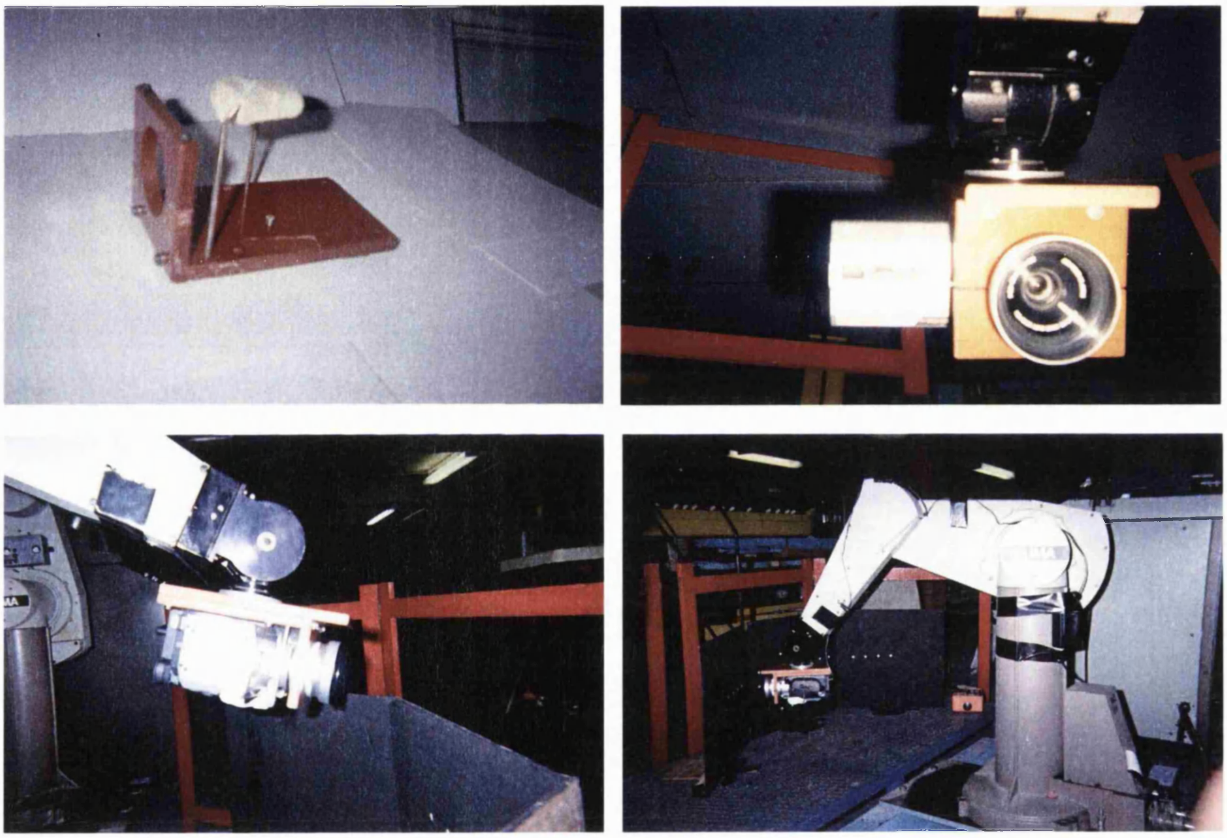


Figure 4.11 Robot-camera attachment system.

4.2.8 Generating variable visibility conditions

During filming, varying degrees of poor visibility were created using dry-ice fog. A metal trough was positioned above the scene, out of view of the camera. Variable quantities of solid CO_2 ("dry-ice") chips were deposited in the trough. During the filming of each poor visibility video sequence, boiling water was continuously poured onto the dry ice chips at varying rates. A dense vapour was formed which steadily drifted down onto the scene during filming (figure 4.12). Using this technique it was possible to create video sequences with varying degrees of poor visibility, ranging from clear visibility to virtually zero visibility. The poor visibility sequences exhibit elements of image degradation similar to those observed in genuine underwater conditions. Both kinds of image (see figure 4.13) exhibit attenuation, back-scatter, blurring, occlusion, camera saturation, shadow, non-uniform lighting and lens distortion. The images also appear visually similar.

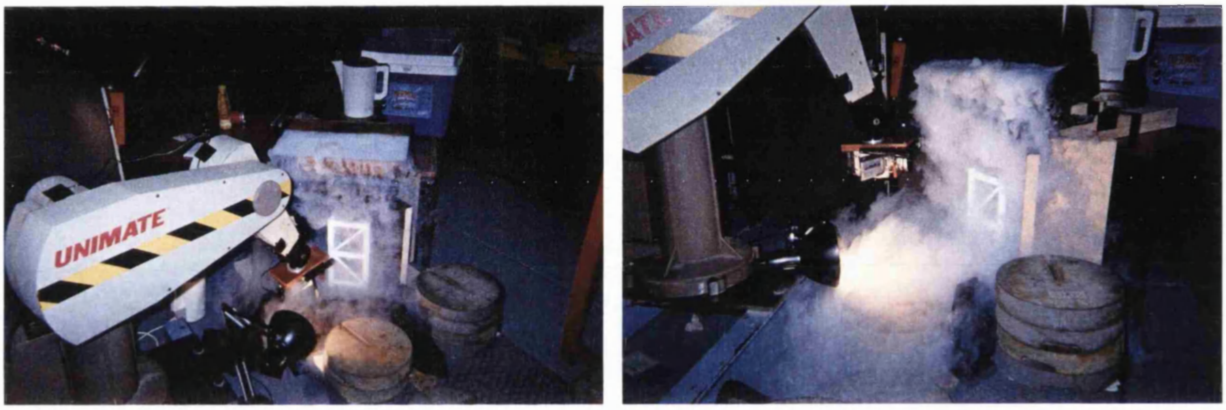


Figure 4.12 **Generating poor visibility using a suspended trough, solid CO_2 "dry ice", and a kettle of boiling water.**

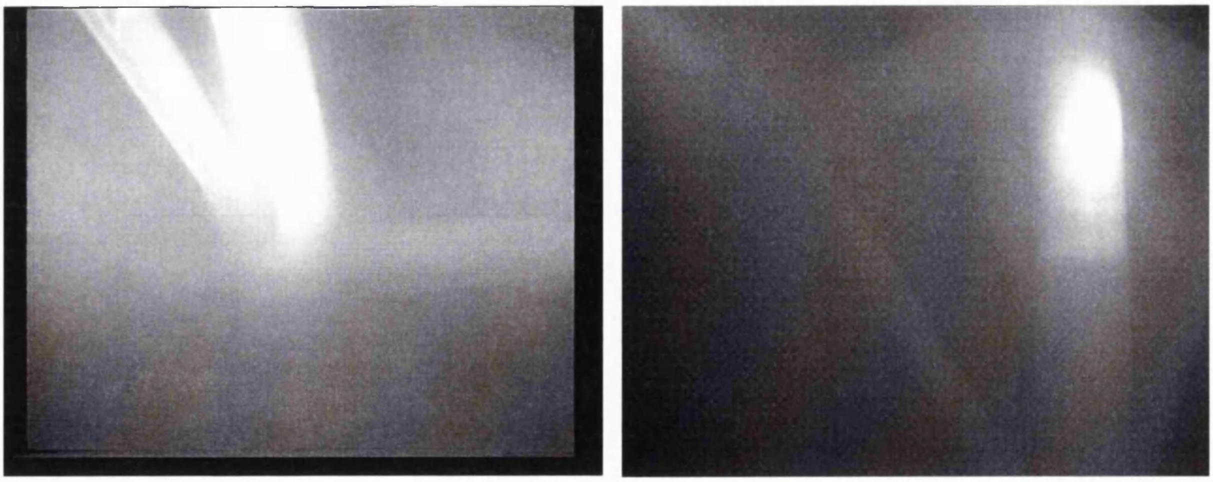


Figure 4.13 **Real underwater image (left) and laboratory image (right) from poor visibility test sequence (degraded with dry ice fog). Both images exhibit similar degradation.**

4.2.9 Generating variable lighting conditions

An important aspect of underwater imagery, filmed from an ROV in limited visibility, is that the lighting is both non-uniform (one portion of the image may be brightly illuminated whereas another portion may be dark) and dynamic (lighting conditions change from one image to the next). These conditions arise as a result of illumination by spotlights mounted on and moving with the underwater vehicle. Dynamic lighting conditions were simulated by mounting a pair of Maglite, focussed beam torches next to the camera on the robot-camera attachment rig.

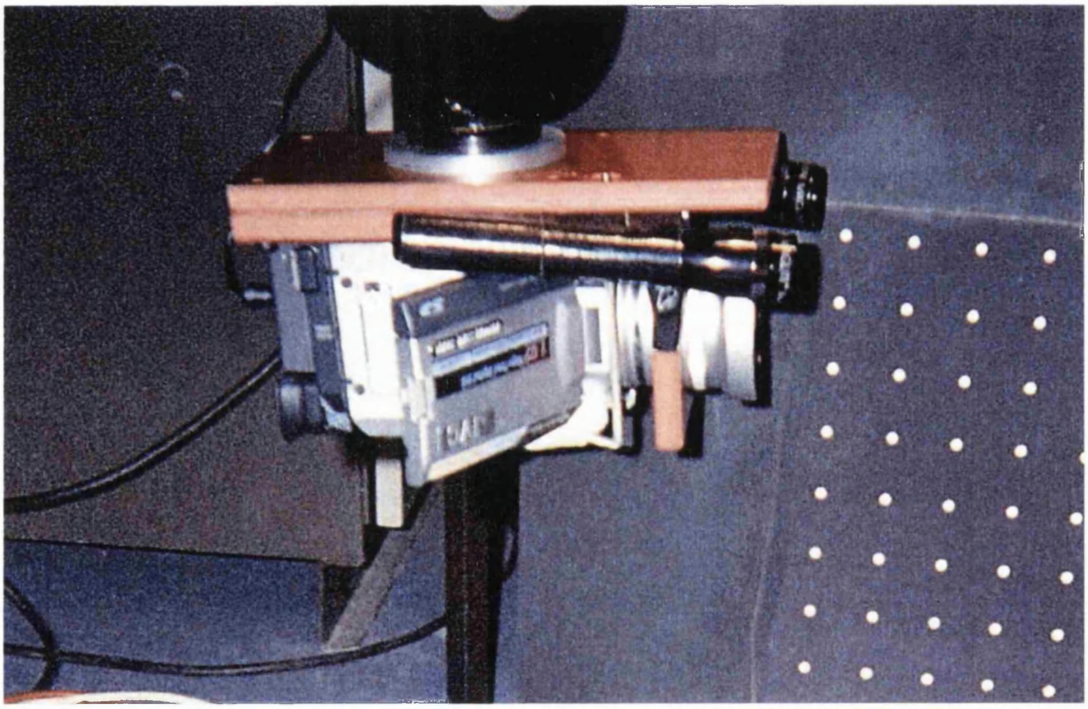


Figure 4.14 Camera rig with attached spotlights.

4.2.10 Image capture

Video sequences were recorded on a JVC GR-DV2000 digital cam-corder at 25 frames per second. The cam-corder was carefully selected to provide a number of important features. It was useful to have a progressive scan facility (as opposed to conventional interlaced scan) so that each frame provided a complete image for individual analysis. Often, practical applications (e.g. ROVs) use conventional interlaced video. This can be inconvenient to analyse, with data needing to be averaged or discarded. However, it seems reasonable to use non-interlaced digital footage for this work, since these camera systems could always be substituted for interlaced video, and may become increasingly popular in the future. It was important that various automatic features (including auto-focus and automatic motion compensation systems) could be switched off so that the camera projection matrix remained unchanged both between and during image sequences.

The cam-corder stores video data in Digital Video format on Mini DV tape cassettes. The sequences were output to a PC via a "Fire-Wire" card. They were then broken down into individual frames using standard software tools. Each frame was compressed and stored in a "portable network graphic" (.png) file format. Both vision algorithms and image processing, calibration and analysis procedures were implemented in JAVA. These programs are able to open and read sequences of .png image files.

4.3 Data analysis

4.3.1 Introduction

Having programmed the robot with a suitable trajectory, many calibration sequences were filmed along that trajectory. Several sequences were also filmed for each object and for each level of visibility (clear, foggy with fixed lights, foggy with moving lights).

A sample of each of these sequences was selected such that good synchronisation (see section 4.2.2) was achieved between each test sequence and the corresponding calibration sequence. Quality of synchronisation was assessed by finding the two frames (one from each sequence) showing the "synchronisation spot" which best matched each other when superimposed using image differencing.

Once a suitable calibration sequence had been selected, it was analysed in order to extract the camera characteristics and the position and orientation of the camera at every frame in the sequence.

4.3.2 Summary of data analysis procedure

Software was constructed that automated most of the steps in the analysis procedure which was as follows:

1) Detect all spots in every image. Output the image co-ordinates of each spot centroid to a text file.

2) User labels by hand a selection of spots in one or more frames. This involves entering the target co-ordinates (of at least four spots from each target) into the text file of corresponding spot centroids.

3) Automatic labelling by projecting (calculate homography from existing spot labels, project all possible spot labels through this homography and look for matches in spot centroid image co-ordinates) and by propagating (use spot labels in one frame to label close matches in adjacent frames both forwards and backwards along the image sequence).

4) Complete steps 1-3 on the video sequence being measured and also on a selection of approximately twenty good calibration images featuring multiple targets. This "calibration set" consists of separate still images containing good views of all three targets or pairs of targets. This set of images is composed partly from the "target relations" images-still images which were filmed separately to the sequence being measured. The set also includes a selection of images from the calibration sequence itself.

5) Use the calibration set images to extract initial estimate values for camera Intrinsic parameters (focal length, pixel aspect ratio and principal point location). Use these to generate initial estimate Extrinsic parameters (position and orientation of camera) for each target in frames showing multiple targets. Initial estimate Extrinsics yield initial estimate target relations matrices (the transformations relating the co-ordinate frames of each target to that of the base target). For initial estimates assume no lens distortion.

6) Now optimise Intrinsics, Extrinsics, Target Relations and Lens Distortion parameters over all images in the calibration set using Powell's method (Press [1992], Nocedal [1999]).

7) Use optimised Intrinsics, Target Relations, Lens Distortion parameters and an extracted homography to optimise for Extrinsics (relative to base target) using Powell's method in one image selected from the middle of the video sequence (calibration sequence).

8) Use the optimised Extrinsics, for the frame selected above, as an initial estimate for adjacent frames. Optimise extrinsics (given the already optimised Intrinsics, Target Relations, Lens Distortion values) for these frames using Powell's method. Thus propagate forwards and backwards through the entire video sequence generating optimised Extrinsics for every frame.

Note that two different sets of images are used during the calibration process. One set is the video sequence of interest, filmed along some camera motion trajectory. This is sometimes referred to as the "calibration sequence" (distinguishing it from the corresponding poor visibility sequence, referred to as a "test" sequence) and also as the "trajectory sequence" (distinguishing it from the individual, still images of the "calibration set"). The other set of images that must be labelled is the "calibration set" consisting of about 20 still images, not filmed as consecutive images in any sequence. This "calibration set" is necessary to provide data about the position and orientation of each target relative to the base target since, depending on the camera trajectory, the base target may not be observed during the trajectory sequence itself.

The base target is important because this allows us to determine the position of the camera in the same co-ordinate system as that in which the position of the object being viewed is known.

It should be noted that Intrinsics, lens distortion parameters and target relations are optimised only over the calibration set (20 images) and not over the entire calibration video sequence (approximately 1000 frames), since this would entail an impossibly high dimensional search space.

This procedure is illustrated in the following flow chart (figure 4.15). Each of these steps will now be explained in detail.

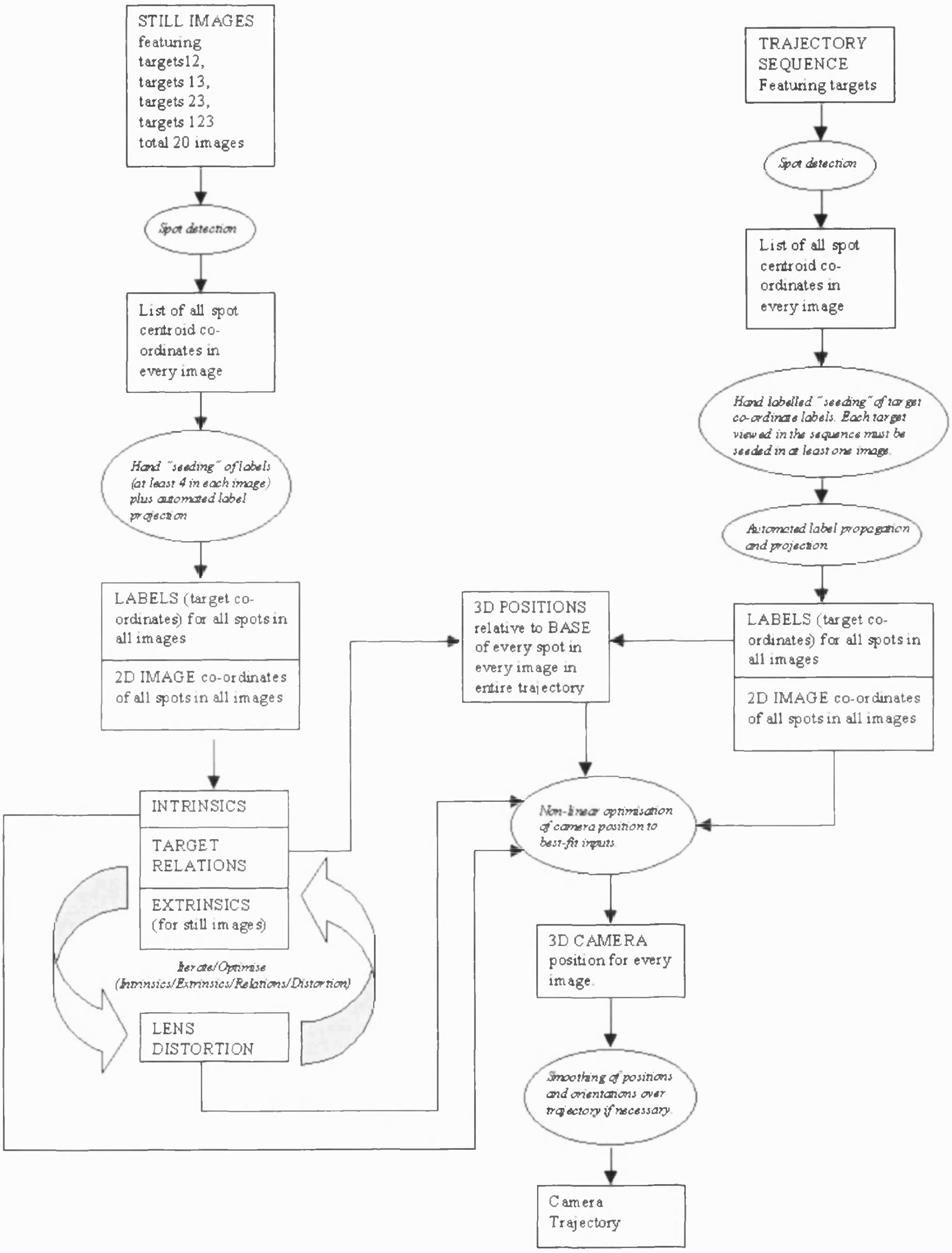

STILL IMAGES featuring targets12, targets 13, targets 23, targets 123 total 20 images
Spot detection
List of all spot centroid co-ordinates in every image
Hand "seeding" of labels (at least 4 in each image) plus automated label projection
LABELS (target co-ordinates) for all spots in all images
2D IMAGE co-ordinates of all spots in all images
INTRINSICS
TARGET RELATIONS
EXTRINSICS (for still images)
Iterate/Optimise (Intrinsics/Extrinsics/Relations/Distortion)
LENS DISTORTION
TRAJECTORY SEQUENCE Featuring targets
Spot detection
List of all spot centroid co-ordinates in every image
Hand labelled "seeding" of target co-ordinate labels. Each target viewed in the sequence must be seeded in at least one image.
Automated label propagation and projection
LABELS (target co-ordinates) for all spots in all images
2D IMAGE co-ordinates of all spots in all images
3D POSITIONS relative to BASE of every spot in every image in entire trajectory
Non-linear optimisation of camera position to best-fit inputs
3D CAMERA position for every image.
Smoothing of positions and orientations over trajectory if necessary.
Camera Trajectory

Figure 4.15 **Calibration and trajectory extraction strategy.**

4.3.3 Feature extraction

"Spot detection" software was created in order to locate the centroids of spots in images. The procedure for detecting spots was as follows:

- The image is severely blurred by convolving with a broad gaussian kernel.

- The blurred image is subtracted from the original image in order to leave the background more consistently dark and improve contrast between the spots and the background.

- The modified image is then thresholded. The threshold is chosen as follows:
  - Since the image consists of bright spots on a dark background, Pixel grey-levels can be expected to be distributed according to two major clusters (see figure 4.16).

  - The optimum threshold grey-level value lies some proportion of the distance between the means of these two clusters. Good results were obtained by using a proportion of 70% (i.e. threshold equals background mean plus 70% of difference between background mean and spot mean).

  - Unfortunately, since the spots have not yet been located, the mean pixel grey-level values are not known for either background or spots. However, since the vast majority of image pixels must be background pixels, the background mean can be approximated to the mean grey-level value for the whole image.

The spot mean can reasonably be approximated by the brightest pixel value in the image (see figure 4.16).

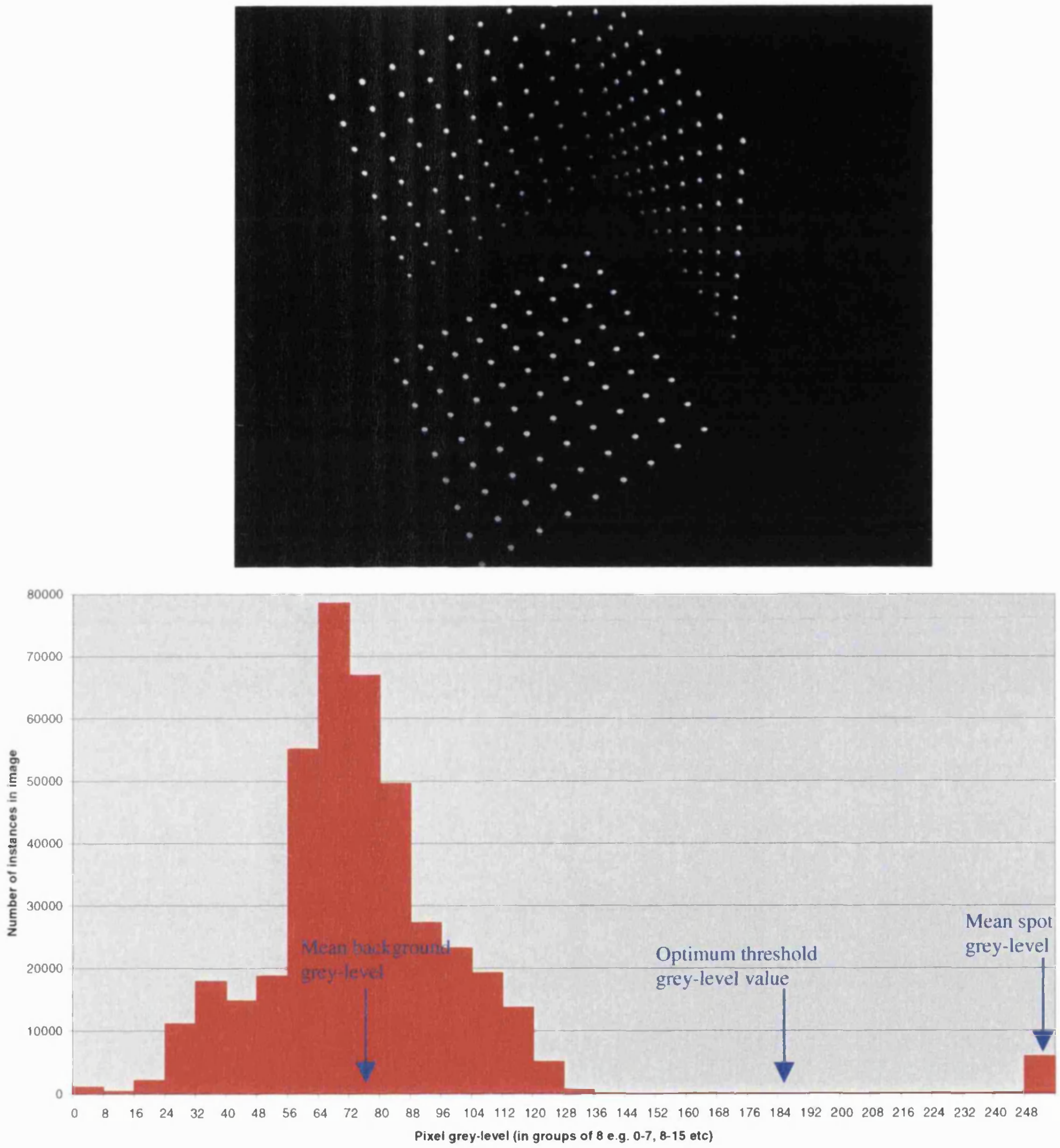


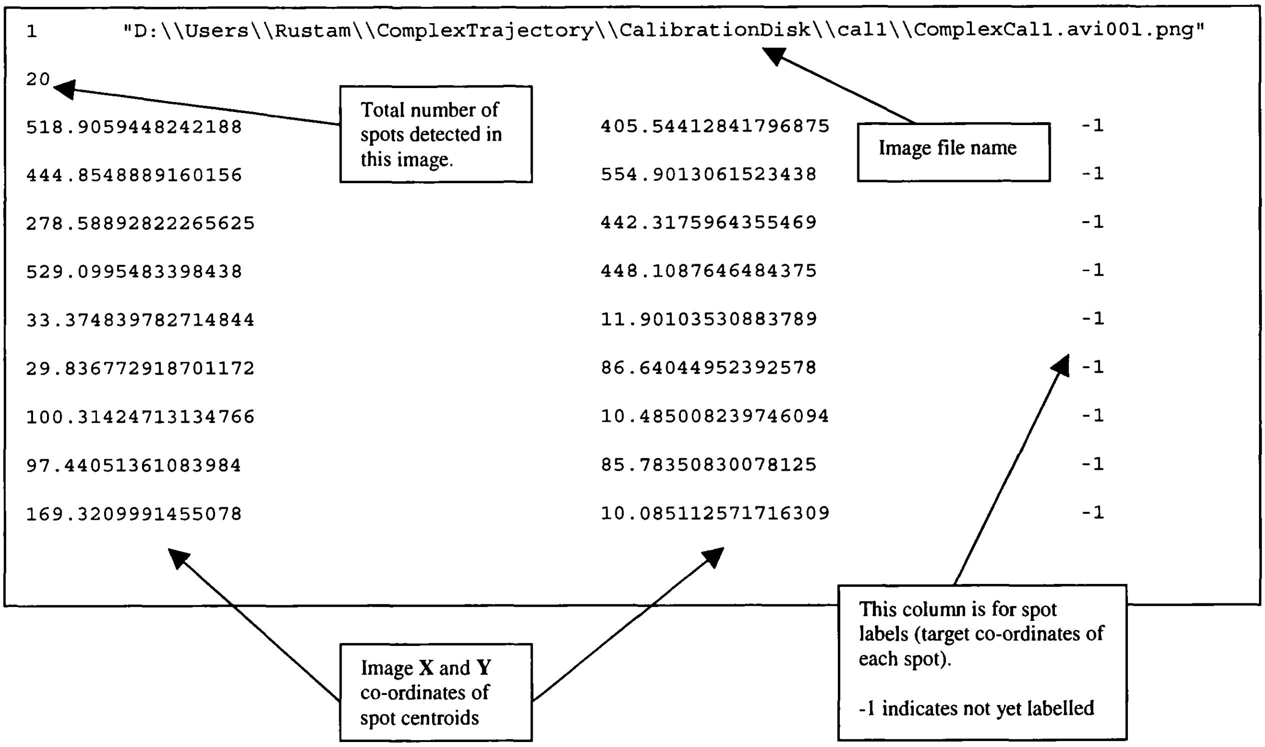
Figure 4.16 **A calibration image and its grey-level histogram. The mean grey-level of spot features can be approximated to the brightest pixel value in the image. The mean background grey-level can be approximated by the mean of the entire image.**

- All pixels above the threshold grey-level value are now clustered into "blobs". A pixel is classified as being a member of a particular blob if it is in contact with any other member pixels of the blob, i.e. a next door neighbour pixel.

- Small or dim blobs are now discarded. A "significance value" is assigned to each blob depending on the number of pixels in the blob and the mean brightness of these pixels. A threshold value for the significance value is specified. Any blobs that fall below the significance threshold level are discarded. Useful significance threshold values can be found by trial and error, depending on the set of images being analysed.

- The remaining blobs are now all considered to be genuine target spots. The centre of these spots is now estimated as the blob centroid. When computing spot centroids, each member pixel is weighted in proportion to its brightness.

- A list of the image co-ordinates of every detected spot centroid in every image is now output as a text file.



```
1      "D:\\Users\\Rustam\\ComplexTrajectory\\CalibrationDisk\\cal1\\ComplexCal1.avi001.png"
20
518.9059448242188                405.54412841796875               -1
444.8548889160156                554.9013061523438                -1
278.58892822265625               442.3175964355469                -1
529.0995483398438                448.1087646484375                -1
33.374839782714844               11.90103530883789                -1
29.836772918701172               86.64044952392578                -1
100.31424713134766               10.485008239746094               -1
97.44051361083984                85.78350830078125                -1
169.3209991455078                10.085112571716309               -1
```

Total number of spots detected in this image.
Image file name
Image X and Y co-ordinates of spot centroids
This column is for spot labels (target co-ordinates of each spot).
-1 indicates not yet labelled

Figure 4.17 Extract from an output text file illustrating the layout of data.

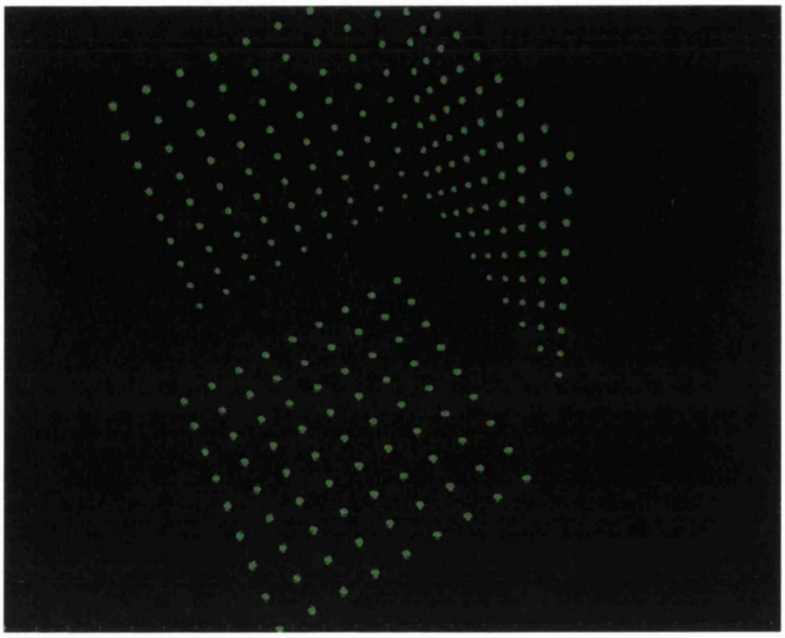


Figure 4.18 **Feature detection. The thresholded pixels have been grouped into "blobs" and any "blob" that is too small or too dim has been discarded. Remaining blobs (shown in green) are assumed to be calibration spots. The centroids of these spots have been marked with a blue pixel.**

4.3.4 Feature labelling

The feature detection process locates, for each image, the centroids of all spots observed in that image. These centroid positions are output as a list of image co-ordinates in units of pixels. It will be necessary to compute homographies (mathematical relationships between points in the target planes and points in the image planes). The nature and use of homographies is discussed in detail in section 4.3.8. In order to compute homographic relationships between the image plane and each of the calibration target planes, it is necessary to determine the positions of the spots in terms of co-ordinate systems set in each of the targets.

To this end (see figure 4.8), each spot on each target is assigned a numerical label according to its X-Y position in that target. The labels of each spot refer to that spot's position within a co-ordinate frame set in the target to which the spot belongs. Another number is used to identify the target itself e.g. "367" indicates the spot on target 3 with X-Y co-ordinates (6, 7). The units of this co-ordinate system are "spot spaces", each spot space being 30mm.

In order to compute homographic relationships, it was thus necessary to determine the "label" of each spot observed in each image. Since video sequences of up to 1000 frames were to be analysed, this process needed to be largely automated. Two algorithms were developed to aid in this process:

1) "Projection"

- A small number of labelled spots within a single image are used to determine the labels of the remaining spots in that image.

  A minimum of four labelled spots are required for each target whose spots appear in the image. Three or more of these spots must not be co-linear as this results in a loss of constraint.

- For each target in the image, the four (or more) labelled spots are used to approximate the homography (see section 4.3.9) between that target and the image plane.

- This homography is used to project the target co-ordinates of the entire 9×9 grid of spots from that target, thus giving the expected image co-ordinates for every spot in that target.

- These predicted image co-ordinates are compared with the list of spot centroid co-ordinates produced by the feature detection process (see section 4.3.8).

- If a detected spot centroid lies within a specified maximum distance (e.g. ±5 pixels) of a predicted spot co-ordinate, then that detected spot is assigned the label of the matching projected spot centroid.

- The ±5 pixels range allows for errors due to lack of knowledge of camera lens distortion parameters and also errors in the homography estimate resulting from a sparse set of known spot correspondences.

2) "Propagation"

- A set of spot labels for one image frame in a video sequence is used to generate labels for spots viewed in chronologically adjacent images.

- The detected spot centroid positions in the labelled frame are compared to the detected spot centroid positions in the adjacent, unlabelled frame. If any spot position in the unlabelled frame lies within a specified maximum distance (e.g. ±5 pixels) of a spot position in the adjacent labelled frame, that label is assigned to the unlabelled spot.

- The ±5 pixels range allows for motion of the camera between successive frames. The optimum value for this error constraint will depend on the speed of camera motion and the frame rate of image acquisition.

Spots must be labelled in two different sets of images. One set is the video sequence of interest, filmed along some camera motion trajectory. This is referred to as the "trajectory sequence". The other set of images that must be labelled is the "calibration set" consisting of about 20 still images, not filmed as consecutive images in any sequence.

- Procedure for labelling "calibration set" images:
  - For each image, every visible target must be "hand-seeded". This involves the user identifying at least four spots in each visible target in the image and entering their target co-ordinate labels into the text file list (output from the feature detection process).

  - For each image, the "projection" labelling process is iterated until no new spot labels are identified. This process normally terminates within two iterations.

- Procedure for labelling "trajectory sequence":
  - A small number of images, scattered throughout the sequence are "hand-seeded" with a small number of spot labels. Each target that is viewed at any time during the sequence, must be hand-seeded in at least one image of the sequence. The hand-seeding must provide the labels for at least four non-co-linear spots in each target.

  - The "projection" labelling process is performed on every image in the sequence. This is iterated until no new labels are created (usually only one or two iterations are necessary).

- The “propagation” labelling process is performed both forwards and backwards from each end, end to end, along the entire image sequence.

- The “projection” labelling process is again performed on every image. The projection and propagation processes are now iteratively alternated until no new labels are found.

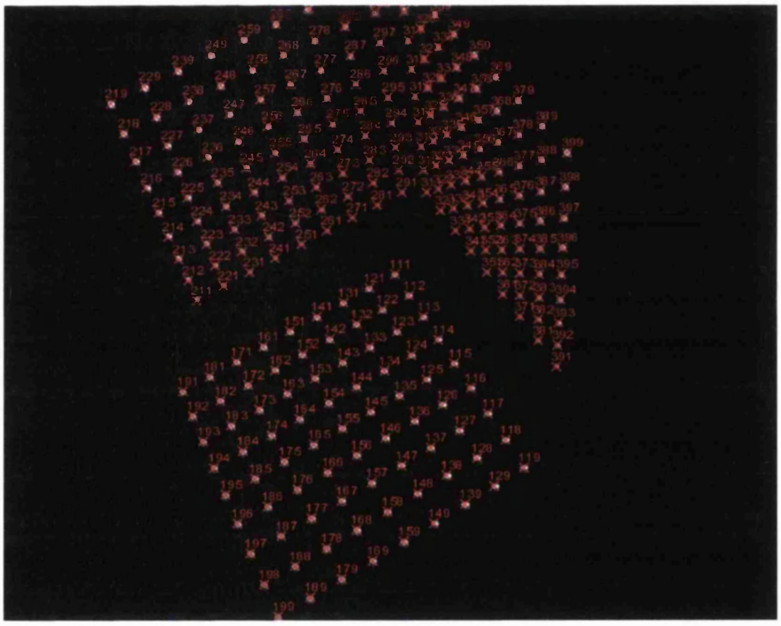


Figure 4.19 **Output from feature labelling process. The figure is a visual aid, illustrating the locations of detected spot-centroids and their computed spot-labels.**

4.3.5 Camera co-ordinate system: perspective projection

The following sections (4.3.5-4.3.15) set out the theory necessary for calibrating the camera and extracting the camera trajectory. The camera is treated as a standard pin-hole model. Consider a 3D point $\underline{\mathbf{X}}_c = (X_c, Y_c, Z_c)$ in the camera co-ordinate frame (see diagram) which projects onto a 2D point $\underline{x}_c = (x_c, y_c)$ in the image plane.

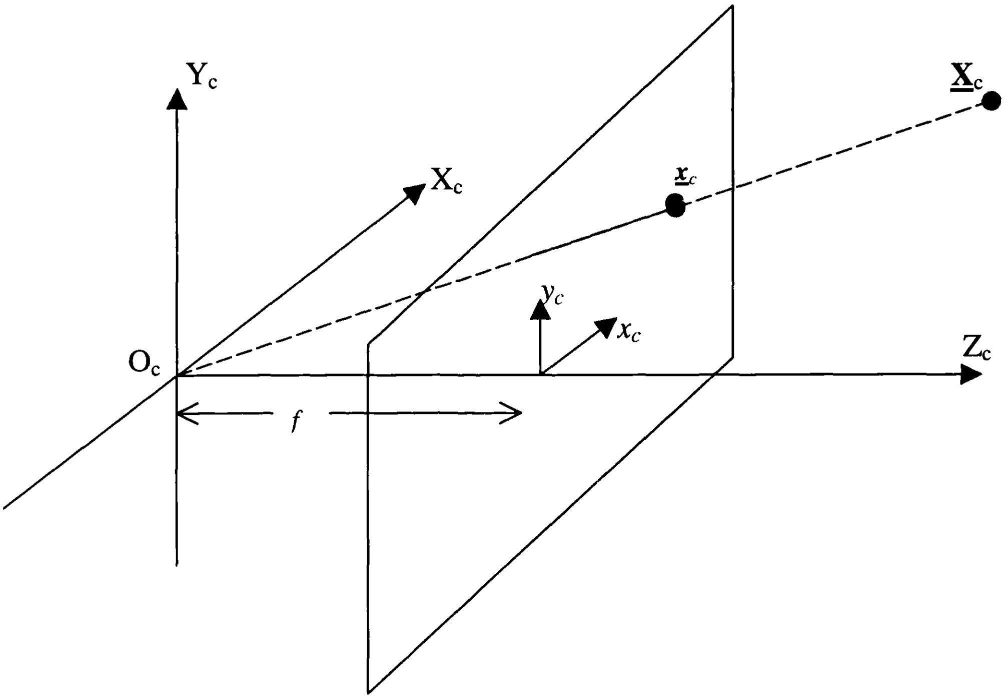


Figure 4.20 Perspective projection of a pin-hole camera.

This mapping from 3D to 2D can be described by a 3×4 "projection matrix" using a homogeneous* co-ordinate system:

$$\begin{bmatrix} x_c \\ y_c \\ f \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}$$ **Equation 4.1**

such that $x_c = \left(\frac{f}{Z_c}\right) X_c$ and $y_c = \left(\frac{f}{Z_c}\right) Y_c$.

*Note: the homogeneous vector $\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$ maps to the 3D point $\begin{bmatrix} x/w \\ y/w \\ z/w \end{bmatrix}$ and the homogeneous vector $\begin{bmatrix} x \\ y \\ w \end{bmatrix}$ maps to the 2D point $\begin{bmatrix} x/w \\ y/w \end{bmatrix}$.

4.3.6 Image co-ordinate system: intrinsic camera parameters

When dealing with digital images it is necessary to consider the pixelated nature of the image plane.

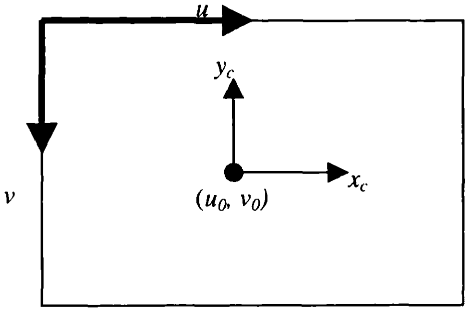


Figure 4.21 Image plane co-ordinate system.

The position of a pixel in an image is described as the u^{th} pixel horizontally and the v^{th} pixel vertically from the top left corner. If the optical axis intersects the image plane at the "principal point" (u_0, v_0) and the number of pixels per unit length in the u and v directions respectively are k_u and k_v then the (x_c, y_c) co-ordinates are related to the (u,v) co-ordinates by a 3×3 upper triangular "camera calibration matrix":

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ f \end{bmatrix} = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}$$

Equation 4.2

where $\alpha = fk_u$ $\beta = fk_v$ and γ is a parameter describing the skewness between the u and v axes. In practice γ is usually close to zero and was assumed to be zero during this work.

4.3.7 World co-ordinate system: extrinsic camera parameters

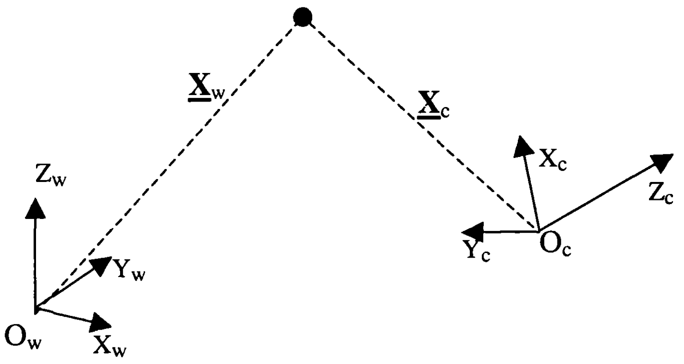


Figure 4.22 Relationship between world co-ordinate system and camera co-ordinate system. Subcript $_w$ stands for "world" and subscript $_C$ stands for "camera".

This section considers the Euclidean transformation between a 3D point, $\underline{X}_w$ in a world co-ordinate frame, and the same 3D point $\underline{X}_c$ described in the camera co-ordinate frame. In general this is a six degree of freedom rigid body transformation which can be expressed as:

$$\underline{X}_c = R\underline{X}_w + T \qquad \textbf{Equation 4.3}$$

Or in homogeneous co-ordinates:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \qquad \textbf{Equation 4.4}$$

Where **R** is a 3×3 rotation matrix and **T** is a translation vector.

The calibration, projection and extrinsic matrices can now be concatenated to give:

$$\underline{\mathbf{x}}_i = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \qquad \textbf{Equation 4.5}$$

This simplifies to give:

$$\underline{\mathbf{x}}_i = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} [\mathbf{R} \quad \mathbf{T}] \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad \textbf{or} \quad \underline{\mathbf{x}}_i = \mathbf{C}[\mathbf{R} \quad \mathbf{T}]\mathbf{X}_w \qquad \textbf{Equation 4.6}$$

In general, to fully calibrate the camera, it is necessary to determine 10 parameters- 6 extrinsic parameters and 4 calibration or "intrinsic" parameters (if γ is assumed to be zero).

4.3.8 Homography between a target plane and its image

It is possible (Zhang [1998]) to calibrate a camera by capturing images of a planar target. If the world co-ordinate system is defined such that the target plane lies on $Z_w = 0$, then points $\underline{\mathbf{X}}_t$ on the target plane are mapped to points $\underline{\mathbf{x}}_i$ on the image plane by:

$$\underline{\mathbf{x}}_i = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{C}[\mathbf{R} \quad \mathbf{T}] \begin{bmatrix} X_t \\ Y_t \\ 0 \\ 1 \end{bmatrix} = \mathbf{C}[\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{r}_3 \quad \mathbf{T}] \begin{bmatrix} X_t \\ Y_t \\ 0 \\ 1 \end{bmatrix} = \mathbf{C}[\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{T}] \begin{bmatrix} X_t \\ Y_t \\ 1 \end{bmatrix} \qquad \textbf{Equation 4.7}$$

Thus $\mathbf{x_i}$ and $\underline{\mathbf{X}}_t$ are related by the *homography* (straight line preserving mapping between two planes) **H** such that:

$$\underline{\mathbf{x}}_i = \mathbf{H}\,\underline{\mathbf{X}}_t \qquad \text{where } \mathbf{H} = \mathbf{C}[\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{T}] = [\mathbf{h}_1 \quad \mathbf{h}_2 \quad \mathbf{h}_3] \qquad \textbf{Equation 4.8}$$

In general a homography possesses 8 degrees of freedom (4 intrinsics, 2 rotations and 2 translations). It should therefore be possible to extract the homography given both the image and target plane/world co-ordinates of four target points (as each planar point yields two constraints-x and y). In practice, extracted image points are subject to noise and so the resulting four pairs of simultaneous equations have no exact solution. Nevertheless, a good estimate for the homography can be obtained (Zhang [1998]) by using non-linear optimisation techniques. In this case Powell's method was used (Press [1992]).

4.3.9 Computing the homography between a target plane and an image

A homography is a transformation which maps points from one plane to another. This transformation is constrained in that straight lines in one plane are mapped to straight lines in the other. Using homogeneous co-ordinates, a homography can be expressed as a *3×3* matrix which multiplies the homogeneous vector of a point in one plane to yield the homogeneous vector (up to some arbitrary scaling factor *w*) describing a point in the other plane.

$$\textbf{e.g.} \qquad w\begin{bmatrix} x_{plane2} \\ y_{plane2} \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \cdot \begin{bmatrix} x_{plane1} \\ y_{plane1} \\ 1 \end{bmatrix} \qquad \textbf{Equation 4.9}$$

This idea is usefully applied to the camera calibration problem (Zhang [1998]). Since planar calibration targets were used (each consisting of a square grid of circular spots), the camera becomes a device which maps points (spot centres) in the target plane onto corresponding image points (located as spot centroids) in the image plane. This transformation is (ignoring lens distortion) clearly an example of an homography. It is useful to compute this homography since it must encode information about both the *intrinsic* and *extrinsic* properties of the camera:

$$\textbf{i.e.} \quad \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \equiv \mathrm{C.E} \equiv \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{T} \end{bmatrix} \qquad \textbf{Equation 4.10}$$

Each spot centroid that is successfully located and labelled yields a pair of target plane co-ordinates (X_n, Y_n) in units of "spot-spaces" and a corresponding pair of image plane co-ordinates (u_n, v_n) in units of pixels.

A homography contains nine elements but is only unique up to some arbitrary scaling factor. Hence each homography has eight degrees of freedom. Since each known spot yields two constraints (mapping of x and y co-ordinates) it follows that, if a minimum of at least four spot centroids are known and labelled, it is possible to deduce a unique 3×3 homography matrix as a closed form solution.

In practice the data yielded by each spot relationship is noisy. It is therefore desirable to use a large number, n , spot relationships and compute the homography which best fits the resulting n sets of simultaneous equations:

$$\begin{bmatrix} w_1u_1 & w_2u_2 & w_3u_3 & . & . & . & w_nu_n \\ w_1v_1 & w_2v_2 & w_3v_3 & . & . & . & w_nv_n \\ w_1 & w_2 & w_3 & . & . & . & w_n \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} . \begin{bmatrix} X_1 & X_2 & X_3 & . & . & . & X_n \\ Y_1 & Y_2 & Y_3 & . & . & . & Y_n \\ 1 & 1 & 1 & . & . & . & 1 \end{bmatrix}$$

Equation 4.11

Expanding the above expression, any one spot relationship yields three simultaneous equations of the form:

$$X_n h_{11} + Y_n h_{12} + h_{13} = w_n u_n \quad \textbf{or} \quad X_n h_{11} + Y_n h_{12} + h_{13} - u_n w_n = 0 \quad \textbf{Equation 4.12}$$

$$X_n h_{21} + Y_n h_{22} + h_{23} = w_n v_n \quad \textbf{or} \quad X_n h_{21} + Y_n h_{22} + h_{23} - v_n w_n = 0 \quad \textbf{Equation 4.13}$$

$$X_n h_{31} + Y_n h_{32} + h_{33} = w_n \quad \textbf{or} \quad X_n h_{31} + Y_n h_{32} + h_{33} - w_n = 0 \quad \textbf{Equation 4.14}$$

Extracting all the unknowns to form an unknown vector, rearranging and then stacking all *n* sets of simultaneous equations, yields the following matrix equation:

$$\begin{bmatrix}
X_1 & Y_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -u_1 & 0 & 0 & . & . & . & 0 \\
0 & 0 & 0 & X_1 & Y_1 & 1 & 0 & 0 & 0 & -v_1 & 0 & 0 & . & . & . & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & X_1 & Y_1 & 1 & -1 & 0 & 0 & . & . & . & 0 \\
X_2 & Y_2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -u_2 & 0 & . & . & . & 0 \\
0 & 0 & 0 & X_2 & Y_2 & 1 & 0 & 0 & 0 & 0 & -v_2 & 0 & . & . & . & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & X_2 & Y_2 & 1 & 0 & -1 & 0 & . & . & . & 0 \\
X_3 & Y_3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -u_3 & . & . & . & 0 \\
0 & 0 & 0 & X_3 & Y_3 & 1 & 0 & 0 & 0 & 0 & 0 & -v_3 & . & . & . & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & X_3 & Y_3 & 1 & 0 & 0 & -1 & . & . & . & 0 \\
. & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . \\
. & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . \\
. & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . \\
. & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . \\
. & . & . & . & . & . & . & . & . & . & etc & . & . & . & . & . \\
. & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . \\
. & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . \\
. & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . \\
X_n & Y_n & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -u_n \\
0 & 0 & 0 & X_n & Y_n & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -v_n \\
0 & 0 & 0 & 0 & 0 & 0 & X_n & Y_n & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1
\end{bmatrix}
.
\begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \\ w_1 \\ w_2 \\ w_3 \\ . \\ . \\ . \\ w_n \end{bmatrix}
=
\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ . \\ . \\ . \\ 0 \end{bmatrix}$$

Equation 4.15

This is of the form **A**.$\underline{\mathbf{x}}$ = 0 where **A** contains only known quantities and $\underline{\mathbf{x}}$ contains only unknown quantities. The unknown vector can readily be found using singular value decomposition (Press [1992]) to yield a least squares best fit for the values of $\underline{\mathbf{x}}$.

4.3.10 Locating one target relative to another

During the calibration sequence, the camera will be moved past several targets during the motion of the robot manipulator. The targets should be arranged such that at least one target is satisfactorily viewed in each image in the video sequence.

Once the intrinsic parameters of the camera have been measured (e.g. from a few initial images of target planes) they do not need to be re-calculated at successive images. Given the intrinsics, the extrinsics can be calculated in successive images from a good view of just one target, thus yielding the camera position relative to that target.

The *world* co-ordinates of the camera are required for every image in the sequence. It is therefore necessary to choose *one* target plane in which to locate the world co-ordinates and then pre-calculate the Euclidean transformation that maps points from frames located in all other targets onto the world co-ordinate frame.

Given a single image of two target planes *A* and *B*, it is possible to extract the homographies between each plane and the image plane of the camera. Let a single 3D point be described by:

$\mathbf{X}_A$ in a co-ordinate frame located in target plane *A*

$\mathbf{X}_B$ in a co-ordinate frame located in target plane *B*

$\mathbf{x}_C$ in normalised image co-ordinates

With the two homographies being $\mathbf{H}_A^C$ and $\mathbf{H}_B^C$ such that:

$$\mathbf{x}_C = \mathbf{H}_A^C \mathbf{X}_A = \mathbf{H}_B^C \mathbf{X}_B \qquad \textbf{Equation 4.17}$$

If the absolute world co-ordinates are chosen to be centred in the A target plane then it may be useful to know the world co-ordinates $\mathbf{X}_A$, given measured co-ordinates $\mathbf{X}_B$. These are given by:

$$\mathbf{X}_A = \left(\mathbf{H}_A^C\right)^{-1}\mathbf{x}_C = \left(\mathbf{H}_A^C\right)^{-1}\mathbf{H}_B^C\mathbf{X}_B \qquad \textbf{Equation 4.18}$$

4.3.11 Constraints on the intrinsic parameters

The following analysis derives from Zhang [1998]. A single image of the target plane allows an homography to be estimated (see section 4.3.9):

$$\mathbf{H} = [\mathbf{h}_1 \quad \mathbf{h}_2 \quad \mathbf{h}_3] = \mathbf{C}[\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{T}] \qquad \textbf{Equation 4.19}$$

Using the knowledge that all column vectors of a rotation matrix are orthonormal (since in general a rotation may only possess three degrees of freedom), yields the following constraints on the intrinsic parameters:

$$\mathbf{r}_1^T\mathbf{r}_2 = 0 \qquad \textbf{Equation 4.20}$$

and

$$\mathbf{r}_1^T\mathbf{r}_1 = \mathbf{r}_2^T\mathbf{r}_2 \qquad \textbf{Equation 4.21}$$

Since $\mathbf{r}_n = \mathbf{C}^{-1}\mathbf{h}_n$ these become:

$$\mathbf{h}_1^T\mathbf{C}^{-T}\mathbf{C}^{-1}\mathbf{h}_2 = 0 \qquad \textbf{Equation 4.22}$$

and

$$\mathbf{h}_1^T\mathbf{C}^{-T}\mathbf{C}^{-1}\mathbf{h}_1 = \mathbf{h}_2^T\mathbf{C}^{-T}\mathbf{C}^{-1}\mathbf{h}_2 \qquad \textbf{Equation 4.23}$$

Thus *one* homography provides *two* constraints on the intrinsic parameters.

4.3.12 Solving for the intrinsic and extrinsic parameters

The matrix of intrinsic camera parameters is typically (Zhang [1998]) characterised as:

$$\mathbf{C} = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

More generally, since the matrix is only defined up to an arbitrary scale factor λ:

$$\mathbf{C} = \begin{bmatrix} \lambda\alpha & \lambda\gamma & \lambda u_0 \\ 0 & \lambda\beta & \lambda v_0 \\ 0 & 0 & \lambda \end{bmatrix}$$

which yields:

$$\mathbf{C}^{-T}\mathbf{C}^{-1} = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix} = \frac{1}{\lambda^2}\begin{bmatrix} \frac{1}{\alpha^2} & -\frac{\gamma}{\alpha^2\beta} & \frac{\gamma v_0 - u_0\beta}{\alpha^2\beta} \\ -\frac{\gamma}{\alpha^2\beta} & \frac{\gamma^2}{\alpha^2\beta^2}+\frac{1}{\beta^2} & -\frac{\gamma(\gamma v_0 - u_0\beta)}{\alpha^2\beta^2}-\frac{v_0}{\beta^2} \\ \frac{\gamma v_0 - u_0\beta}{\alpha^2\beta} & -\frac{\gamma(\gamma v_0 - u_0\beta)}{\alpha^2\beta^2}-\frac{v_0}{\beta^2} & \frac{(\gamma v_0 - u_0\beta)^2}{\alpha^2\beta^2}+\frac{v_0^{\;2}}{\beta^2}+1 \end{bmatrix}$$

Equation 4.24

During this work, γ was assumed to be zero. For simplification, α and β were assumed to be equal *as an initial estimate* (i.e. pixels were assumed to be square). These initial approximations considerably simplify equation 4.24. (In fact, the pixels of the cam-corder are *not* square. The final, non-square values for α and β are found during the final stage of non-linear optimisation where all the intrinsic quantities are iteratively refined (see section 4.3.14)).

With the above assumptions, **C** simplifies to give: $\mathbf{C} = \lambda \begin{bmatrix} \alpha & 0 & u_0 \\ 0 & \alpha & v_0 \\ 0 & 0 & 1 \end{bmatrix}$

Giving $$\mathbf{C}^{-T}\mathbf{C}^{-1} = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix} = \frac{1}{\lambda^2} \begin{bmatrix} \frac{1}{\alpha^2} & 0 & \frac{-u_0}{\alpha^2} \\ 0 & \frac{1}{\alpha^2} & \frac{-v_0}{\alpha^2} \\ \frac{-u_0}{\alpha^2} & \frac{-v_0}{\alpha^2} & \frac{u_0^2}{\alpha^2} + \frac{v_0^{\ 2}}{\alpha^2} + 1 \end{bmatrix}$$ **Equation 4.25**

Since this matrix is symmetric, it can be defined by a 6D vector:

$\mathbf{b} = [B_{11}, B_{12}, B_{22}, B_{13}, B_{23}, B_{33}]^T$

But, since $B_{11} = B_{22}$ and B_{33} is a function of B_{13}, B_{23} and λ, **b** need only contain 4 elements (to encode information about the 4 variables $\alpha, u_0, v_0, \lambda$):

$\mathbf{b} = [B_{11}, B_{13}, B_{23}, B_{33}]^T$

It is now possible to express $\mathbf{h}_i^T \mathbf{C}^{-T} \mathbf{C}^{-1} \mathbf{h}_j$ in the form: $\mathbf{v}_{ij}^T \mathbf{b}$ **Equation 4.26**

Where $\mathbf{v}_{ij}^T = [(h_{i1}h_{j2} + h_{i2}h_{j2}), (h_{i3}h_{j1} + h_{i1}h_{j3}), (h_{i3}h_{j2} + h_{i2}h_{j3}), h_{i3}h_{j3}]$

The two constraints from each homography can now be written as:

$$\begin{bmatrix} \mathbf{v}_{12}^T \\ (\mathbf{v}_{11} - \mathbf{v}_{22})^T \end{bmatrix} \mathbf{b} = 0$$ **Equation 4.27**

Given *n* such homographies (obtained from *n* images of a single target or one image of *n* targets), *n* such pairs of equations can be stacked to give:

$$\mathbf{Vb} = 0 \qquad \text{where } \mathbf{V} \text{ is a } 2n\times 4 \text{ matrix.}$$ **Equation 4.28**

If the number of homographies n is greater than 3 (or 2, discounting the skewness parameter γ) then it should be possible to solve for **b** and hence the intrinsic camera parameters **C**. In practice there will not be an exact solution due to noise in the measurement/modelling process, however a least squares estimate can be obtained from singular value decomposition (Press [1992]).

Given the intrinsics, the extrinsics are readily solved from **C** and **H**. When solving for **R** and **T**, measurement noise may result in values being obtained for **R** that do not properly conform to the requirements of a rigid body rotation. In this case an approximate rotation matrix can be best fitted to the data (Zhang [1998]).

4.3.13 Dealing with lens distortion

Digital cam-corders exhibit significant radial lens distortion (barrelling) which can be corrected by shifting pixels in the distorted image as a function of their radial distance from the optical axis.

Let (u,v) and $(\hat{u},\hat{v})$ be the pixel co-ordinates on a true pinhole image and a radially distorted image respectively. For each pixel in the image, the degree of distortion is related to the radial distance r of that pixel from the principal point. The following distortion model was adopted which is that typically adopted in the literature (Zhang [1998]):

$$\hat{u} = u + (u - u_0)(k_1 r^2 + k_2 r^4)$$ **Equation 4.29**

and $$\hat{v} = v + (v - v_0)(k_1 r^2 + k_2 r^4)$$ **Equation 4.30**

where $r^2 = (u - u_0)^2 + (v - v_0)^2$

Initially the distortion parameters k_1 and k_2 are approximated to zero. Optimum values are then computed by iteratively refining the distortion parameters whilst simultaneously refining the camera intrinsics, extrinsics and target relations transformations using non-linear optimisation (see next section 4.3.14).

4.3.14 Refining parameters with non-linear optimisation

Once initial estimates of camera parameters have been extracted using geometrical and analytical principles, it is possible to mutually refine these parameters by a method of non-linear minimisation of an error function, resulting in a maximum likelihood estimate for all parameters.

Given initial estimates of intrinsic and extrinsic parameters, radial distortion parameters and target relations transformations, the following error function may be minimised:

$$\sum_{\text{target } t=1}^{n} \sum_{\text{spot } s=1}^{m} \left\| \mathbf{x}_{image_{ts}} - \hat{\mathbf{x}}_{image_{ts}}\left(\mathbf{C}, k_1, k_2, \mathbf{R}_t, \mathbf{T}_t, \mathbf{X}_{target_{ts}}\right) \right\|^2 \qquad \textbf{Equation 4.31}$$

Where, for m points (spot centres) extracted from n target views, $\mathbf{x}_{image_{ts}}$ is the observed image in pixelated camera co-ordinates of the world co-ordinate target point $\mathbf{X}_{target_{ts}}$, and $\hat{\mathbf{x}}_{image_{ts}}$ is the expected image of that point given the current estimates of the camera parameters $(\mathbf{C}, k_1, k_2, \mathbf{R}_t, \mathbf{T}_t)$. Note that the values of the co-ordinates of $\mathbf{X}_{target_{ts}}$ are also dependent on the current estimates of target relations transformations and these transformations are also being iteratively refined.

These non-linear minimisation problems may be solved using a standard non-linear optimisation strategy. In this case Powell's method was used. The error function to be minimised is the sum of the squares of the discrepancies between predicted and observed spot positions over the set of calibration images.

There are many algorithms (Press [1992], Nocedal [1999], Hartley [2000]) that can be used for performing non-linear optimisation of a function in multiple dimensions. Different strategies achieve different trade-offs between speed of convergence and robustness to local minima. Powell's method is an example of a "Direction Set" method. It works by choosing an optimal direction. The function to be minimised is then minimised along a line in this optimal direction before a new direction is chosen. Although alternative algorithms might have been used, Powell's method is well established, robust and rapid. It was also convenient since pre-written code for this algorithm was available within the research group.

4.4 Results of data set construction

4.4.1 The extracted trajectory

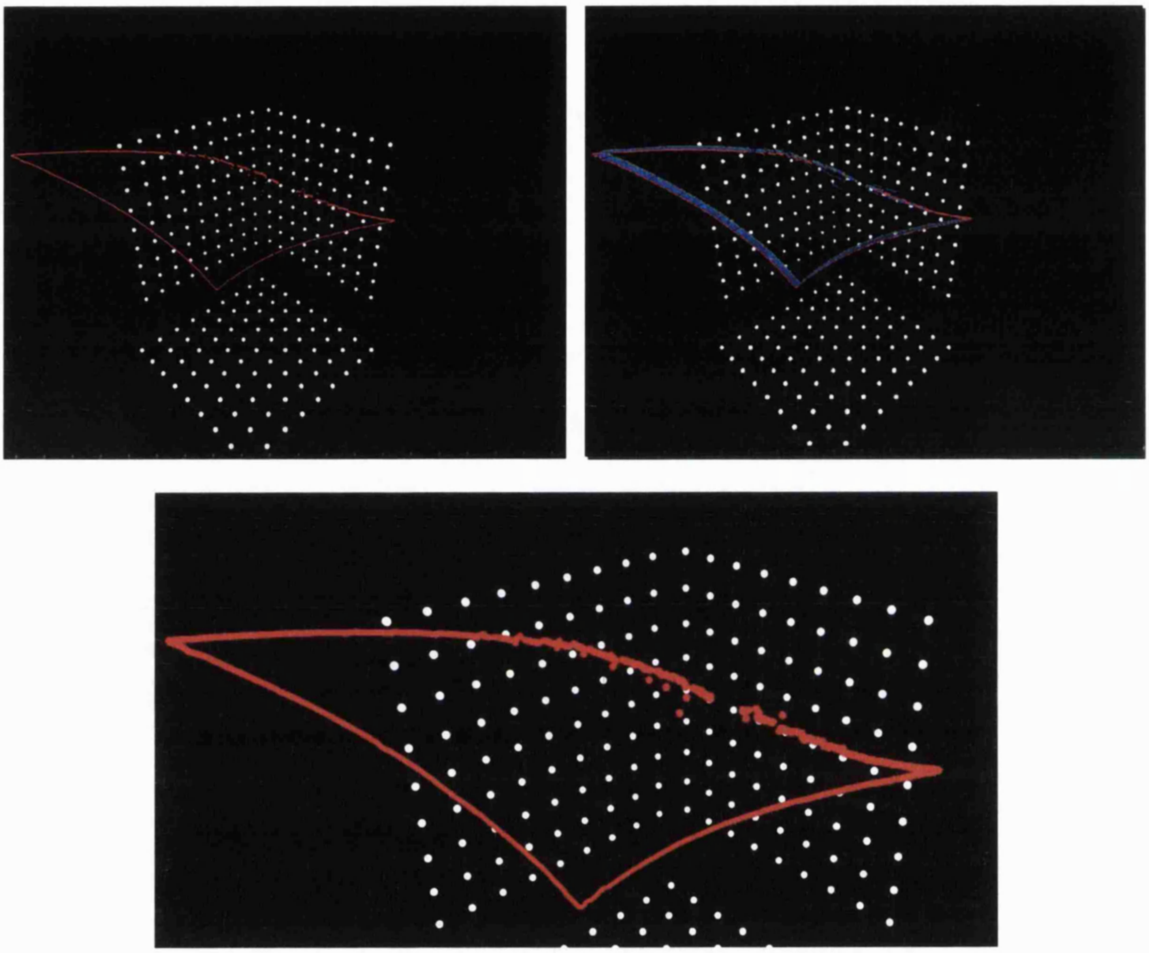


Figure 4.23 **The computed trajectory for a six-degree of freedom motion video sequence. The camera position at each frame is illustrated by a small red sphere.**

The trajectory is illustrated in relation to the spots of the calibration target structure.

Top right also illustrates the orientation of the camera. For each frame the camera is located at the red dot and looks along the blue line towards the green dot.

Bottom image shows an enlarged portion of top left.

A six-degree of freedom motion, incorporating both smoothly curving and sharp cornered trajectory segments, was programmed into the PUMA 560 robot arm. Video sequences were filmed along this trajectory showing various objects in good visibility and varying degrees of poor visibility. Calibration sequences, viewing calibration targets, were filmed along this trajectory. A sample of each type of sequence was chosen such that all the sequences were synchronised to within ± 1 pixel when comparing synchronisation spots. The calibration sequence was analysed as described in section 4.3, yielding a list of camera positions and orientations for every frame, intrinsic camera parameters and lens distortion parameters. This trajectory is summarised in the images of figure 4.23. The specific calibration data measured for this image sequence is appended at the end of the thesis.

4.4.2 Smoothness of the trajectory

The trajectory plots of figure 4.23 are a useful visual representation of the complex six degree of freedom motion video sequence that has been analysed. The trajectory appears to be smooth and this consistency implies a high degree of positional accuracy.

It is apparent (see bottom image, figure 4.23) that one section of the trajectory appears broken, erratic and non-smooth. This section corresponds to the beginning and end of the trajectory. During these portions of motion, the camera is moved from (and back to) a position fixated on the synchronisation spot to a more central position regarding the main areas of calibration target. For this reason, during these portions of motion, comparatively few calibration spots are in the field of view. This results in a sparse set of point correspondences with which to triangulate the position of the camera, leading to inaccurate measurements. These portions of the camera motion do not correspond to visually interesting portions of the video sequence and are not needed for the purposes of testing vision algorithms. The only use for these beginning and end sections is to enable the use of the synchronisation spot for determining chronological correspondence between matching video sequences.

In order to quantify smoothness and to assess any apparent discrepancies in the computed motion, the rotational and translational components of the motion were plotted (see figure 4.25). These plots indicate the trajectory measurements to be smooth and consistent. Making the assumption that deviations from smoothness equate to measurement noise gives an estimate of positional measurement accuracy that approaches the mechanical limits of the robot itself, i.e. of the order of ±0.2mm.

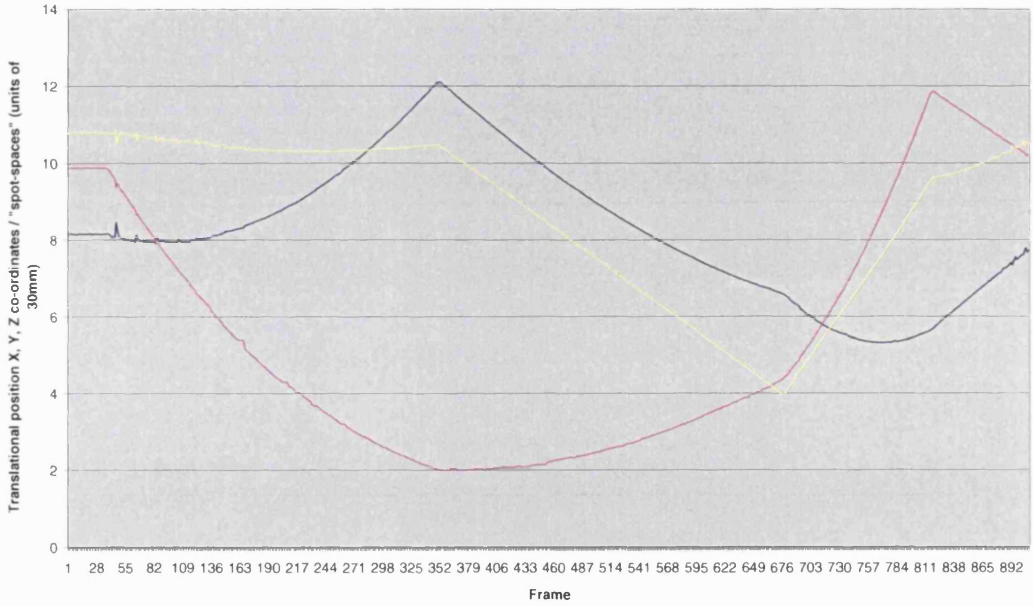


Figure 4.24 **X,Y and Z components (blue, red, yellow respectively) of camera motion in world co-ordinate system (relative to base target origin).**

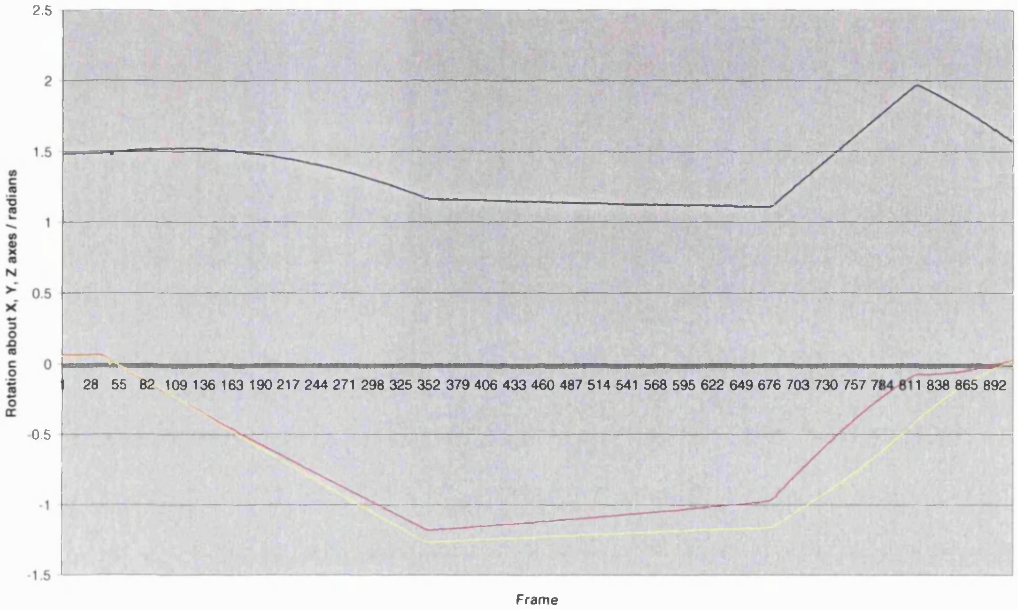


Figure 4.25 **Rotational components about X, Y and Z axes of camera motion in world co-ordinate system (relative to base target origin).**

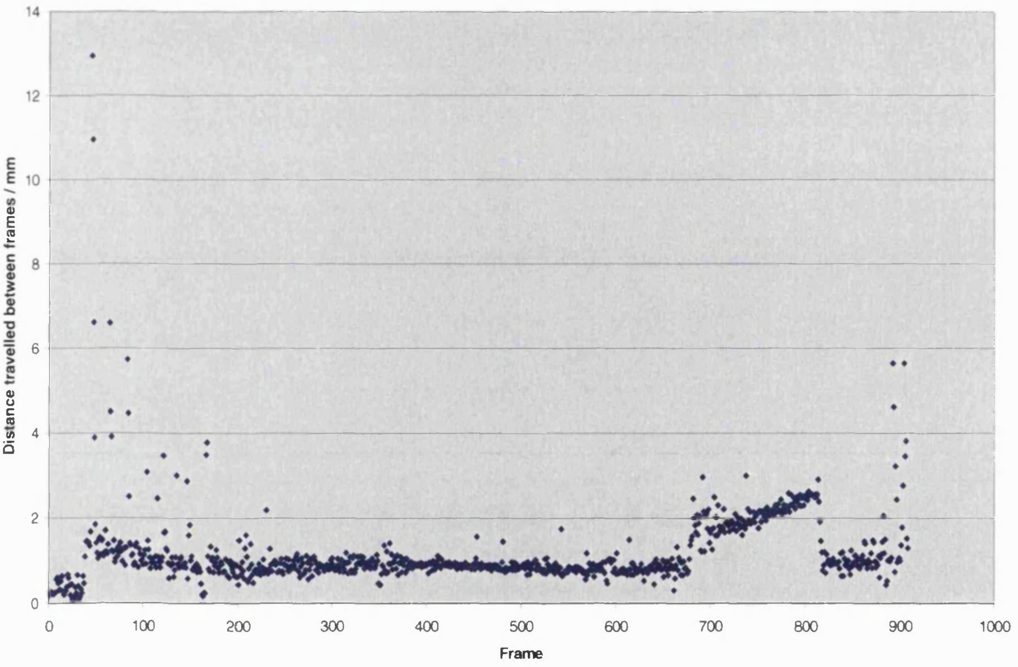


Figure 4.26 **Total distance moved by the camera from one image to the next. This gives some indication of "jerkiness" in the estimated trajectory. Any percieved jerkiness is, at worst, attributable to trajectory measurement noise, but may in fact be genuine jerkiness in the robot motion.**

There are obvious large discrepancies at the beginning and end of the motion due to sparsity of spots when moving to and from the synchronisation spot. In the central, smooth portion of the motion, the movement is consistentlv around 1mm between each frame.

4.4.3 Error associated with video sequence synchronisation

There is an obvious source of error associated with the synchronisation procedure. Clearly two video sequences can only be matched to the nearest frame. Since it is not possible to synchronise the camera with the motion of the robot arm, there will be no frame in the test sequence which occurs at *exactly* the same time during the motion as any corresponding frame in the calibration sequence.

At worst, frames in the test sequence will occur, chronologically, exactly halfway between frames in the calibration sequence. This will cause a synchronisation error of half a frame period. At 25 frames per second this results in ± 0.02 seconds.

How significant is this error? Ultimately we are not concerned with temporal error, but rather in errors in the extracted camera co-ordinates for each frame i.e. position and orientation errors. In this case, position and orientation errors are caused by temporal errors in synchronising test sequences with calibration sequences. The "real" error resulting from a worst case temporal error of ± 0.02 seconds is thus dependent on the speed of motion of the camera and robot arm. A high speed will lead to large errors and a low speed will result in small errors.

There are two main ways in which this synchronisation error can be reduced:

- Programming the robot to move at very slow speeds.
- Filming a large number of repeats of each video sequence.

Since the degree of temporal overlap between any two sequences is dependent on when the camera was switched on, i.e. random, a large number of samples of each sequence increases the probability of finding one pair of sequences (test and calibration) that form a good match. For example, if ten examples of each sequence are filmed, there are now one hundred possible pairs of test sequence and calibration sequence. If these are distributed randomly in time, we might expect the synchronisation error to be reduced by a factor of one hundred on average.

In practice, it was possible to synchronise the two sequences such that when corresponding frames were superimposed, the "visual match" error was often less than ± one pixel.

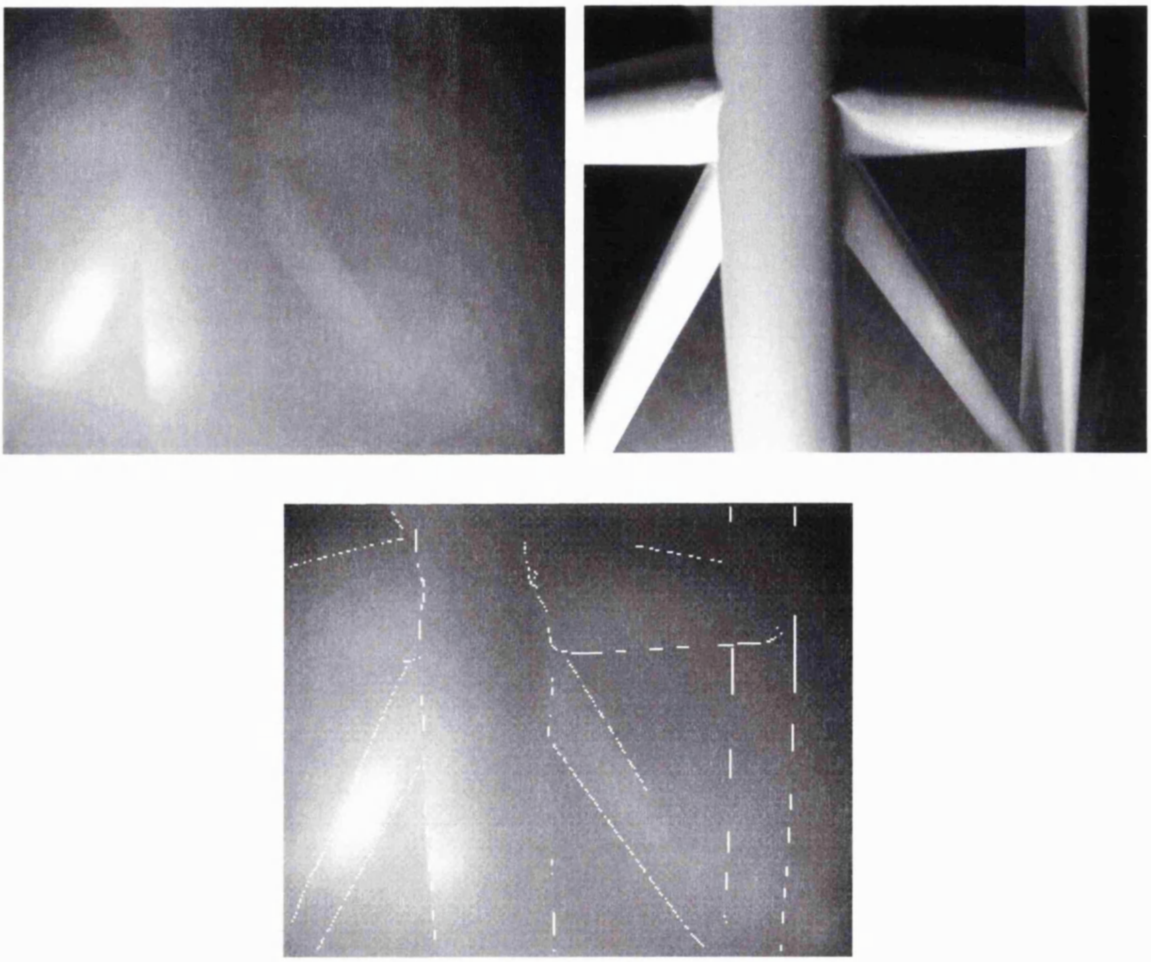


Figure 4.27 **Corresponding images from poor visibility sequence (top left) and good visibility sequence (top right). Bottom image shows an edge detected version of the good visibility image superimposed over the poor visibility image.**

Note that the edges of the object being viewed match up exactly (to within the accuracy of the image capture technology i.e. to the nearest pixel), indicating a high level of precision in the synchronisation of these two sequences.

4.4.4 Error associated with robot motion

Industrial robot arms are highly *repeatable.* This means that the same motion can be performed many times with the end-effector (in this case the camera mounted on the terminal link of the arm) returning to the same position with a high degree of precision each time.

Note that *repeatability* is not the same thing as *accuracy*, which is normally taken to mean the degree of correspondence between programmed position co-ordinates and the actual positions achieved. In general, the *repeatability* of an industrial robot will be several orders of magnitude better than its *accuracy.*

A simple test of robot repeatability was performed as follows:

- Mount the camera securely on the robot.
- Position the robot such that a static real world feature (e.g. set of calibration spots) is visible in the field of view.
- Set the camera running.
- Run the robot along a varied, six-degree of freedom motion that includes pauses at three different positions during the motion.
- Film whilst performing this motion several times.
- Compare the three pause images in one sequence with the pause images in a later sequence by superimposing and differencing.

These tests reveal excellent repeatability in the PUMA 560 robot. Superimposing the images reveals a barely visually discernible error of better than ± one pixel. This implies that errors associated with robot repeatability are so small that they approach the scale of the noise associated with the camera itself.

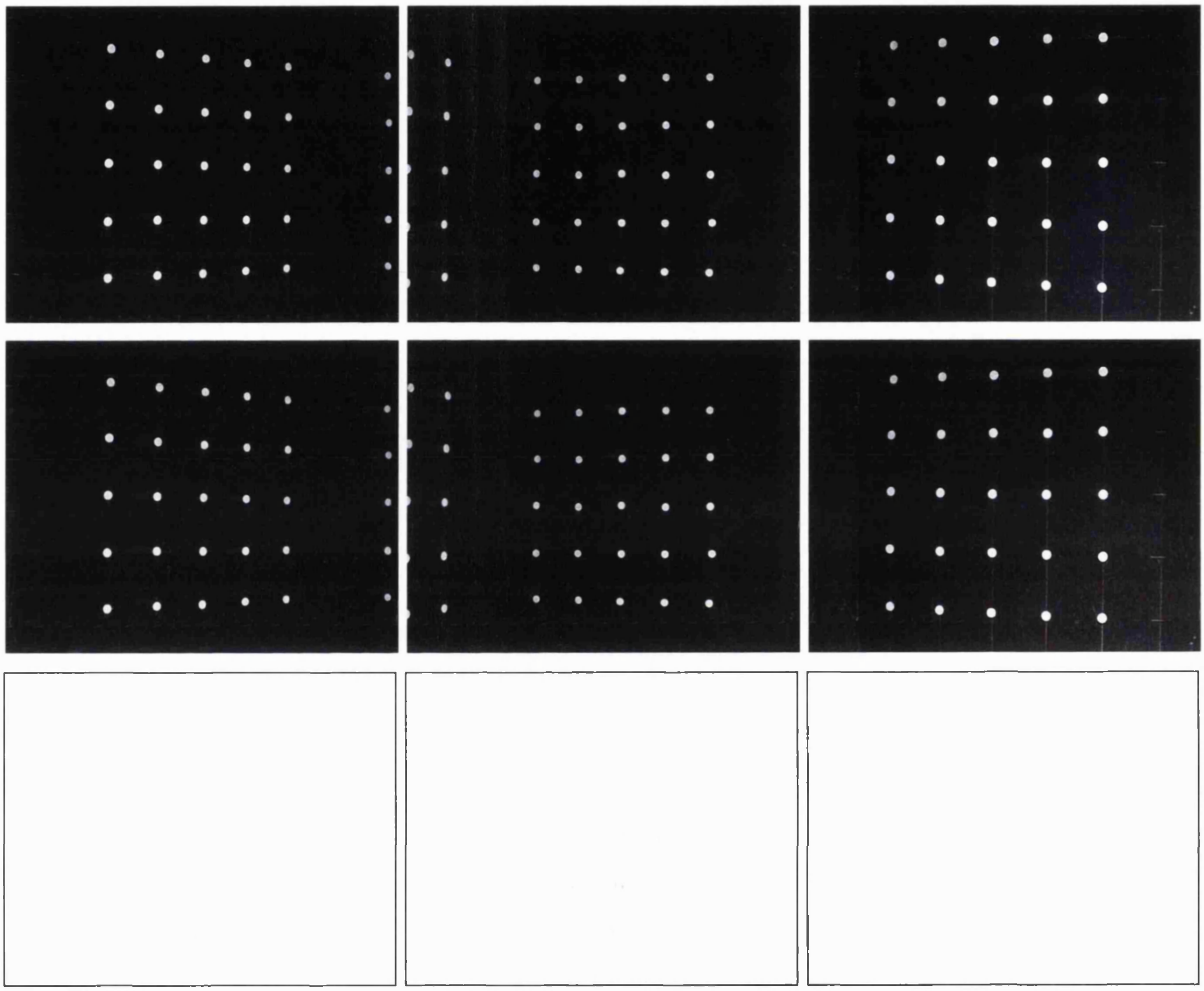


Figure 4.28 Three different pauses along a trajectory.

The top row and second row are taken from two different video sequences filmed along the same trajectory.

The bottom row is the difference between corresponding images of the same pause position taken from the two sequences. The difference image has been negativised to improve visibility of the very faint features.

White negative difference images would imply no difference between the two images being compared i.e. an exact correspondence (to the nearest pixel). These images appear to match with an accuracy better than ± one pixel. Very small discrepancies can just be distinguished. These may be due to differences in lighting causing spots to appear bigger or smaller rather than true robot position differences.

4.4.5 Error associated with computer based data analysis

The procedure used to assess error during the calibration and trajectory measurement process was as follows:

- The measured (extracted from the calibration process) camera position and orientation, lens distortion parameters and intrinsic camera parameters were used to create a camera model through which world co-ordinate points could be projected onto predicted points in the image plane for each image being considered.

- A model of the calibration target structure was constructed, listing the position of every spot with respect to co-ordinate systems in the respective targets to which the spots belonged.

- Knowledge of the position and orientation of each target with respect to the base target was used to compute the world co-ordinates (relative to the base target) of every spot over the complete set of all three targets.

- Every spot position (world co-ordinates) was then projected through the camera model to create a corresponding set of expected image plane spot centroid positions.

- These were compared to the list of spot centroids generated by the feature detection stage of the calibration procedure (see figure 5.19). A root mean square error was calculated, representing the average discrepancy (over that image) between measured (from the feature extraction process) spot centroid positions and expected spot positions given the computed camera model.

Typical error values were 0.6 pixels rms error per spot. There are several reasons why these errors might occur:

- Centroids of image spots may not be truly representative of the true spot centres.
  - In general a circular spot projects as a distorted ellipsoid in an image. The centroid of this shape is rarely the same as the true spot centre.
  - Some spots may lie in shadow such that an off-centred portion of the spot is detected during the image thresholding process.

- Camera intrinsic parameters, lens distortion parameters, and camera position and orientation parameters are refined over an error space using a non-linear optimisation process (Powell's method). This process may have converged on a local minimum of the error space i.e. there may be some better values of the camera parameters that would result in a better fit to the observed data.

- The camera model may be over-constrained, i.e. certain aspects of the camera may not have been modelled. For example, Zhang [1998] includes a parameter in the camera model which represents skewness in the pixel array (i.e. horizontal and vertical lines of pixels may not be exactly perpendicular). In this work, such skewness was ignored, with the camera pixel array assumed to be perfectly perpendicular.

4.4.6 Visualising the overall error

An obvious way to inspect the accuracy of the measured trajectory is to reconstruct the images that would be generated by the camera moving along that trajectory. These images can then be compared (by super-imposing corresponding images) to those of the real video sequences filmed along that trajectory.

The "synthetic" video sequences are generated by creating a model (in world co-ordinates) of the object viewed in the real sequence (e.g. block object or oil-rig object). This model is then projected according to the measured camera characteristics (position and orientation, intrinsic parameters, lens distortion parameters). Methods for modelling and projecting the viewed objects are detailed in section 3.10.

An alternative method for projecting "synthetic" images was to use the ray tracing software package "POV-ray for Windows" (http://www.povray.org). This allows an object to be built at a fixed location in a world co-ordinate system. A camera can then be introduced at a desired position and orientation. Since POV-ray does not permit camera distortion to be modelled, images had to be distorted after they were created. The forms of distortion that were applied included:

- Vertical stretch (due to rectangular rather than square pixel aspect ratio).
- Principal point shifted away from the image centre.
- Radial lens distortion.

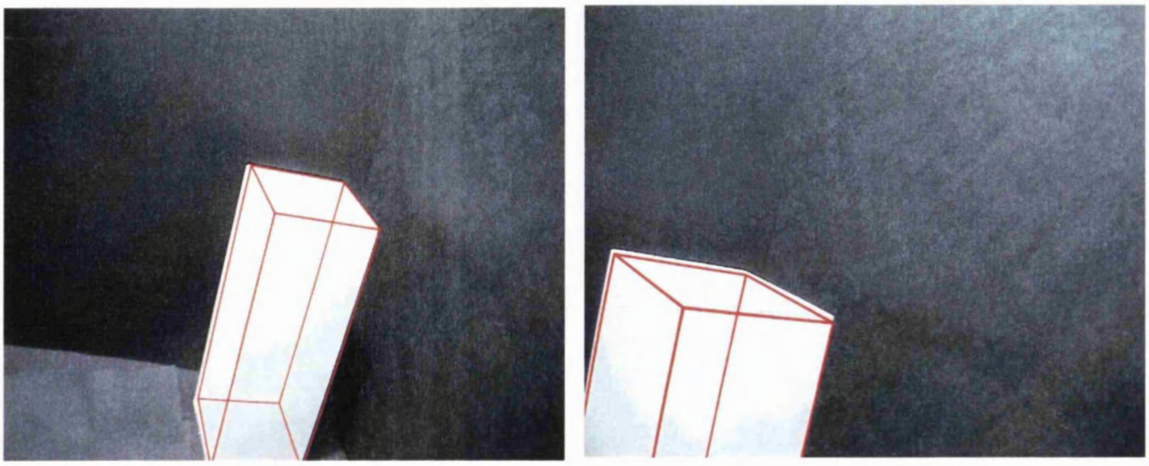


Figure 4.29 **Two images from the good visibility "block object" video sequence. In each case, the measured camera position for the frame has been used to project a predicted image (shown as a red wire frame) and this predicted image has been superimposed over the real image. This helps illustrate the errors involved (in this case ± 3 pixels discrepancy in block edges).**

4.4.7 Disparities between predicted error and actual error

The observed error (section 4.4.6) is an order of magnitude larger than what would be expected, given the errors measured during the calibration and trajectory measurement process (section 4.4.5).

When the image positions of calibration target spots were reconstructed (section 4.4.5) and these expected spot positions were compared with those observed, the typical root mean square error was 0.6 pixels per spot. In contrast, when expected images of objects (block, oil-rig etc) have been projected and compared to the corresponding real images, an error of several pixels is observed.

There are several possible explanations for this discrepancy including:

- Since the observed objects are at different ranges from the camera than the calibration spots, camera position errors that project small errors in image spot position may also project relatively large errors in image object position.

- Centroids of image spots may not be precisely representative of the true spot centres.
    - In general a circular spot projects as a distorted ellipsoid in an image. The centroid of this shape is rarely the same as the true spot centre projection.
    - Some spots may lie in shadow such that an off-centred portion of the spot is detected during the image thresholding process.

- The camera model and camera location were optimised to best fit the observed location of spots in calibration targets. These spots were all co-planar. It may be that the camera position was over-fitted to known points at a particular range or in a particular plane and correspondingly under-fitted to any points in space outside of the target planes (e.g. corners of the block object).

- The objects being viewed are considerably smaller than the space covered by the calibration spots. Thus, when the camera moves in closer to the relatively small space occupied by the objects, errors may be magnified relative to the views from the calibration set of images.

- In many image frames, only a single calibration target was viewed. In theory (Zhang [1998]), only a single target view is necessary to uniquely locate a calibrated camera, however, in practice, degrees of freedom are introduced because some small camera rotations have a similar effect on the image as some small translations. Thus, for each true camera position it may be possible to find a combination of small translational and rotational errors which leaves the calibration image almost unchanged, but shifts the projected position of objects placed in front of the calibration spots. If, instead, three significantly non-co-planar targets are viewed, this small amount of freedom can be constrained.

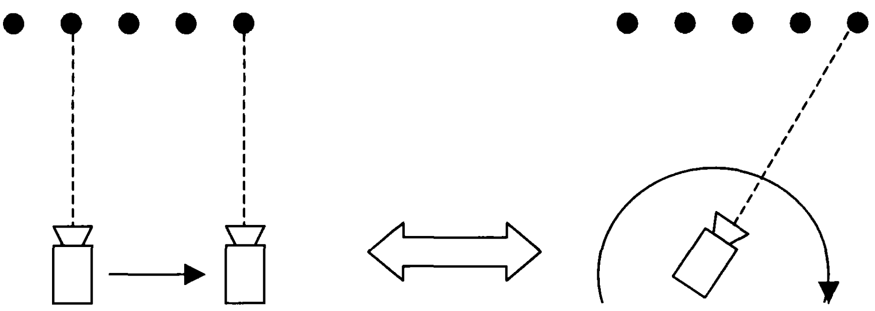


Figure 4.30 **Equivalent effect on observed image of a rotation and a translation. In the figure these movements are exaggerated, however if the translation is small and the range of the camera from the target is large, then the corresponding small rotation will leave the camera remaining approximately perpendicular to the target.**

4.4.8 How the data capture procedure might be improved

In response to the problems discussed in section 4.4.6, several suggestions arise as to how the experimental procedure should be changed in order to produce more accurate results in the future:

- The robot trajectory should be programmed such that the camera has a good view of all three targets in every frame. This constrains the problem of translation/rotation equivalence.

- The camera should be calibrated from a set of images filmed at a variety of different ranges from the targets. This prevents over-fitting to points lying in the target planes and under-fitting to points in the space outside of those planes.

- The objects to be viewed should be constructed such that they largely fill the volume of space within the calibration targets. This prevents error magnification when the camera moves in from a wide view of the targets to a close range view of a relatively small object.

4.4.9 A suggested technique for assessing accuracy

The main contribution of this thesis is to develop vision algorithms for interpreting poor visibility video sequences by combining observed and predicted data. Several steps (4.4.2-4.4.7) have been taken to assess the accuracy of the data set construction procedure. These assessments are sufficient to enable the generated data sets to reasonably validate the performance of the vision algorithms that form the main contribution of this thesis. However, a more comprehensive and systematic approach for future researchers might be as follows:

- Create a computer model of some calibration targets featuring grids of spots.

- Create a computer model of a camera with known intrinsic parameters and lens distortion parameters.

- Create a trajectory (a smoothly varying list of camera positions and orientations).

- Project synthetic images of the calibration targets by placing the camera at each of the co-ordinates from the synthetic trajectory list.

- Feed the resulting images into the calibration and trajectory extraction process.

- Compare the output (measured trajectory) with the input (synthetic trajectory).

5 Results

5.1 Layout of this chapter

Sections 5.2 and 5.3 examine the detailed workings of the EM/E-MRF algorithm as it analyses a single image. An observed image is fed into the vision system along with an initial position estimate for that frame. The position estimate is based on the ground-truth camera position (see chapter 4) but contains a deliberate error in one of the co-ordinates. The performance of the algorithm is assessed for various different starting errors.

To aid understanding of the algorithm, section 5.2 describes the various steps contained within a single iteration of the Expectation Maximisation algorithm, during the analysis of a single image frame. Each stage of the algorithm is illustrated using partially processed images produced by that step. Section 5.3 examines the performance of the algorithm over multiple EM iterations, when subjected to various different degrees of error in the initial position estimate at that frame. Section 5.4 presents the results of attempting to track camera trajectories over image sequences containing large numbers of frames.

Although the entire filmed trajectory lasted over 800 frames, the vision algorithm has been tested on sequences of between 51 and 201 frames. These were selected from the total filmed footage as exhibiting appropriate levels of visibility-neither "good" nor impossibly bad.

Often an observed image frame is shown with a superimposed red outline. The red outline is derived (by edge detection) from the predicted image output by the EM/E-MRF algorithm (what it "thinks" it is seeing). Where poor visibility images are shown, linear contrast stretching has been performed to aid the reader.

Trajectories are illustrated with a 3D plot. The ground truth trajectory is shown in red and the algorithm output is shown in green. The positions of the calibration target spots (30mm spacing) are also shown to provide a visual reference frame.

5.2 Stages of the EM/E-MRF algorithm

5.2.1 Initial position estimate

The EM/E-MRF algorithm is intended for visual tracking during an extended video sequence filmed along a trajectory. Under these conditions, the initial position estimate at each frame is generated by extrapolating the prior trajectory of the camera (see section 3.8), however for the purposes of testing the algorithm on individual images, an initial position estimate containing a known error of varying severity is input to the system.

The data used to illustrate the stages of the EM/E-MRF algorithm in the following sections (5.2.1-5.2.4) are based on the first EM iteration of the sequence presented (later) in section 5.3.1, in which the initial camera position estimate input to the system contains a deliberate error of 28.4mm in the X co-ordinate (this figure is entirely arbitrary and arose initially from varying the ground truth co-ordinate by a factor of ten percent).

Position estimates are conveniently visualised by superimposing the outline of their corresponding predicted image (shown here in red) over the observed image (figure 5.2).

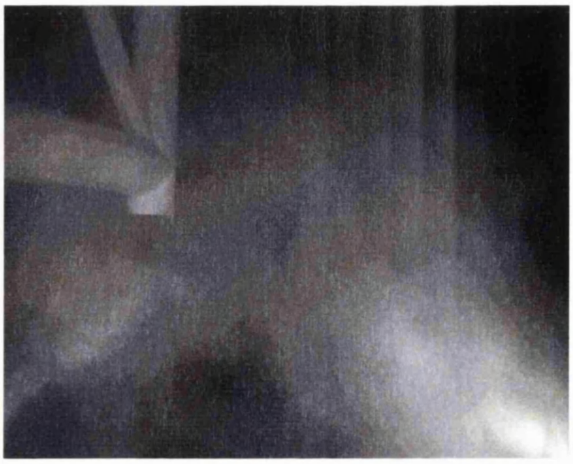


Figure 5.1 The observed image.

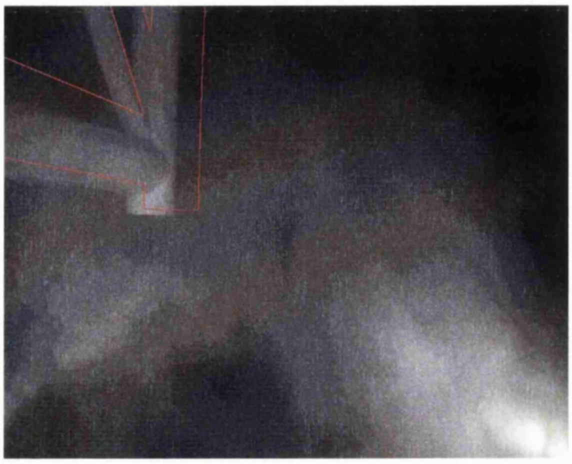


Figure 5.2 Erroneous initial position estimate.

5.2.2 Thresholding

The predicted image, based on the initial position estimate, is combined with pixel grey-level data from the observed image (see chapter 3, equation 3.13), to estimate class conditional probability density functions (modelled as normal distributions) for the two classes ("object" and "background"). The intersection of these distributions (figure 5.3) defines a discriminating value which is then used to threshold the observed image (figure 5.4).

The use of normal distributions as image models is justified in that the true image histograms (showing distribution of pixel grey-levels within object and background image regions) are often uni-modal and bell shaped. The true histograms for these class regions in the specimen image are also shown in figure 5.4 for comparison. Further work (section 6.5.1) will consider ways of modelling images for which the class conditional distributions are multi-modal and vary with position in the image.

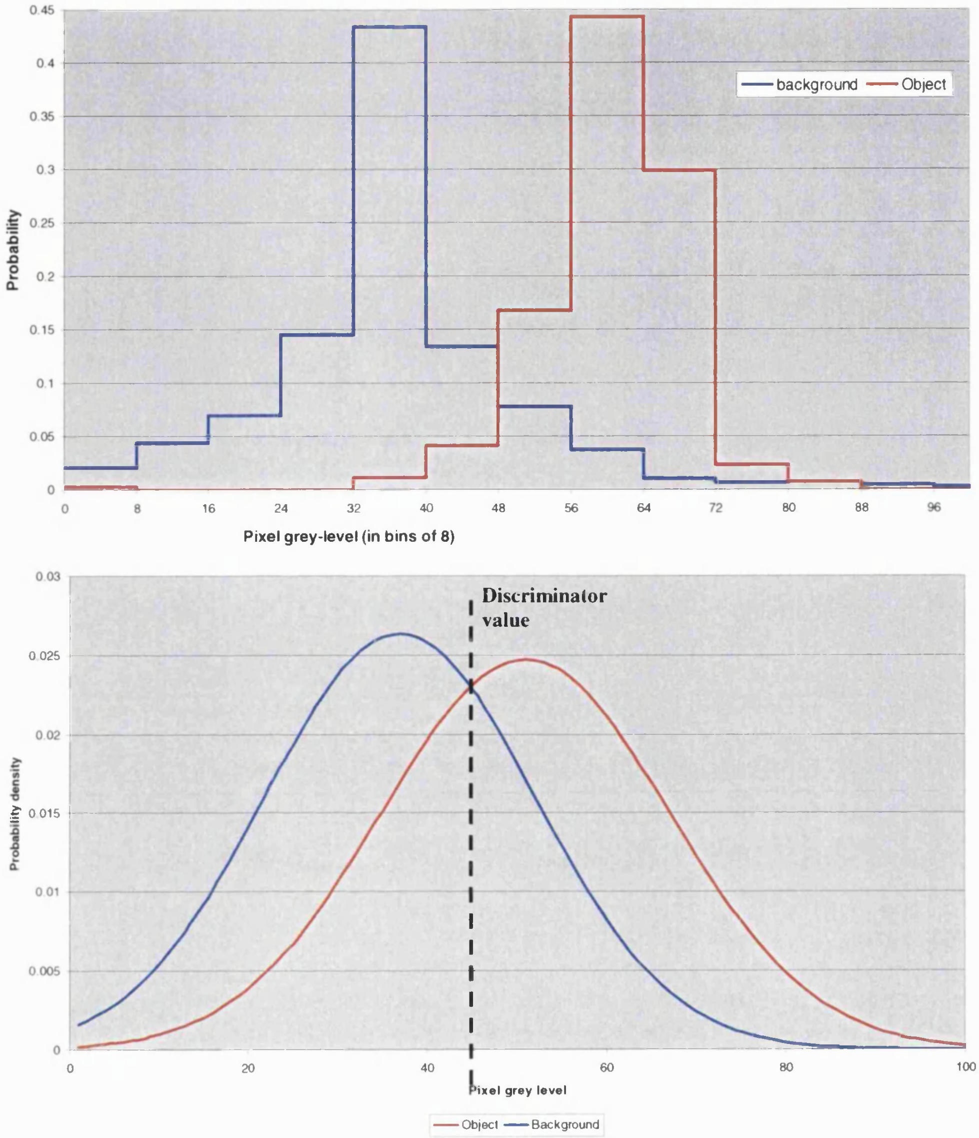


Figure 5.3 Top: true class conditional histograms i.e. distribution of pixel grey-levels within "true" object and background image regions. Bottom: estimated class conditional distributions and discriminating value.

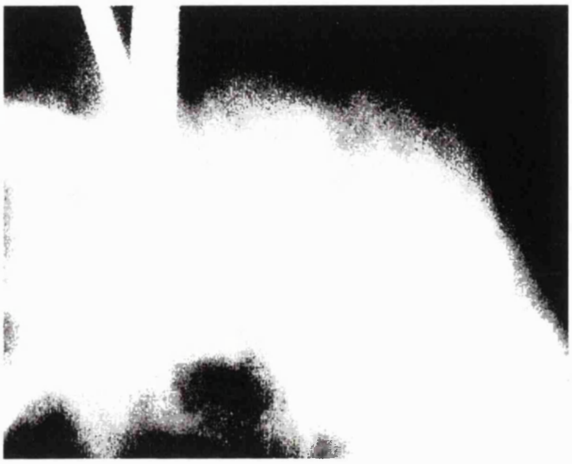


Figure 5.4 Thresholded image

5.2.3 E-MRF segmentation

The thresholded image (figure 5.4) is now used as the starting point for segmentation (figure 5.5) by Extended-Markov Random Field (see section 3.3). When assigning a class to each pixel, Markov dependency is extended to include a contribution from corresponding pixels in the predicted image as well as contributions from neighbouring pixels in the observed image. The advantages, in poor visibility, of the E-MRF compared to conventional MRF methods were investigated and demonstrated in Fairweather [1997a], however an example is included here (figure 5.6), showing comparative segmentation results by each method.

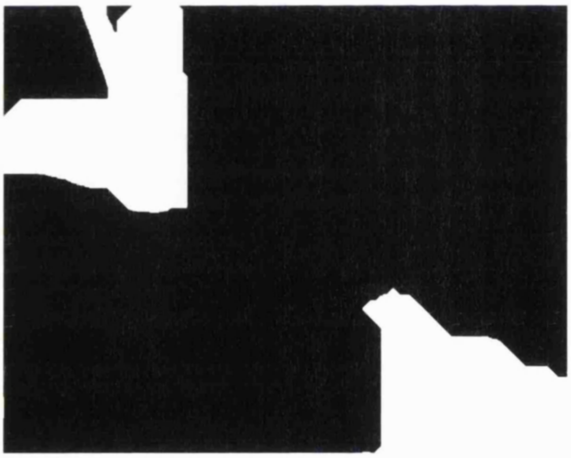


Figure 5.5 Segmentation by E-MRF (1^{st} EM iteration)

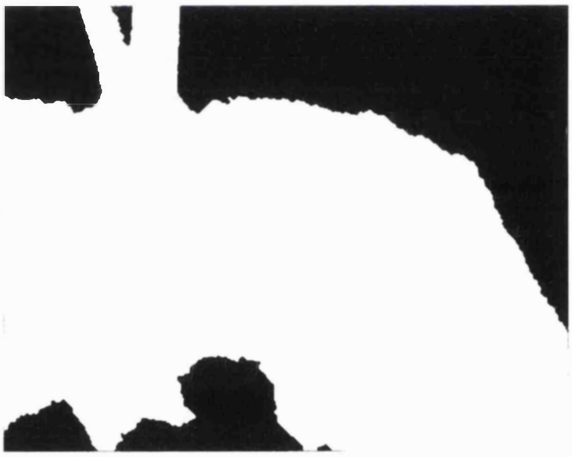


Figure 5.6 Conventional MRF

The quality of the segmented image is partly dependent on the accuracy of the predicted image. Consequently, E-MRF segmentation improves with each iteration of the Expectation Maximisation algorithm (figures 5.7 and 5.8). The following section (5.2.4) describes how the relative weightings between observed and predicted data are chosen during this segmentation process.

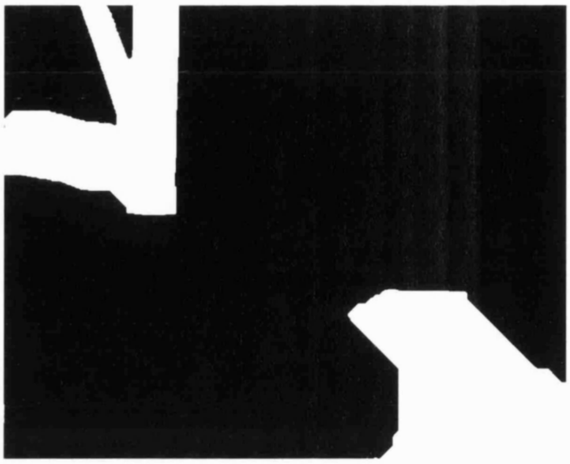


Figure 5.7 **Segmentation during 2nd EM iteration**

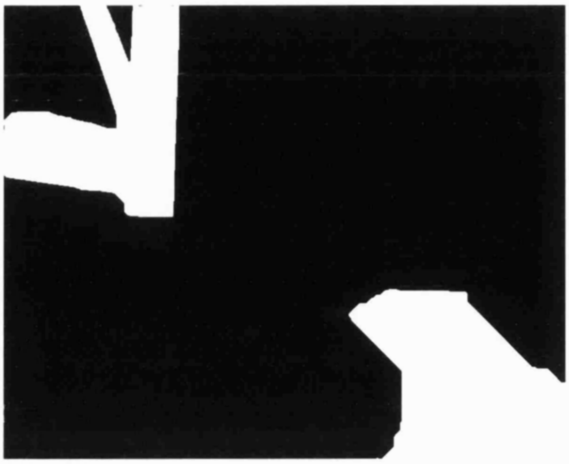


Figure 5.8 **Segmentation during 3rd EM iteration**

5.2.4 Choosing a weighting factor for predicted data

In chapter 3 (section 3.4) a negative log-likelihood function was developed (equation 3.14). This expression is reproduced here for convenience:

$$\sum_{m,n\in k} S_1[J(C_{i,j},C_{i+m,j+n})] + S_2[J(C_{i,j},\hat{C}_{i,j})] + \tfrac{1}{2}\log_e\left(\sigma^2_{C_{i,j}}\right) + \frac{\left(I_{i,j}-\mu_{C_{i,j}}\right)^2}{2\sigma^2_{C_{i,j}}}$$

Equation 3.14

S_1 and S_2 are weights which determine the significance (to the prior probability term) of the class values of nearest neighbour pixels and predicted pixels respectively. They thus effect the relative significance of observed and predicted data. It is not obvious how these values should be determined and other researchers (Fairweather [1997a], Dubes [1990], Bouthemy[1998], [1999]) suggest experimenting to find useful values for these constants by trial and error. Expression 3.14 consists of three parts, namely a class conditional component, the conventional MRF (spatial) prior probability component and a predictive prior probability component. S_1 was set to unity, which experimentally appears to produce similar magnitudes of contribution from the class conditional and spatial MRF terms. S_2 was then varied to find a prediction weighting that yielded a reasonable trade off between over-prediction and over-reliance on noisy observed data (figures 5.9-5.12). Values for S_2 of between 1.0 and 4.0 are often found to produce useful results. Note that the S_1 value, determined through trial and error based on a small number of test images, is likely to be sub-optimal and might profitably be investigated more thoroughly in further work (see section 6.5.1).

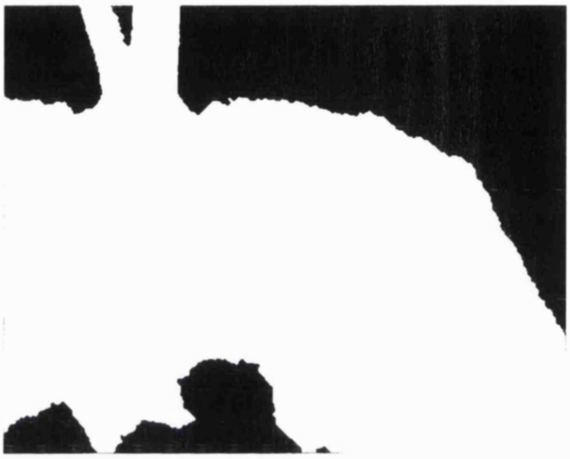


Figure 5.9 S_2 **= 0.0 (no prediction)**

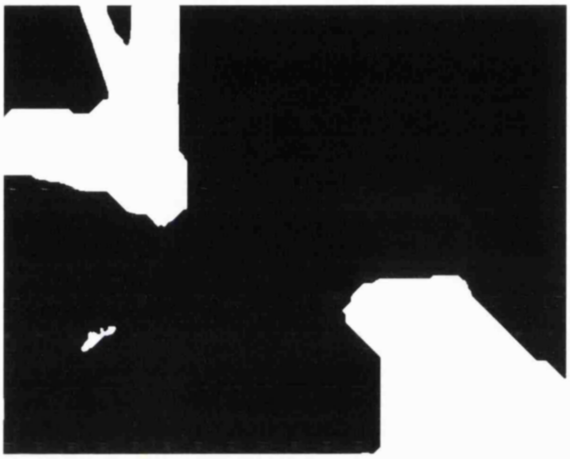


Figure 5.10 S_2 **= 1.0**

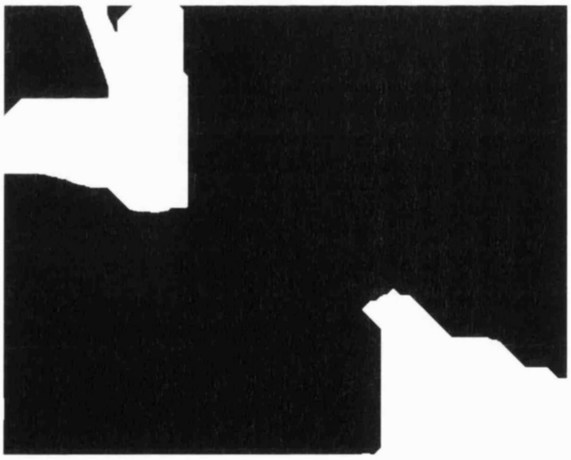


Figure 5.11 S_2 **= 1.75**

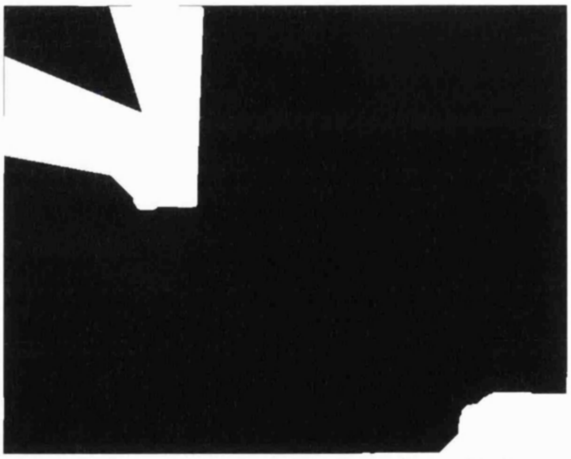


Figure 5.12 S_2 **= 4.0 (too much prediction)**

5.2.5 Position extraction by model fitting

In order to extract a new camera position estimate, successive projections of the object model are best fitted to the segmented image by means of a non-linear gradient ascent method. In this case, for proof of principle, Powell's method is used for convenience (see section 3.5), though alternative non-linear optimisation schemes may be better suited for real-time implementations (see section 6.3). The initial position estimate (figure 5.13) is used as a starting point for the optimisation process. The final position estimate (figure 5.14) is that which maximises the correlation (figure 5.15) between the predicted image, projected from that position estimate, and the segmented image.

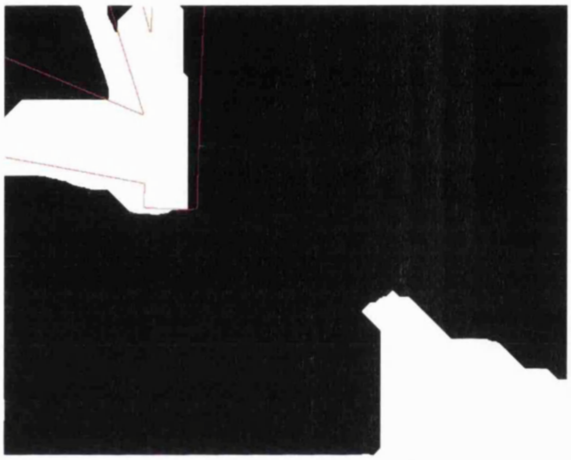


Figure 5.13 Initial position estimate

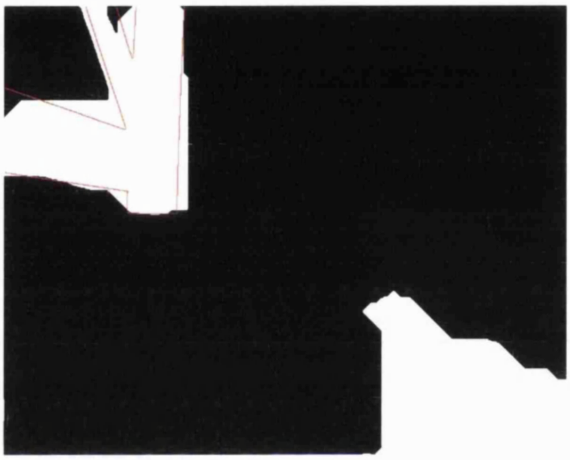


Figure 5.14 New position estimate after fitting model to segmented image

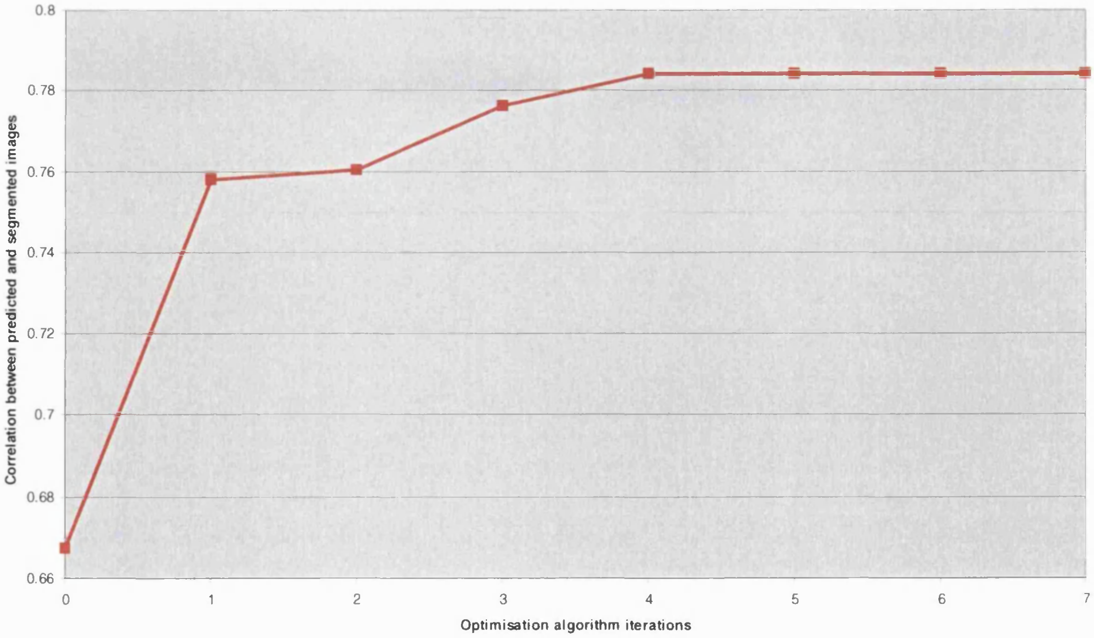


Figure 5.15 Improvement in correlation between predicted and segmented images during model fitting

5.3 Successive EM iterations with various starting errors

In the previous section, the various stages of a single EM iteration were examined in detail. The following sections demonstrate the behaviour of the algorithm during successive iterations. Different scenarios comprising various kinds of starting error are presented. It is interesting to note how the class conditional distributions change as the algorithm "homes in". This represents a form of machine learning since the algorithm is continually refining its models of both "background" and "object".

Typically the distributions will separate with successive iterations, with the "background" mean decreasing and the "object" mean increasing, as the algorithm progressively learns that object is bright and background is dark. In cases (see sections 5.3.3 and 5.3.5) where the initial position estimate is so bad that the algorithm will not converge on a better solution, the estimated class conditional distributions have very similar means, in other words "object" and "background" have become indistinguishable.

Translational errors are presented both as overall Euclidean distance between the ground-truth camera position and the position estimate output by the algorithm and also as the components of translational error in particular directions. An overall rotational error is calculated as the magnitude (in radians or degrees) of the rotation which would align the camera orientation output by the algorithm with that of the ground-truth. This difference, between algorithm estimate and ground-truth values for the camera position rotation matrix, is computed by multiplying one rotation matrix by the inverse of the other. The resulting rotation is then expressed (see section 3.8) as an axis of rotation and a magnitude (angle) of rotation about that axis. These errors are measured relative to a world co-ordinate system. It might also be useful to consider errors relative to a co-ordinate frame set in the camera and alternative ways of quantifying and presenting performance errors will be discussed under further work in chapter 6 (section 6.5.3).

5.3.1 Translational starting error of 28.4mm on x co-ordinate.

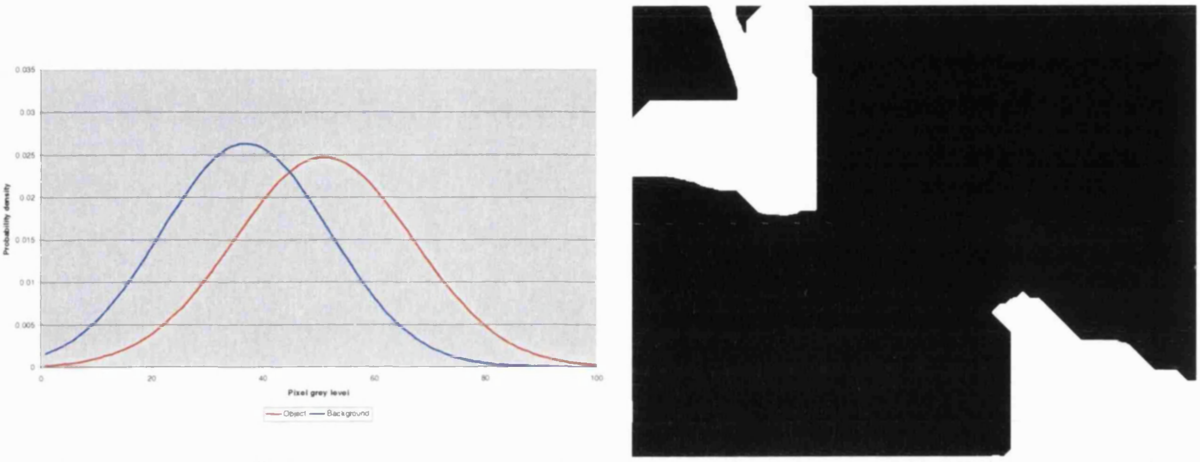


Figure 5.16 **1st EM iteration (class conditional distribution estimate and E-MRF segmentation)**

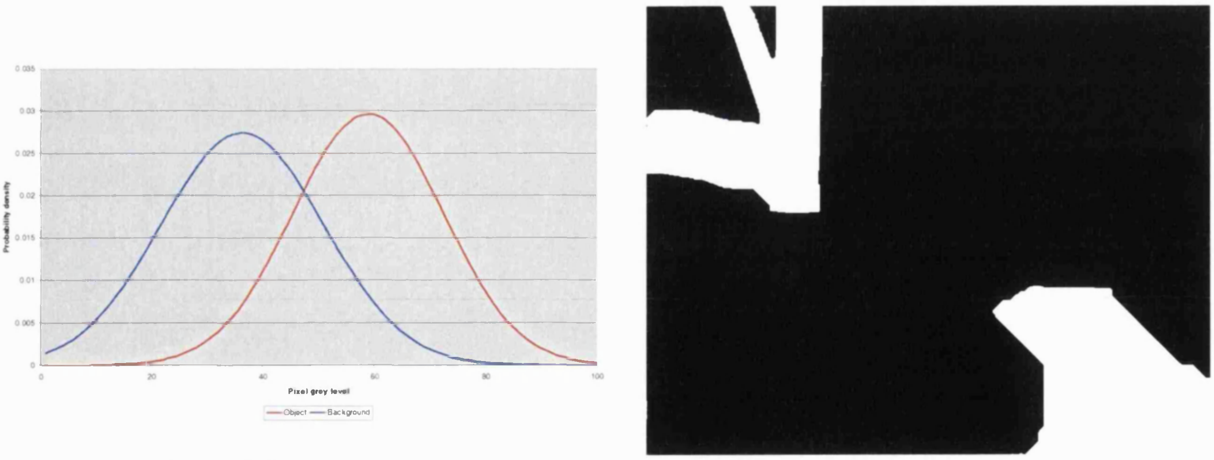


Figure 5.17 **2nd EM iteration**

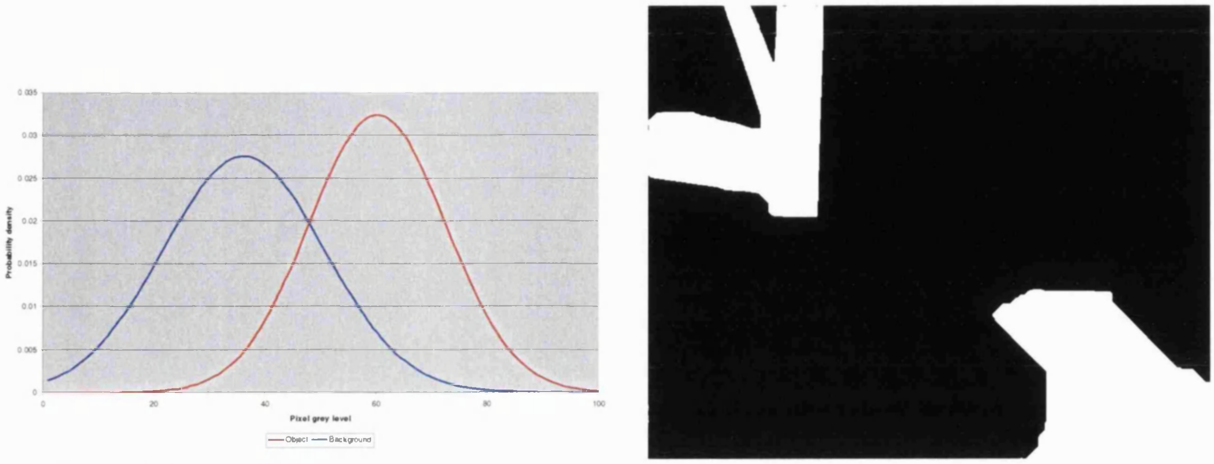
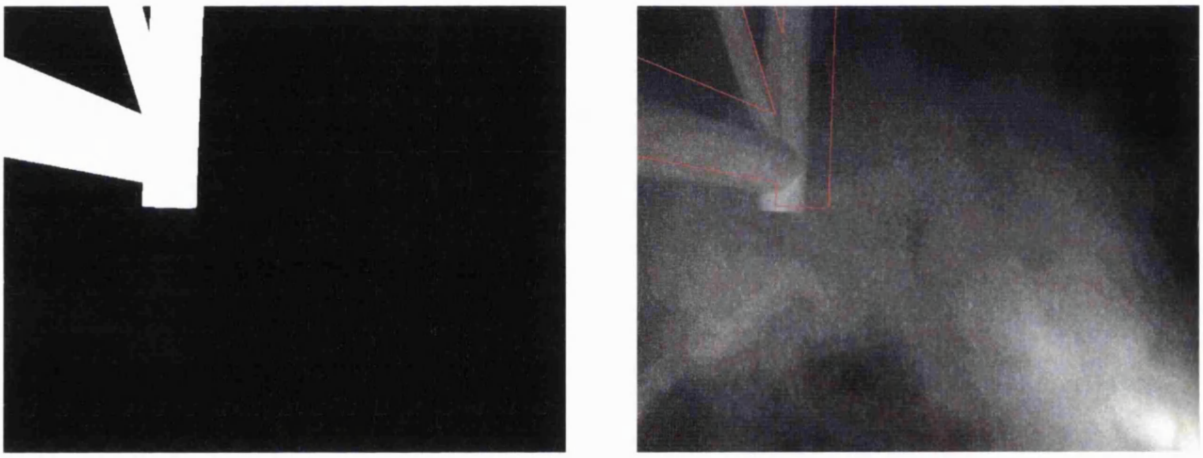
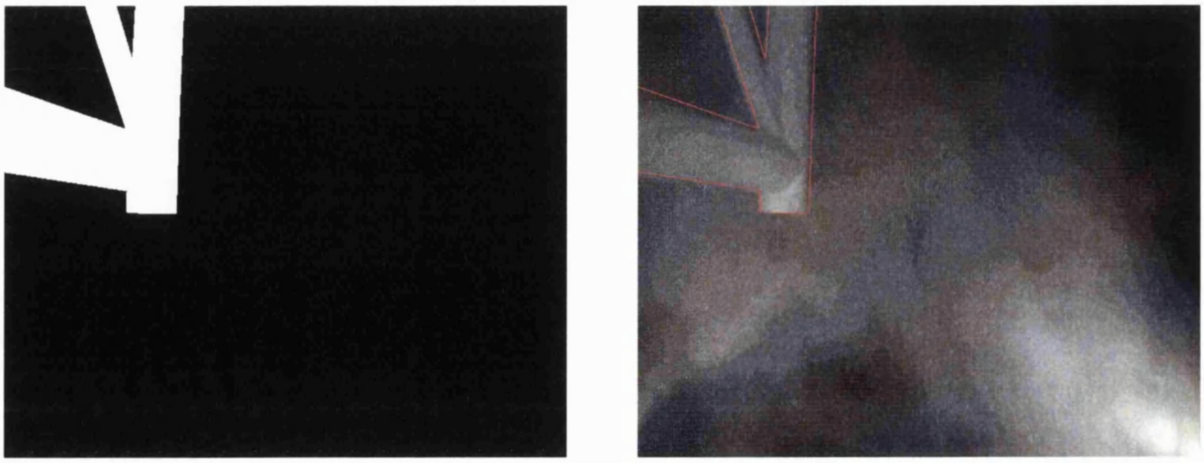


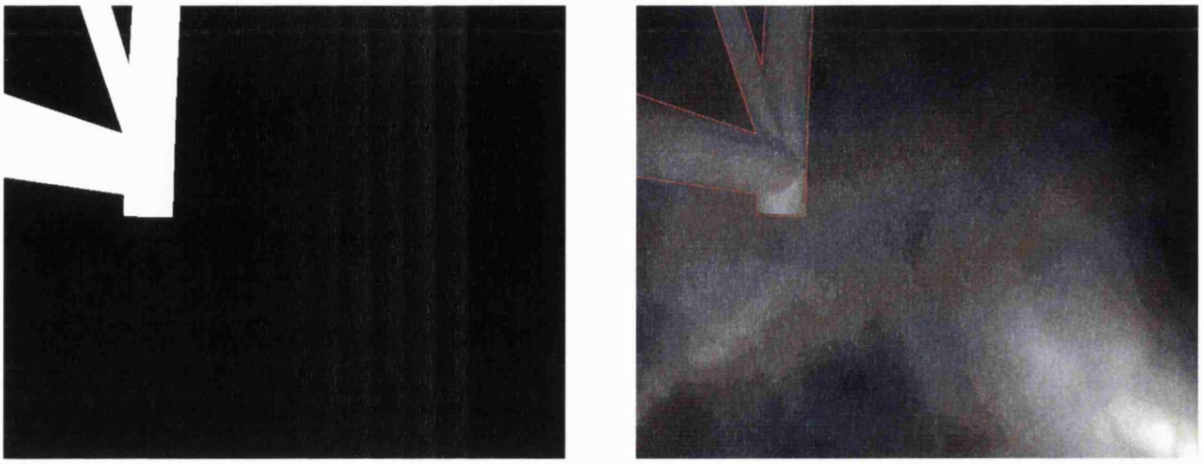
Figure 5.18 3rd EM iteration



1st EM iteration



2nd EM iteration



3^{rd} EM iteration

Figure 5.19 **Improvement of predicted image with successive EM iterations. The left hand image shows the predicted image based on the current camera position estimate. The right hand image shows the outline (in red) of this prediction, superimposed over the observed image.**

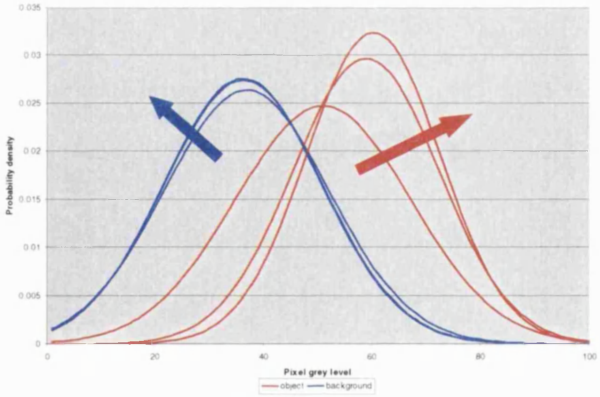


Figure 5.20 **Progressive re-learning of class conditional distributions with successive iterations.**

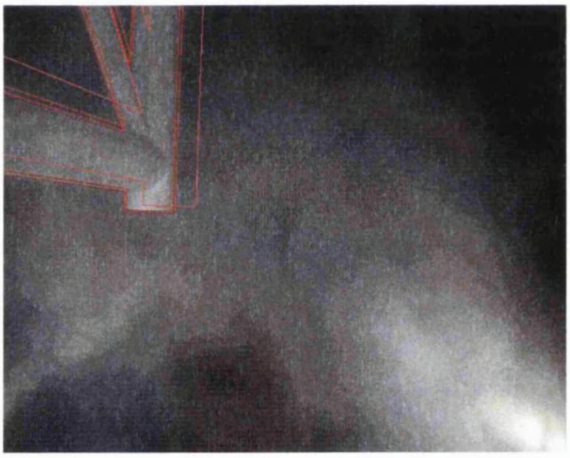


Figure 5.21 **Improvement in position estimate with successive iterations.**

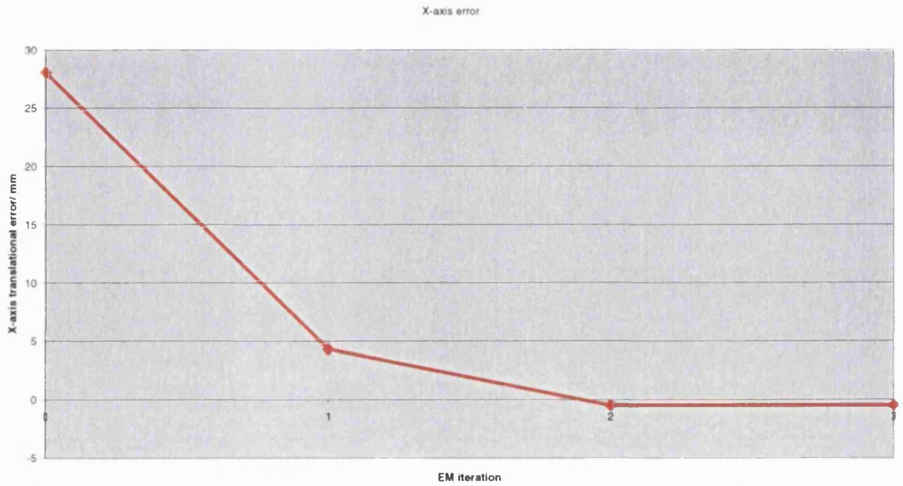


Figure 5.22 Decrease in x co-ordinate error with successive EM iterations. The initial position estimate for this image was the ground-truth position plus a 28.4 mm x co-ordinate error.

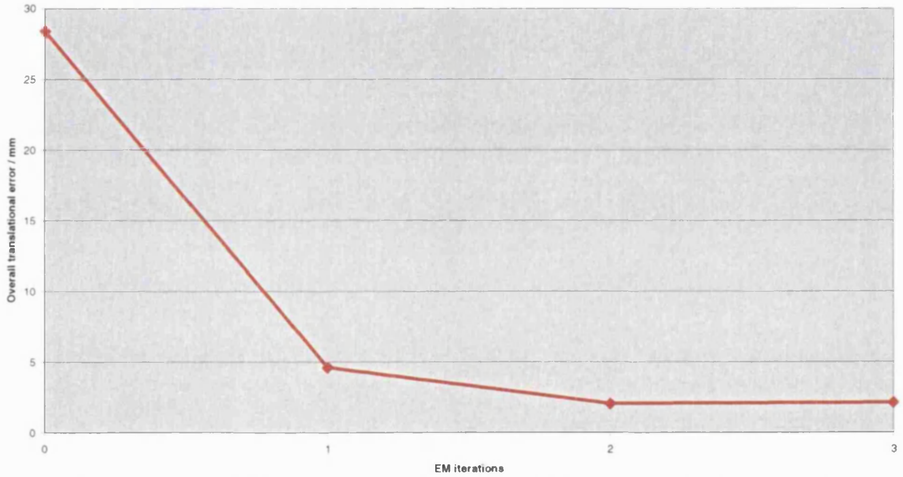


Figure 5.23 Decrease in overall translational error with EM iterations.

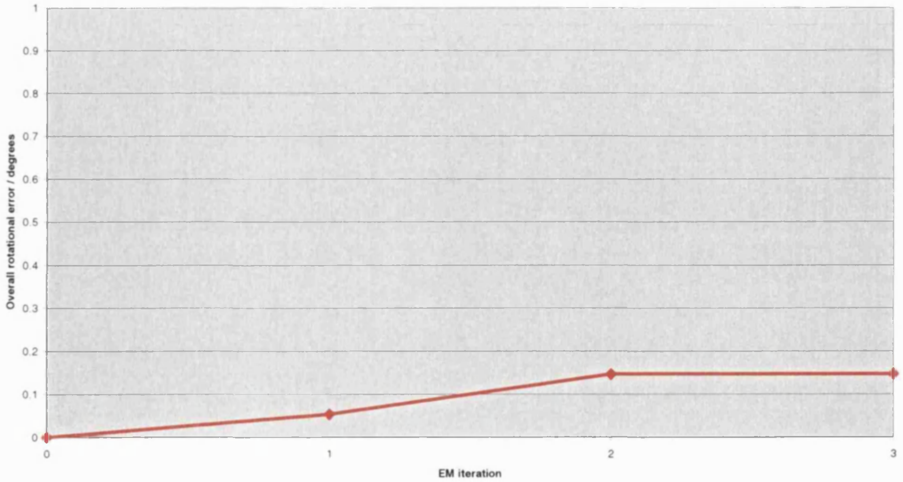


Figure 5.24 Variation of overall rotational error with EM iterations. The rotational error actually increases-but by a very small amount (from zero to 0.147 degrees).

5.3.2 Translational starting error of 56.8mm on x co-ordinate

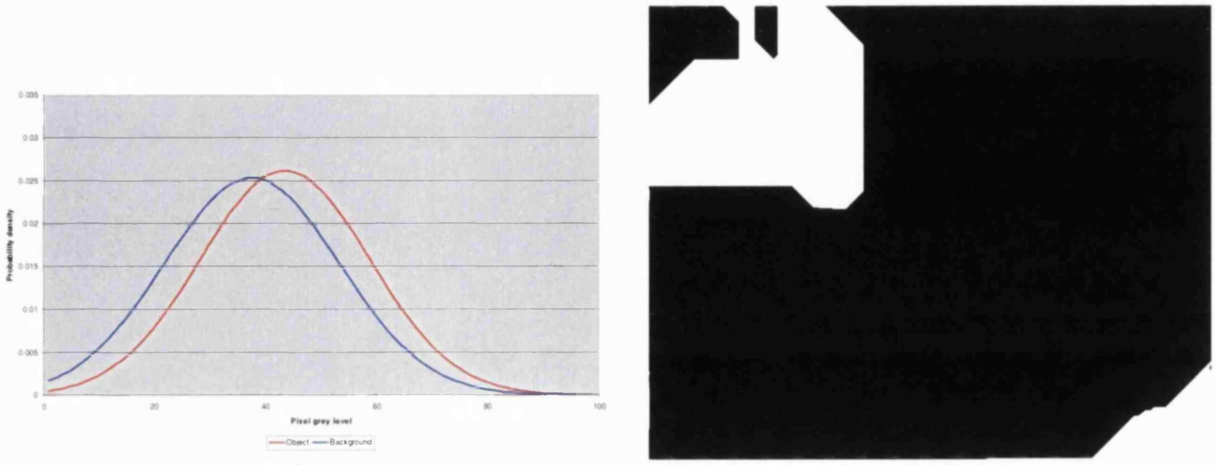


Figure 5.25 1st EM iteration (class conditional distribution estimate and E-MRF segmentation)

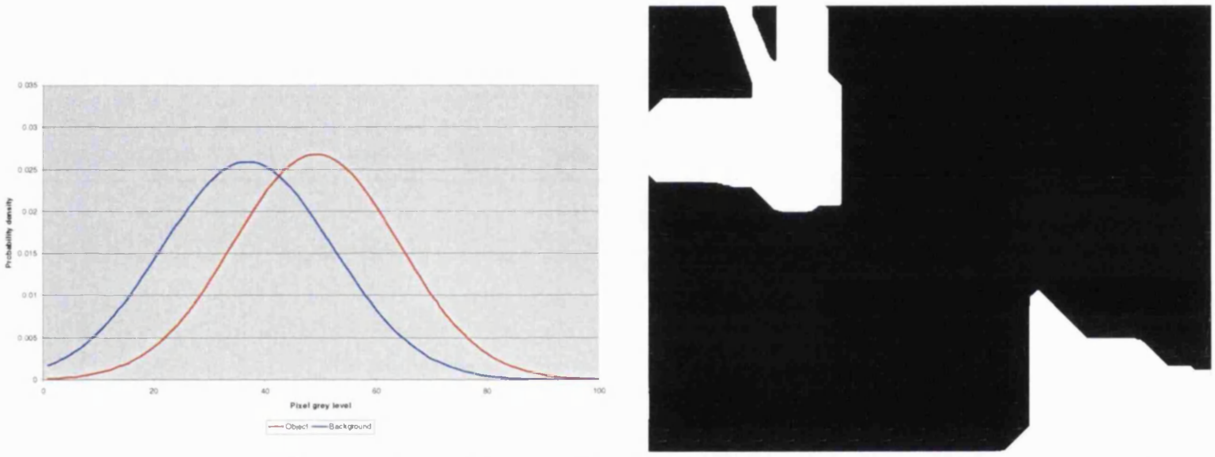


Figure 5.26 2nd EM iteration (class conditional distribution estimate and E-MRF segmentation)

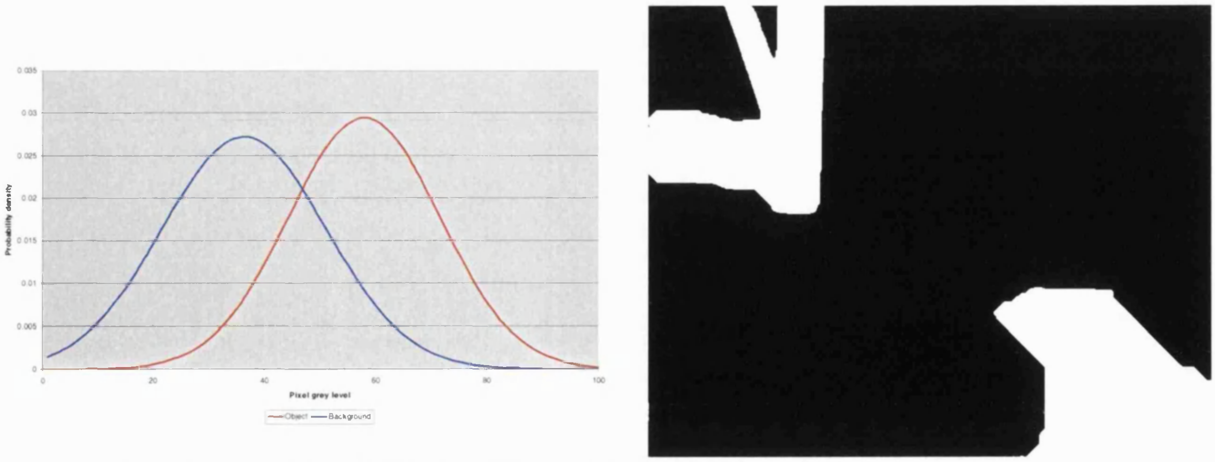


Figure 5.27 3rd EM iteration (class conditional distribution estimate and E-MRF segmentation)

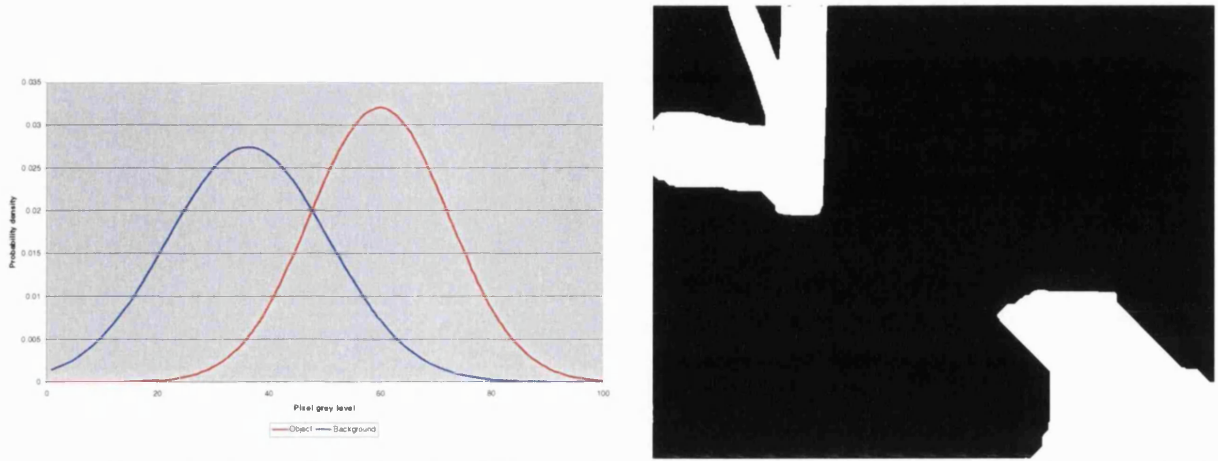
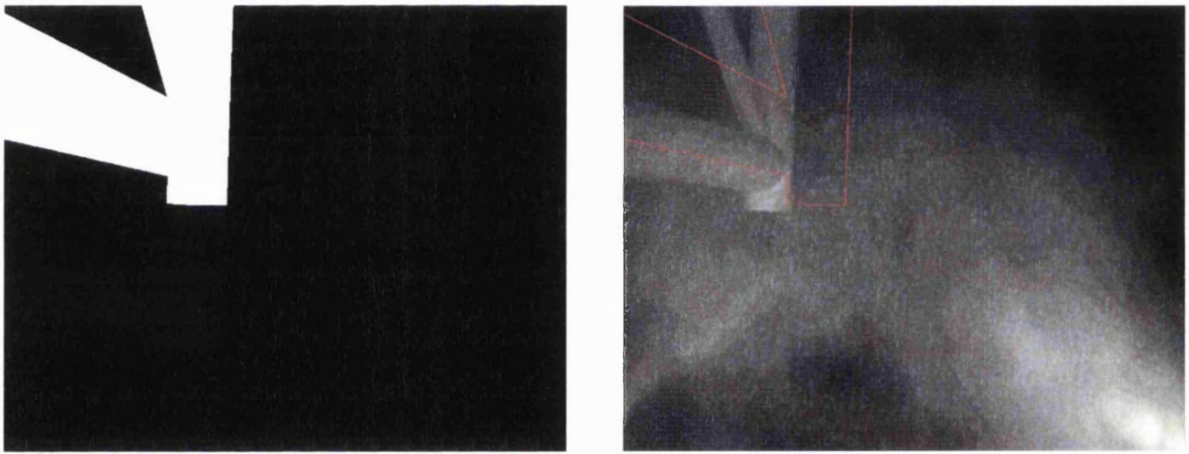
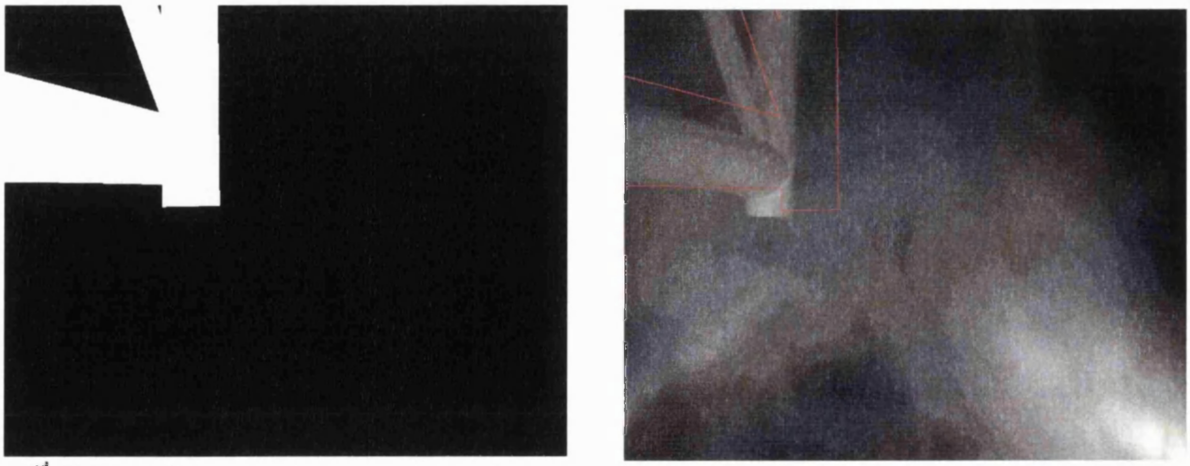


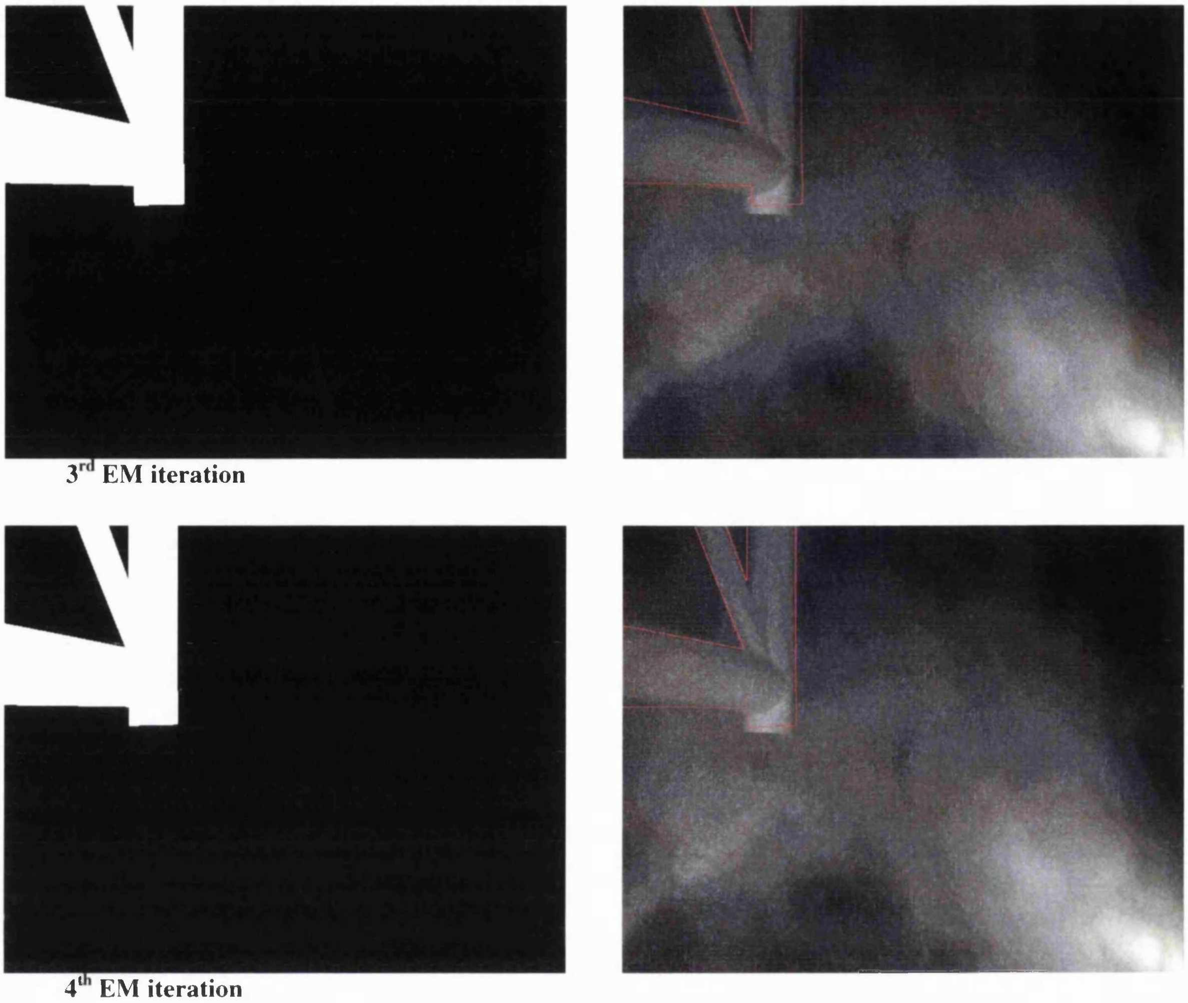
Figure 5.28 4^{th} EM iteration (class conditional distribution estimate and E-MRF segmentation)



1^{st} **EM iteration**



2^{nd} **EM iteration**



3rd EM iteration

4th EM iteration

Figure 5.29 Improvement of image predicted image with successive EM iterations. The left hand image shows the predicted image based on the current camera position estimate. The right hand image shows the outline (in red) of this prediction, superimposed over the observed image.

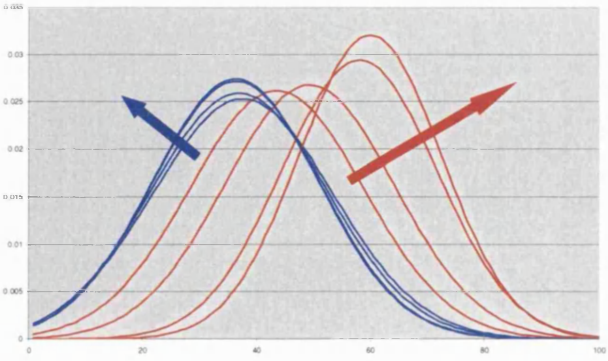


Figure 5.30 Progressive re-learning of class conditional distributions with successive iterations.

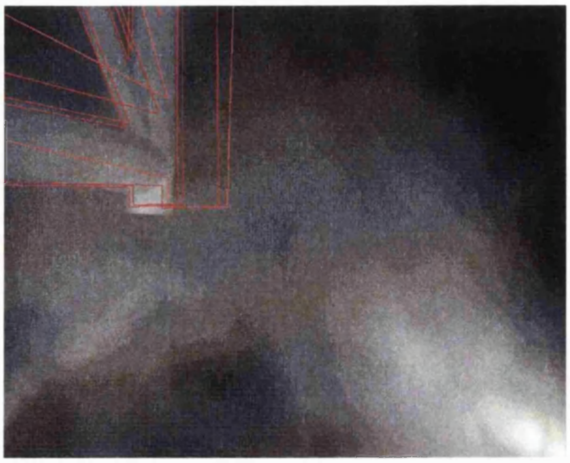


Figure 5.31 Improvement in position estimate with successive iterations.

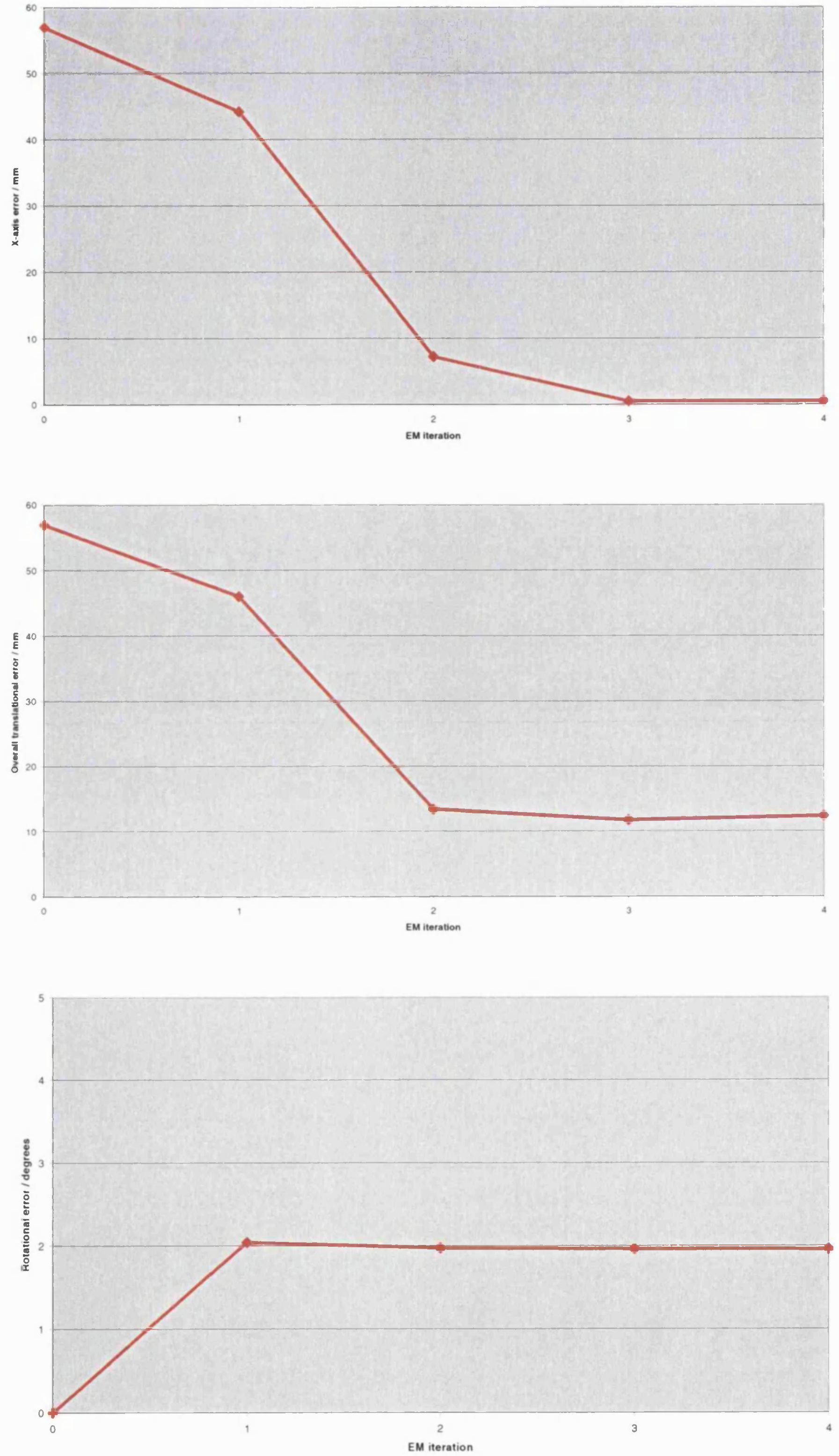


Figure 5.32 **Variation of translational and rotational errors with EM iterations for a starting error of 56.8mm. Translational error is greatly improved, though a rotational error of 1.96 degrees is a small but non-negligible deterioration in orientation estimate.**

5.3.3 Translational starting error of 85.2mm on x co-ordinate

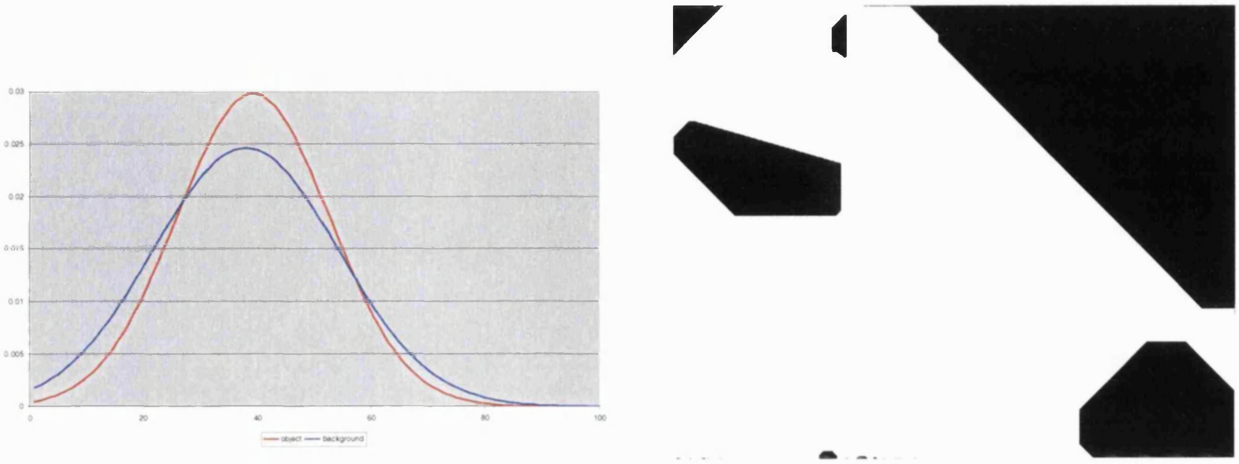


Figure 5.33 **1st (and only) EM iteration (class conditional distribution estimate and failed E-MRF segmentation)**

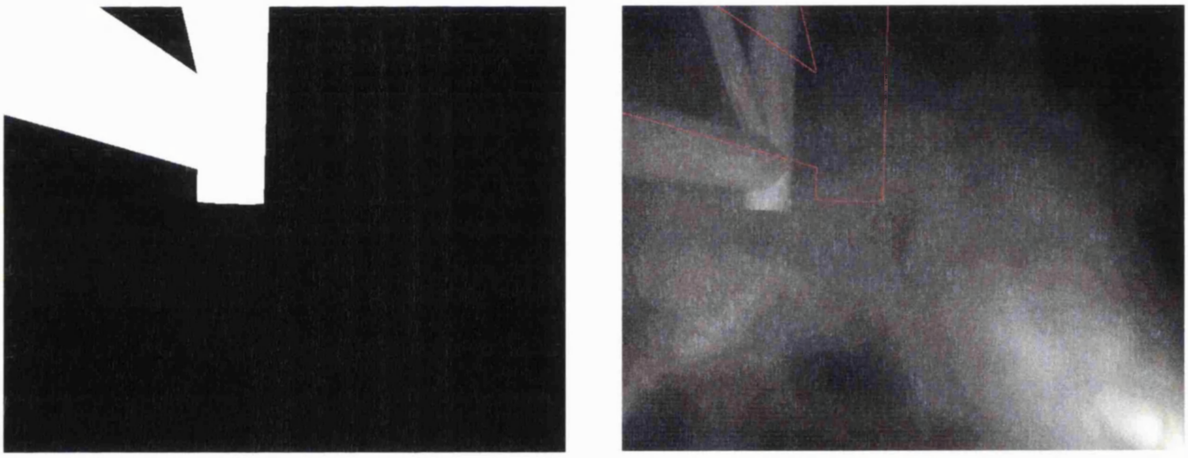


Figure 5.34 **Initial position estimate.**

In this case (figures 5.33 and 5.34) the starting error is too great for the algorithm to recover from. Critically, the "object" portion of the initial predicted image intersects with so little of the "object" in the observed image that the estimated class conditional distributions have approximately equal means. This means that the algorithm cannot correct prediction error by a strong discrimination between classes in the observed data. The algorithm terminates after a single iteration with no improvement in estimated position.

5.3.4 4.2 degrees rotational starting error

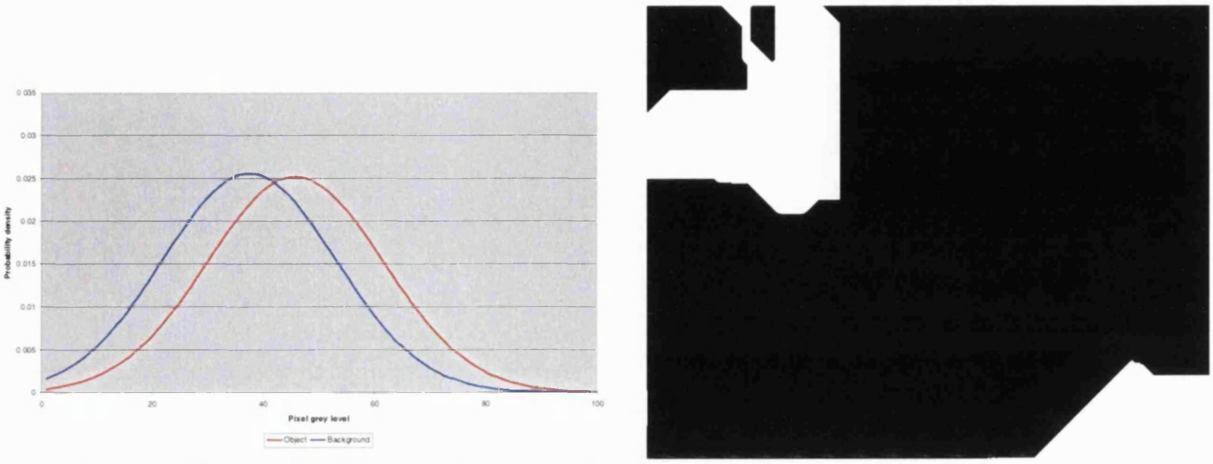


Figure 5.35 1st EM iteration (class conditional distribution estimate and E-MRF segmentation)

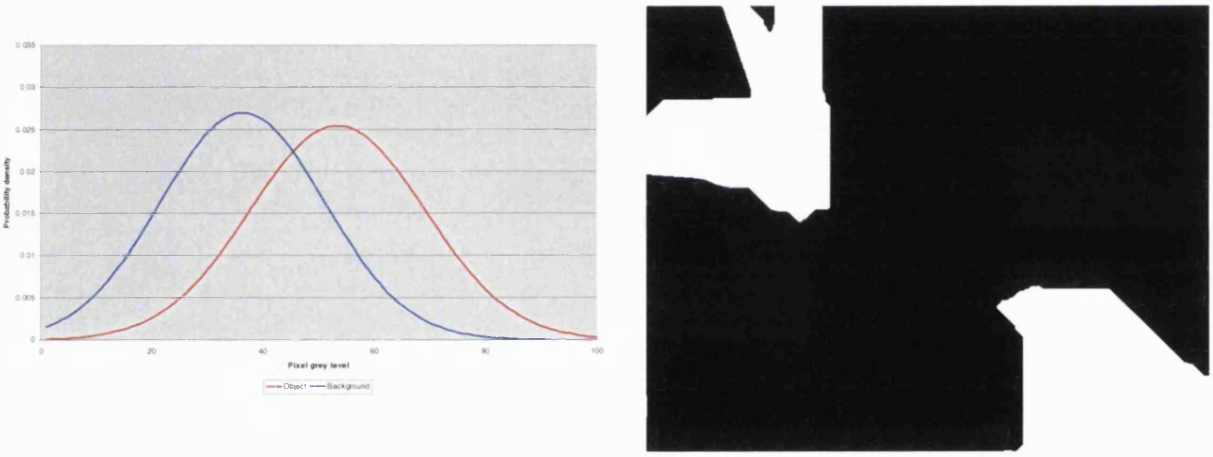


Figure 5.36 2nd EM iteration (class conditional distribution estimate and E-MRF segmentation)

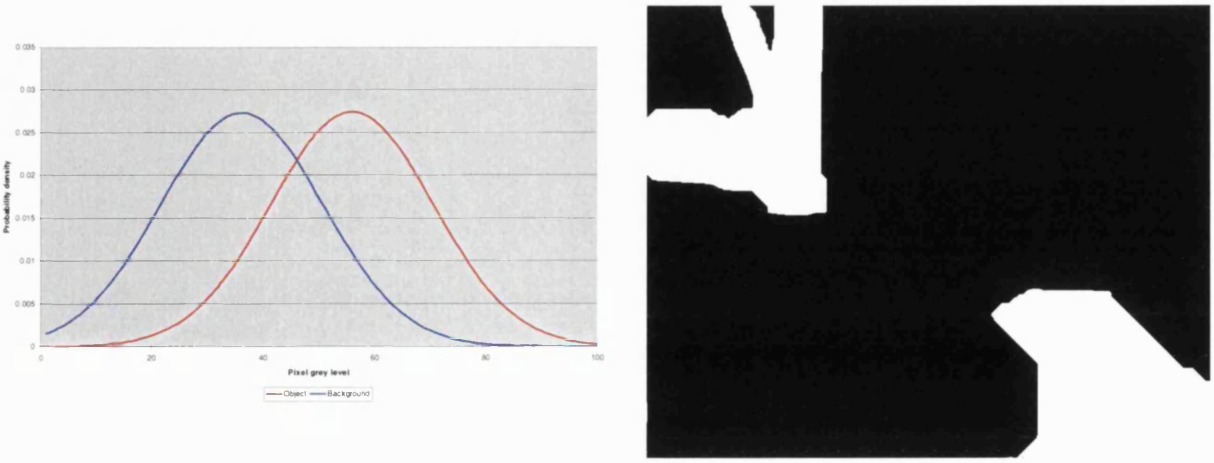


Figure 5.37 3rd EM iteration (class conditional distribution estimate and E-MRF segmentation)

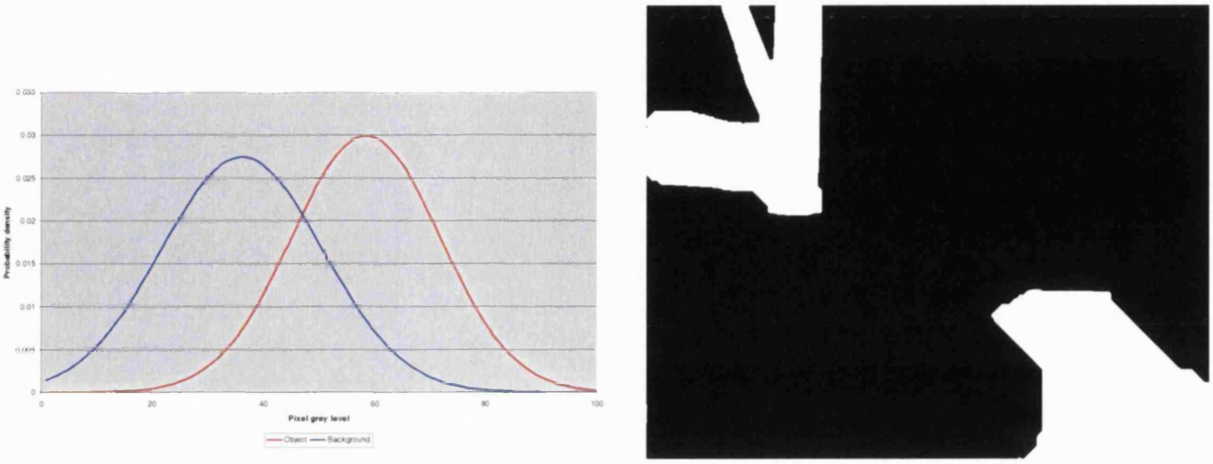


Figure 5.38 4th EM iteration (class conditional distribution estimate and E-MRF segmentation)

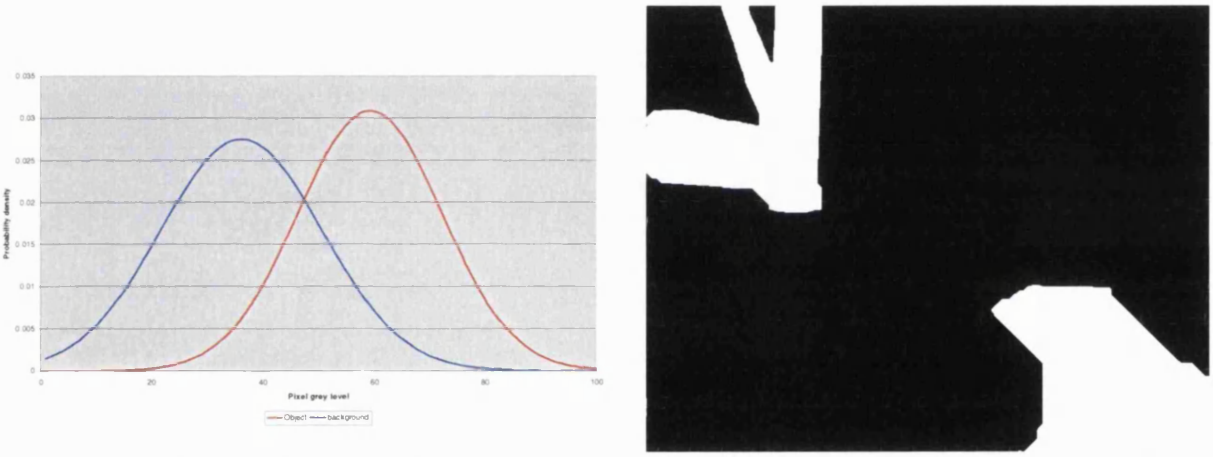
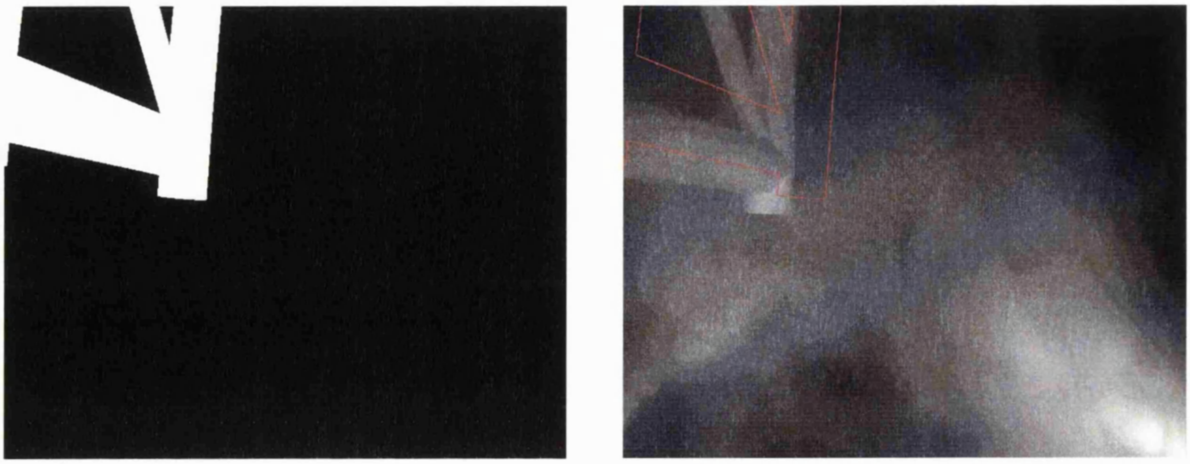
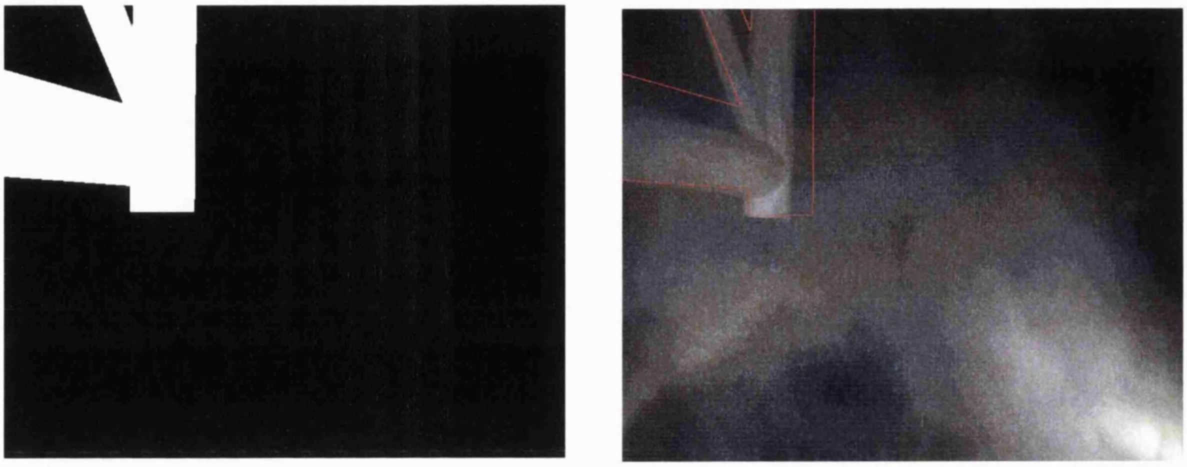


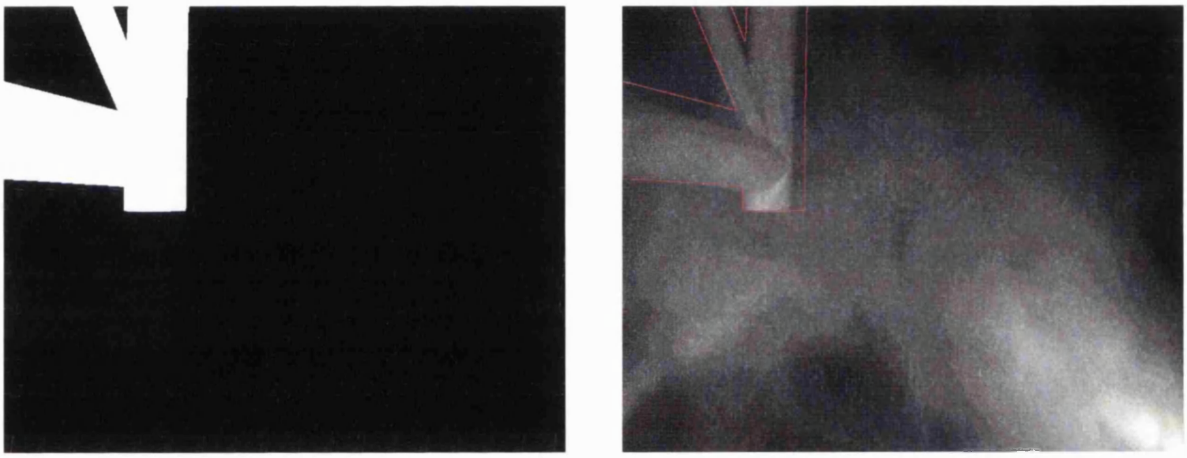
Figure 5.39 5th EM iteration (class conditional distribution estimate and E-MRF segmentation)



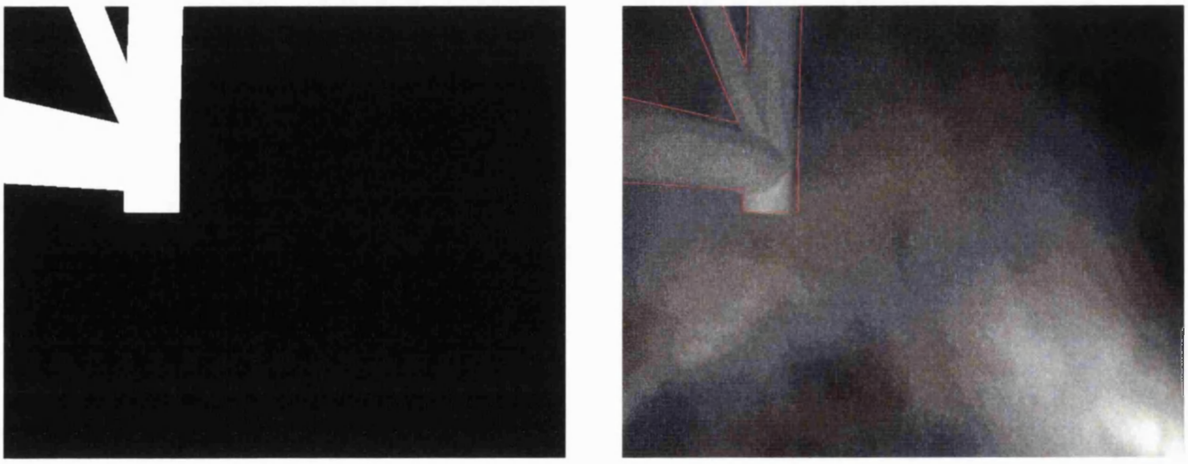
1st EM iteration



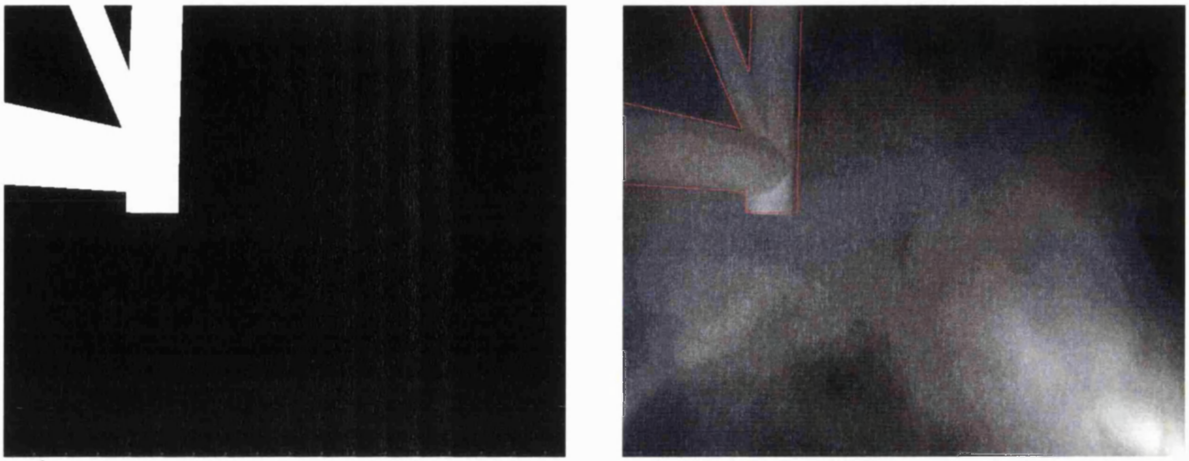
2nd EM iteration



3rd EM iteration



4th EM iteration



5th EM iteration

Figure 5.40 Improvement of predicted image with successive EM iterations. The left hand image shows the predicted image based on the current camera position estimate. The right hand image shows the outline (in red) of this prediction, superimposed over the observed image.

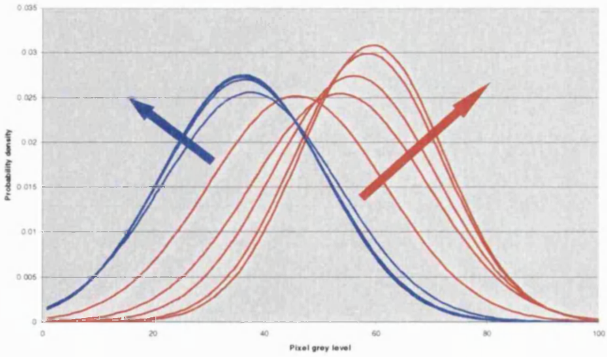


Figure 5.41 Progressive re-learning of class conditional distributions with successive iterations.

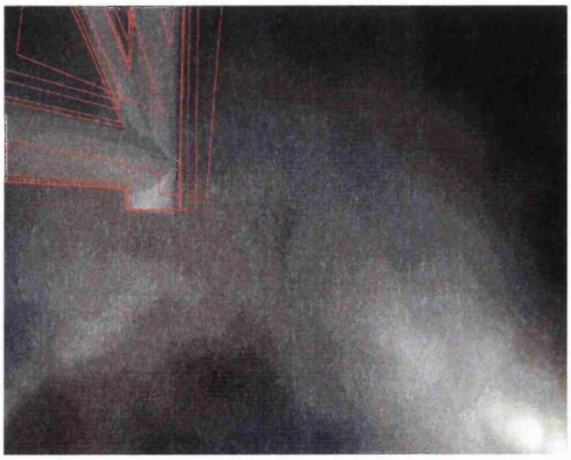


Figure 5.42 Improvement in position estimate with successive iterations.

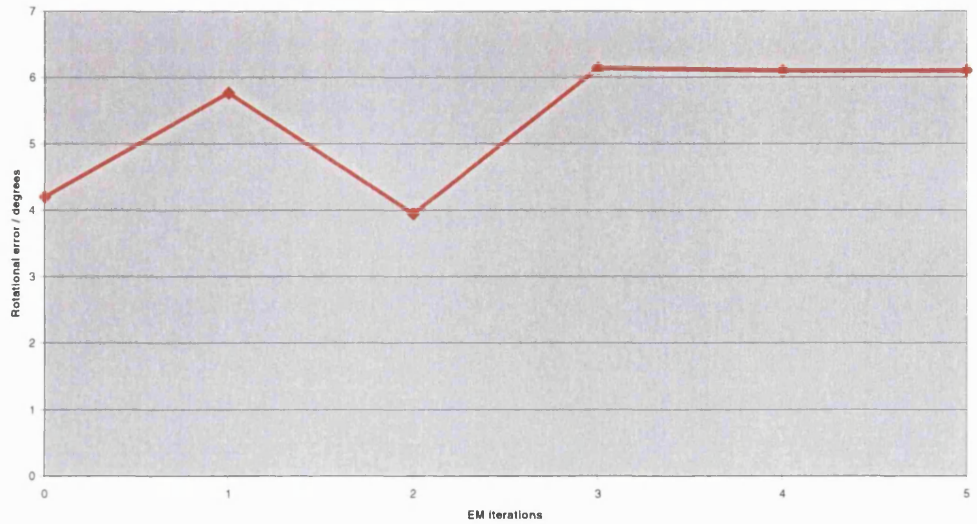


Figure 5.43 Variation of rotational error with EM iterations.

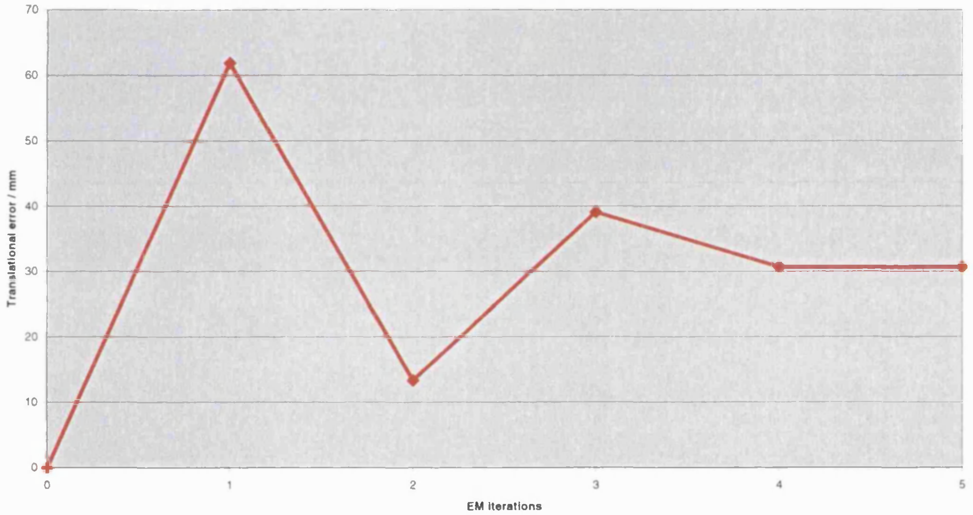


Figure 5.44 Variation of overall translational error with EM iterations.

The above result is interesting in that the image has been very accurately segmented (object has been correctly distinguished from background, see figure 5.40) but the rotational and translational errors have significantly increased (figures 5.43 and 5.44). The algorithm has in effect compensated for a rotational error by increasing a coupled translational error.

This is a fundamental limitation of vision based tracking systems. There may not be a one-to-one correspondence between observed images and unique camera positions. This may be because of symmetries in the observed object (e.g. a sphere looks the same when viewed from any direction and a cube looks the same when viewed from several different directions) but may also be because a range of camera positions produce very similar looking images resulting in an error space around the true camera location.

In particular, there is a coupling between certain directions of translation and rotation. Certain *small* rotations of the camera may be equivalent to corresponding *small* translations in terms of the observed effect on the position of important image features. Thus model based object tracking algorithms must generally be prone to convergence on moderately erroneous camera positions for which a combination of *small* translational and rotational errors results in a seemingly correct projection of an object model onto the observed image. This effect is illustrated in figure 5.45. Consistent with this idea, when complete camera trajectories are analysed in section 5.4, a high degree of correlation will be observed between rotational errors and translational errors.

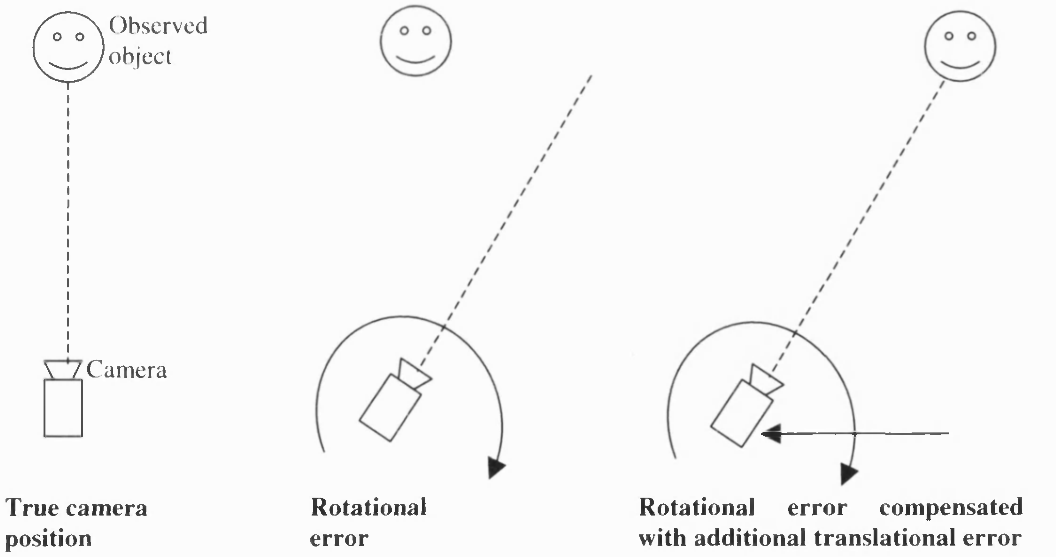

Observed object
Camera
True camera position
Rotational error
Rotational error compensated with additional translational error

Figure 5.45 **A combination of small rotational and translational errors can result in an erroneous camera position which still projects images similar to those projected from the true camera position.**

5.3.5 9.0 degrees rotational starting error

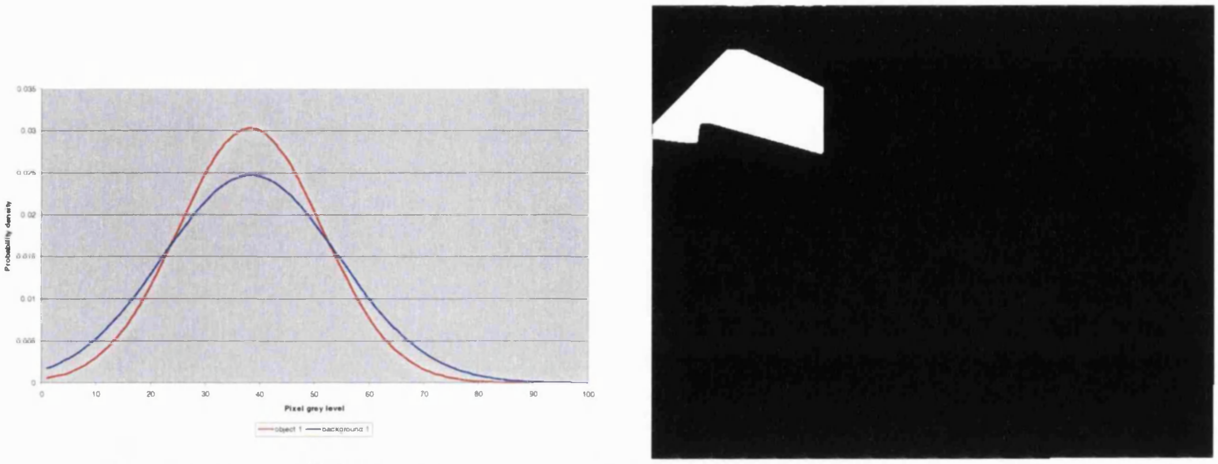


Figure 5.46 1st EM iteration (class conditional distribution estimate and E-MRF segmentation)

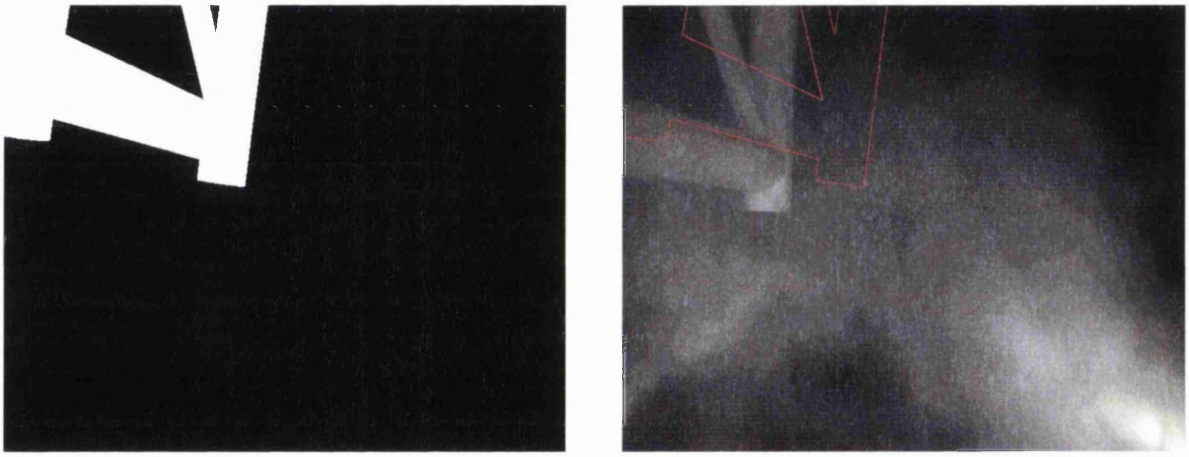


Figure 5.47 Predicted image based on the initial camera position estimate. The right hand image shows the outline (in red) of this prediction, superimposed over the observed image.

This result is similar to that of section 5.3.3. Again the starting error is too great for the algorithm to recover from. The "object" portion of the initial predicted image intersects with so little of the "object" in the observed image that the estimated class conditional distributions have approximately equal means. This means that the algorithm cannot correct prediction error by a strong discrimination between classes in the observed data. The algorithm terminates after a single iteration with no improvement in estimated position.

5.4 Trajectory tracking

This section presents the results of testing the EM/E-MRF algorithm on extended image sequences. Poor visibility image sequences are used, and also an "ideal" visibility synthetic image sequence. The performance of the algorithm is investigated while varying two key parameters.

The first parameter (u from equation 3.58, see section 3.8.2) determines the degree of interpolation between the predicted (via trajectory extrapolation) camera position and the measured (via the EM/E-MRF algorithm) camera position. In this respect, u functions in a similar fashion to a Kalman gain (Kalman [1960], Welsch [2002]). With a u value of zero, the vision system becomes a dead reckoning system, estimating the current camera position purely by extrapolating the prior trajectory (assuming constant velocity) and completely ignoring any observed information. With a u value of 1.0, the vision system will rely exclusively on the position extracted from each image via the EM/E-MRF algorithm. Consequently it will be observed that overly high values of u produce jagged, erratic trajectory estimates while small values of u produce smooth trajectory estimates which gradually drift away from the ground truth.

The second parameter (S_2 in equation 3.14, see section 3.4) determines the weighting, during E-MRF segmentation, assigned to likelihood function contributions due to corresponding pixels in the predicted image. This parameter controls how much prediction is used during image segmentation. An S_2 value of zero results in segmentation by conventional MRF, having no contribution from a predicted image. A very high S_2 value will result in the segmentation process simply reproducing the predicted image, with no contribution from the observed image.

In good visibility, it is desirable to use a high u value and a low S_2 value in order to incorporate as much (good quality) information from the observed image as possible. In poor visibility it is necessary to decrease u and increase S_2 in order to compensate for missing observed information by using additional predicted information. Too much prediction eventually results in pure dead reckoning which steadily accumulates errors over time.

5.4.1 Synthesised "perfect" images-best case scenario

The purpose of this experiment is to gain understanding of the upper limit of performance of the vision system. Artificial images were projected, corresponding to views from the ground-truth camera positions from a 201 frame portion of the trajectory measured in chapter 4. These artificial images constitute "perfect" visibility in that they show "object" pixels as pure white (grey level of 255) and "background" pixels as black (grey level of zero). The algorithm can never be expected to perform better on any other kind of image or under any other visibility conditions. This test is also useful since it eliminates any possible errors in the ground-truth trajectory measurements or camera model.

For this experiment u was assigned the value 0.7 and S_2 was assigned the value 1.0. The vision system is observed (figure 5.48) to successfully track the image sequence, even when the camera trajectory includes a sharp corner. The rms translational error during this sequence was 3.02 mm and the rms rotational error was 1.17 degrees.

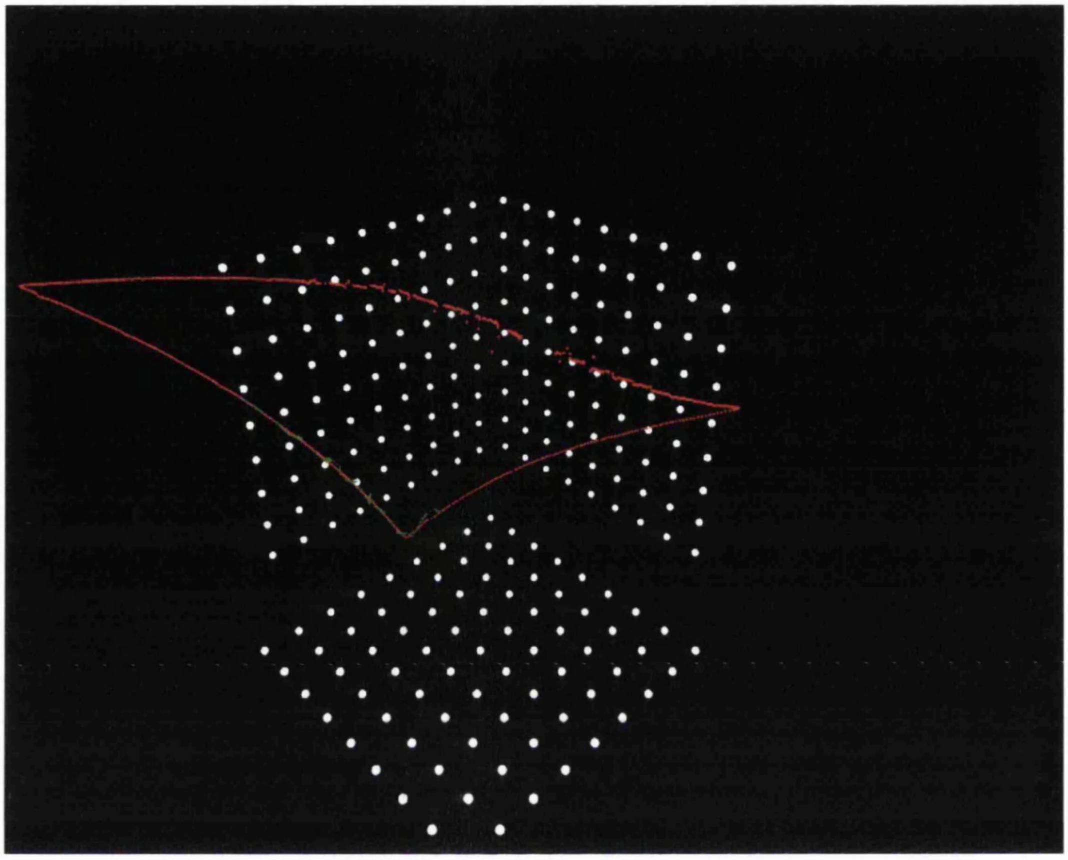


Figure 5.48 **Measured trajectory output by the vision system (green) compared to ground truth trajectory (red) for a 201 frame good visibility (synthetic) image sequence for which the camera follows a trajectory incorporating a sharp corner. The calibration target spot positions are shown as a visual reference.**

It is interesting to note (figure 5.49 and figure 5.50) that translational errors are highly correlated with rotational errors. This supports the idea, expressed in section 5.3.4, that certain combinations of camera rotations and translations result in very little change to a projected image, thus imposing a limit on the accuracy to which a camera position can be extracted from an observed image.

The accuracy of image segmentation achieved with a relatively erroneous position estimate (figure 5.51) is almost as good as that achieved with an extremely accurate position estimate (figure 5.52).

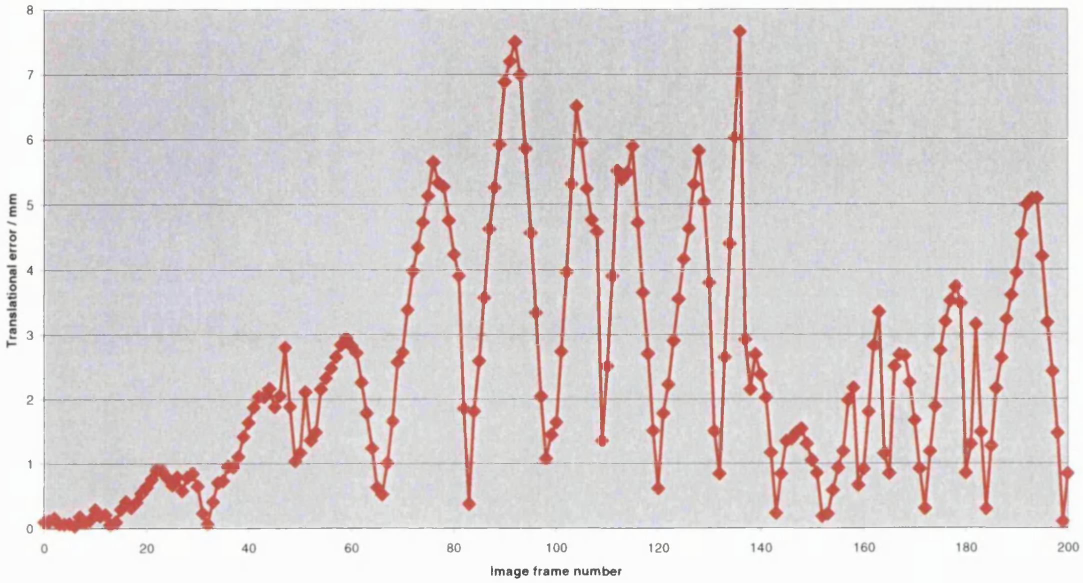


Figure 5.49 Translational error at each frame.

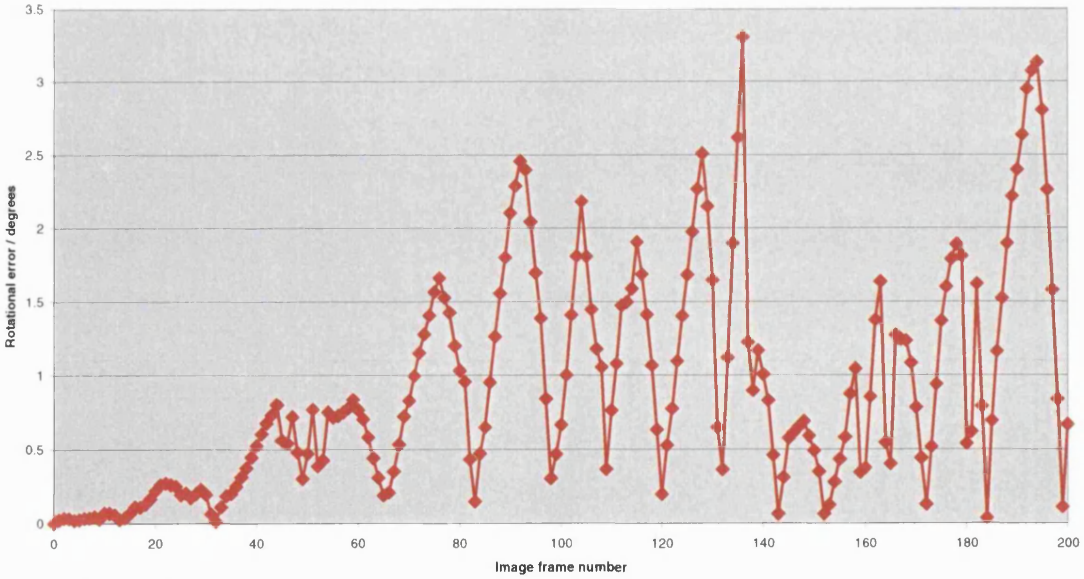


Figure 5.50 Rotational error at each frame.

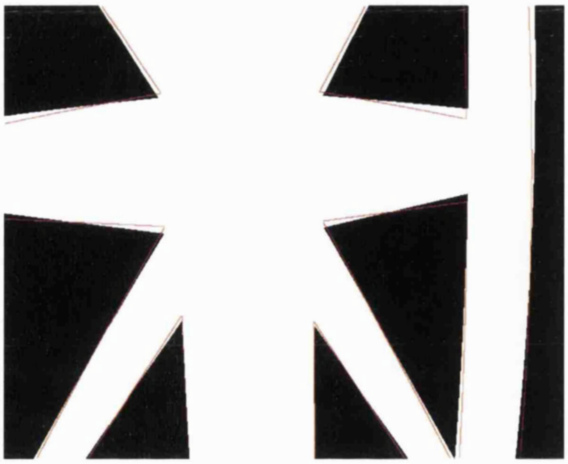


Figure 5.51 Frame 136 vision system output (red) superimposed over observed image. Even though the output camera position contains a relatively large error (7.7 mm and 3.3 degrees), the segmentation and model fitting appear to be accurate.

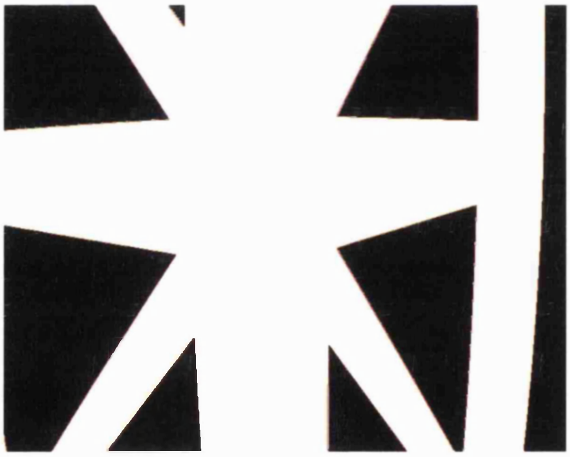


Figure 5.52 Frame 143 vision system output (red) superimposed over observed image. For this frame, the camera position output by the vision system is extremely accurate (0.3mm translational error and 0.06 degrees rotational error).

It is also interesting to observe that the vision system can recover from a relatively erroneous position estimate (frame 136, see figure 5.51) to achieve an extremely accurate position estimate a few frames later (frame 143, see figure 5.52).

5.4.2 Poor visibility with over-prediction

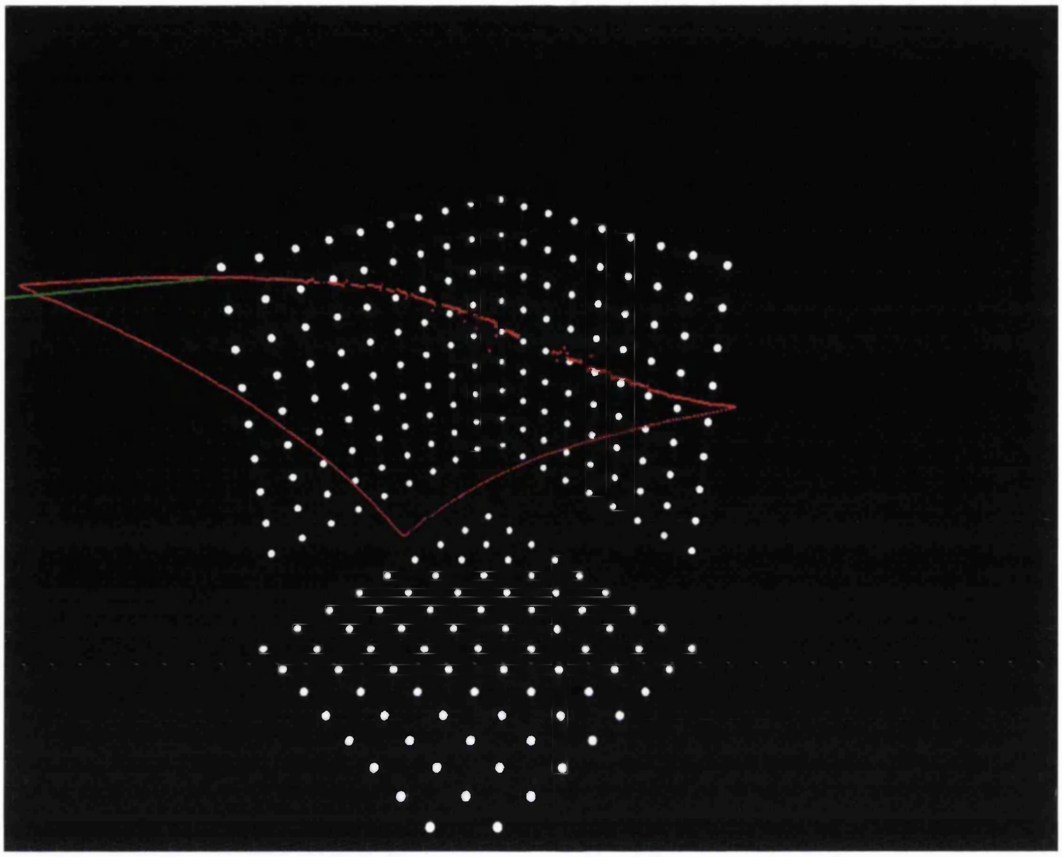


Figure 5.53 **Measured trajectory output by the vision system (green) compared to ground truth trajectory (red) for a poor visibility (dry ice fog and moving lights) image sequence for which the camera follows a trajectory incorporating a sharp corner. The calibration target spot positions are shown as a visual reference.**

This experiment attempts to tackle a poor visibility (dry ice fog and moving, focussed beam light sources) image sequence. A u value of 0.3 and an S_2 value of 4.0 were used. The u value implies a relatively heavy weighting in favour of predicted position when interpolating between predicted and observed position estimates. The S_2 value implies a relatively high significance of predicted pixel class during image segmentation.

The vision system produces a trajectory estimate which is smooth but deteriorates in a similar fashion to dead reckoning navigation systems, with positional error gradually increasing with time. The vision system fails to negotiate a sharp corner in the trajectory.

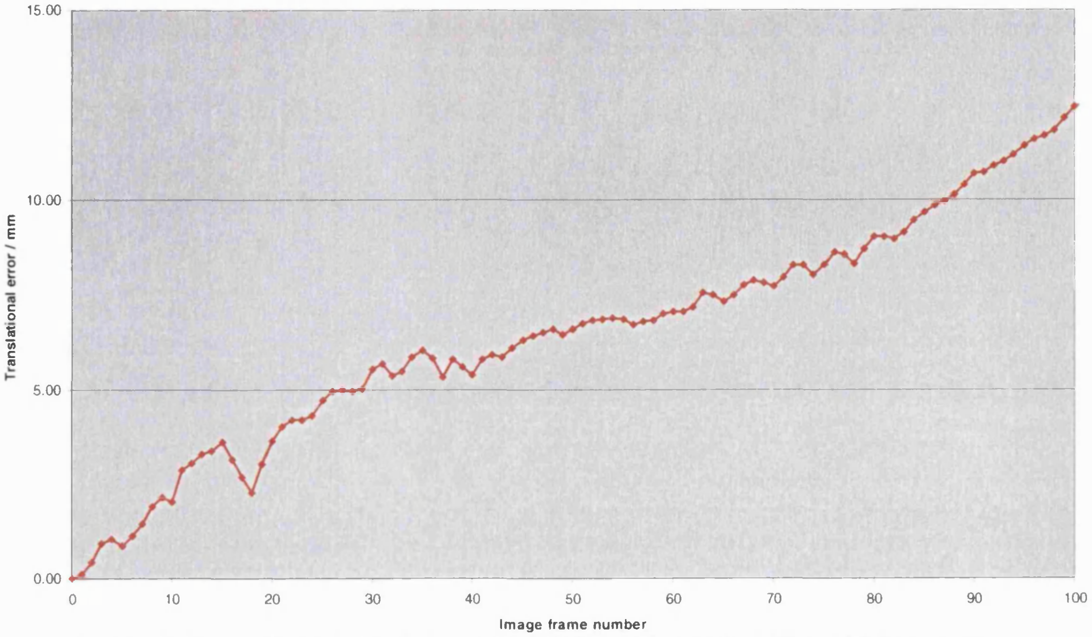


Figure 5.54 **Translational error at each frame (analysis of first 100 frames of sequence).**

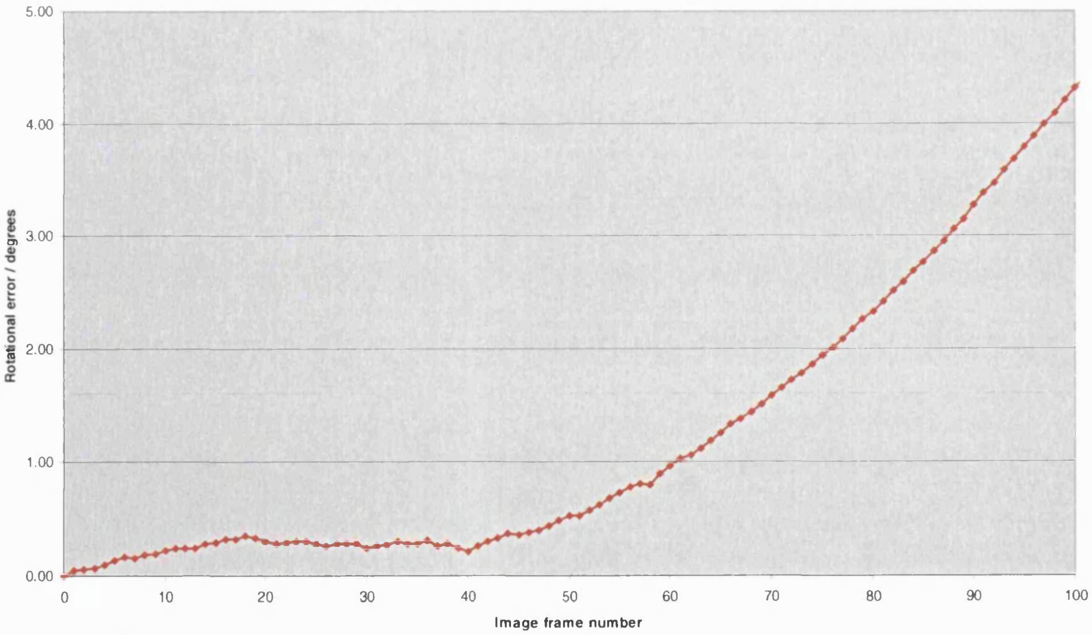
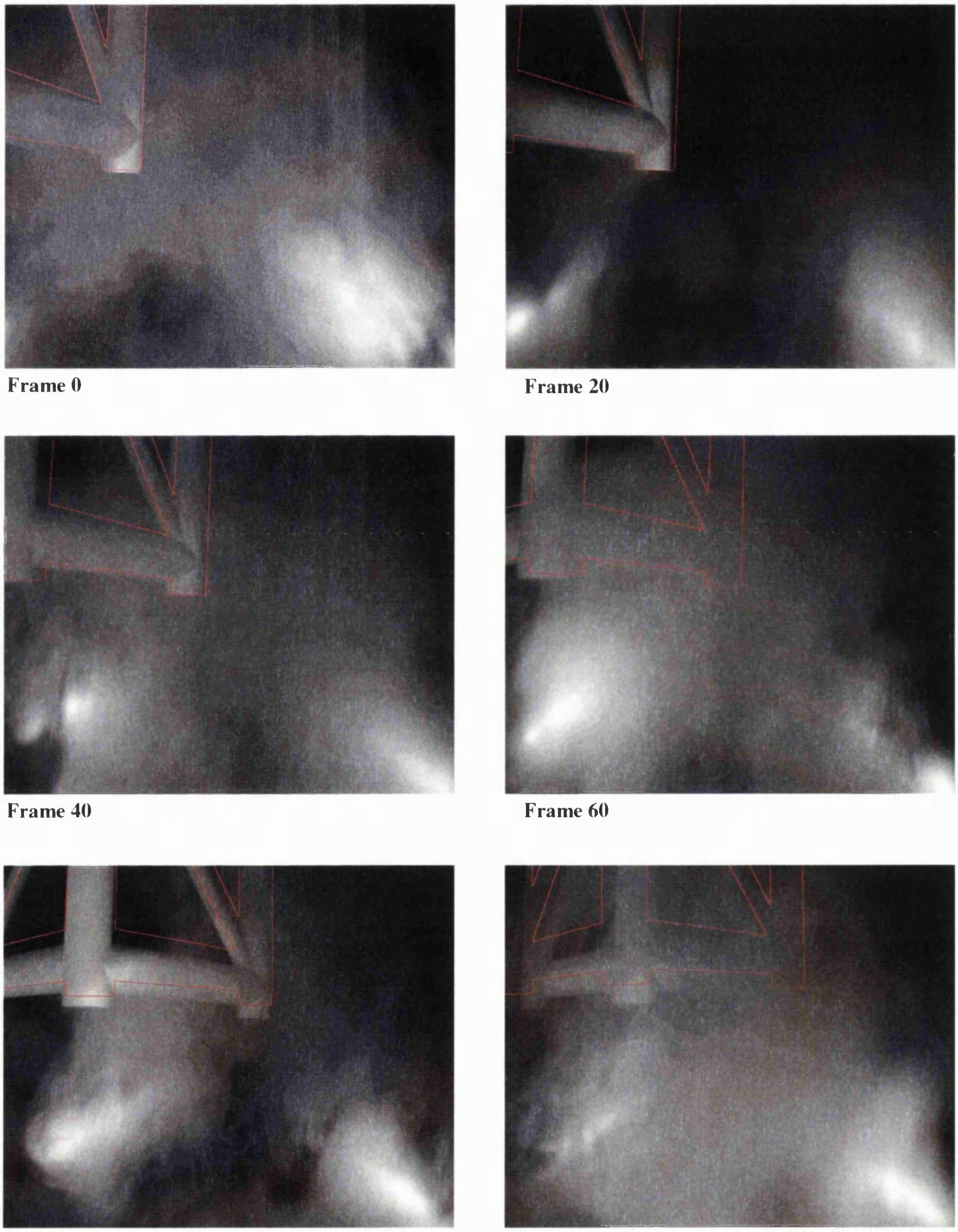


Figure 5.55 **Rotational error at each frame (analysis of first 100 frames of sequence).**



Frame 0

Frame 20

Frame 40

Frame 60

Frame 80

Frame 100

Figure 5.56 Vision system output superimposed over observed images. The sequence has been sampled once every 20 frames.

5.4.3 Poor visibility with under-prediction

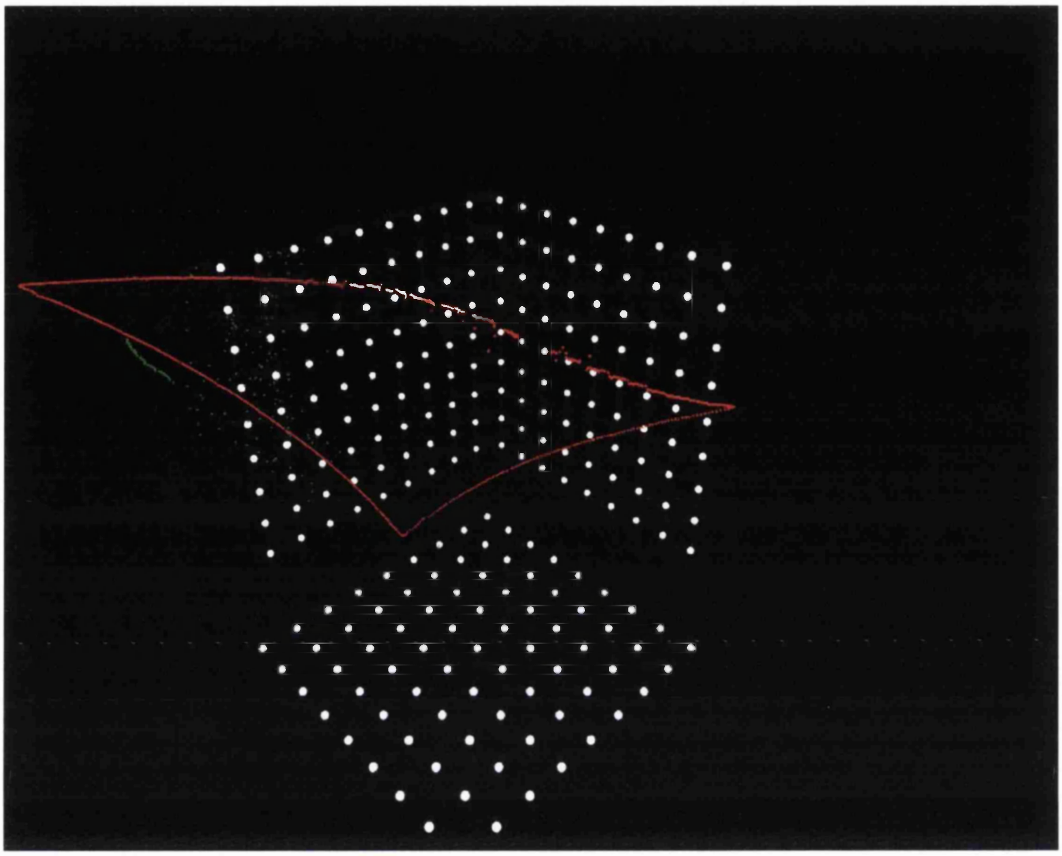


Figure 5.57 Measured trajectory output by the vision system (green) compared to ground truth trajectory (red) for a poor visibility (dry ice fog and moving lights) image sequence along a smooth trajectory section.

For this experiment a u value of 1.0 and an S_2 value of 1.0 were used. These values imply a relatively small predictive contribution and an over-acceptance of poor quality observed data. The vision system roughly tracks for a short period before failing.

5.4.4 Poor visibility with over-prediction and under-interpolation

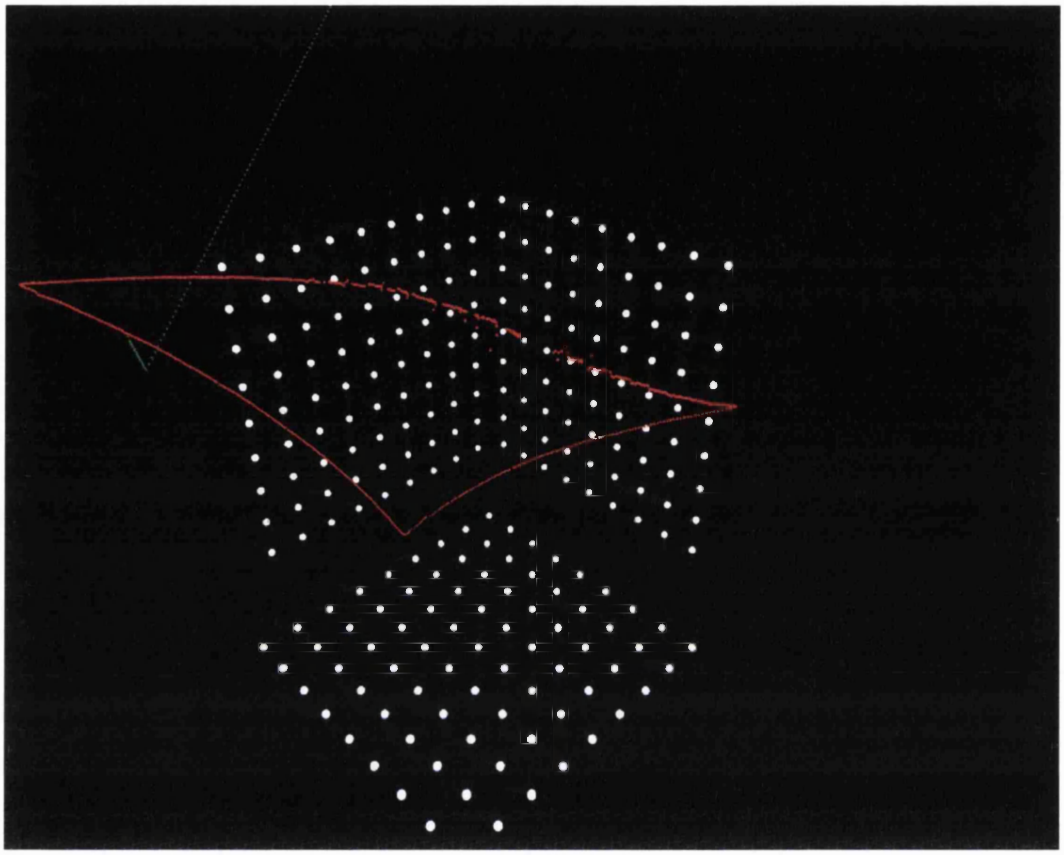


Figure 5.58 **Measured trajectory output by the vision system (green) compared to ground truth trajectory (red) for a poor visibility (dry ice fog and moving lights) image sequence along a smooth trajectory section.**

For this experiment a u value of 1.0 and an S_2 value of 3.0 were used. The u value implies that whatever position estimate is computed by the vision system is accepted without any interpolation with the position predicted by trajectory extrapolation, i.e. total reliance on observed position. In contrast, the S_2 value implies a large significance for the predicted class of pixels during image segmentation. This choice of values can be thought of as a mixture of over-prediction and under-prediction. The behaviour shares similarities with the over-predicted case of section 5.4.2 in that the system ultimately appears to follow its own path regardless of observed information.

The difference is that in this case the final direction is not an extrapolation of the original ground truth trajectory.

5.4.5 Successful tracking with moderate prediction

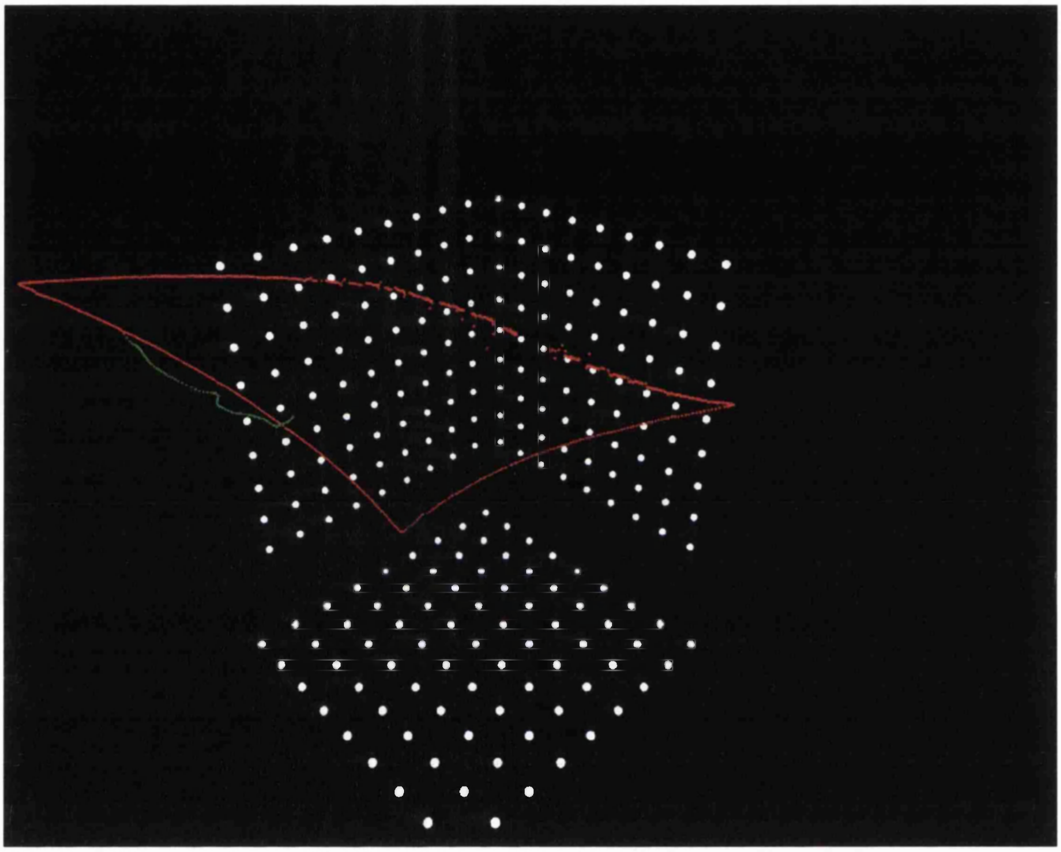


Figure 5.59 Measured trajectory output by the vision system (green) compared to ground truth trajectory (red) for a 101 frame poor visibility (dry ice fog and moving lights) image sequence along a smooth trajectory section.

For this experiment a u value of 0.6 and an S_2 value of 1.5 were used. Both of these values are mid range and represent a moderate weighting between observed and predicted data. The rms translational error over this 101 frame sequence was 24.9mm and the rms rotational error was 3.6 degrees.

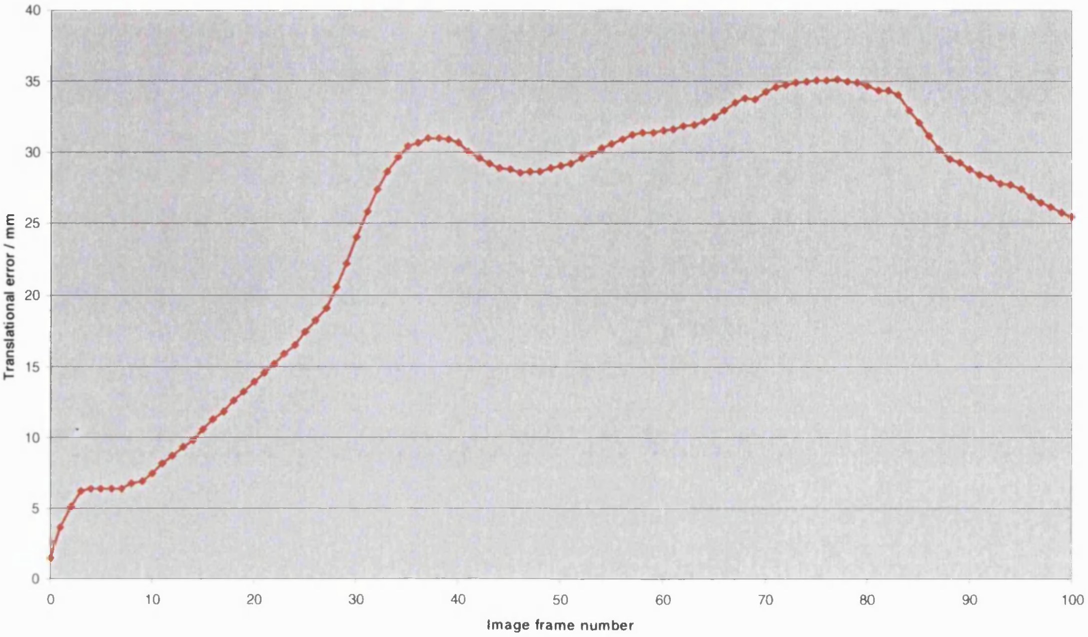


Figure 5.60 Translational error at each frame.

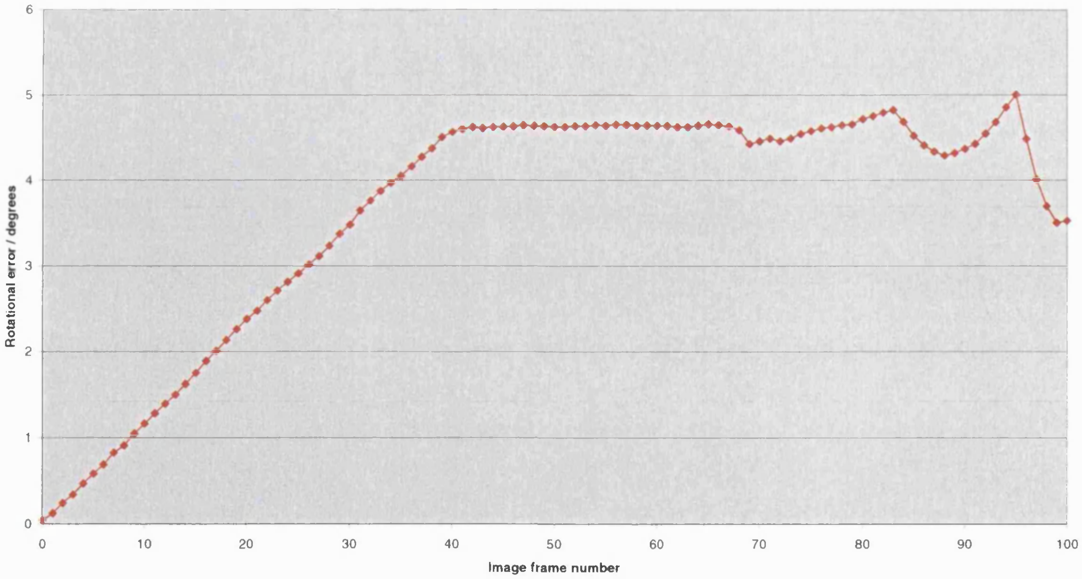
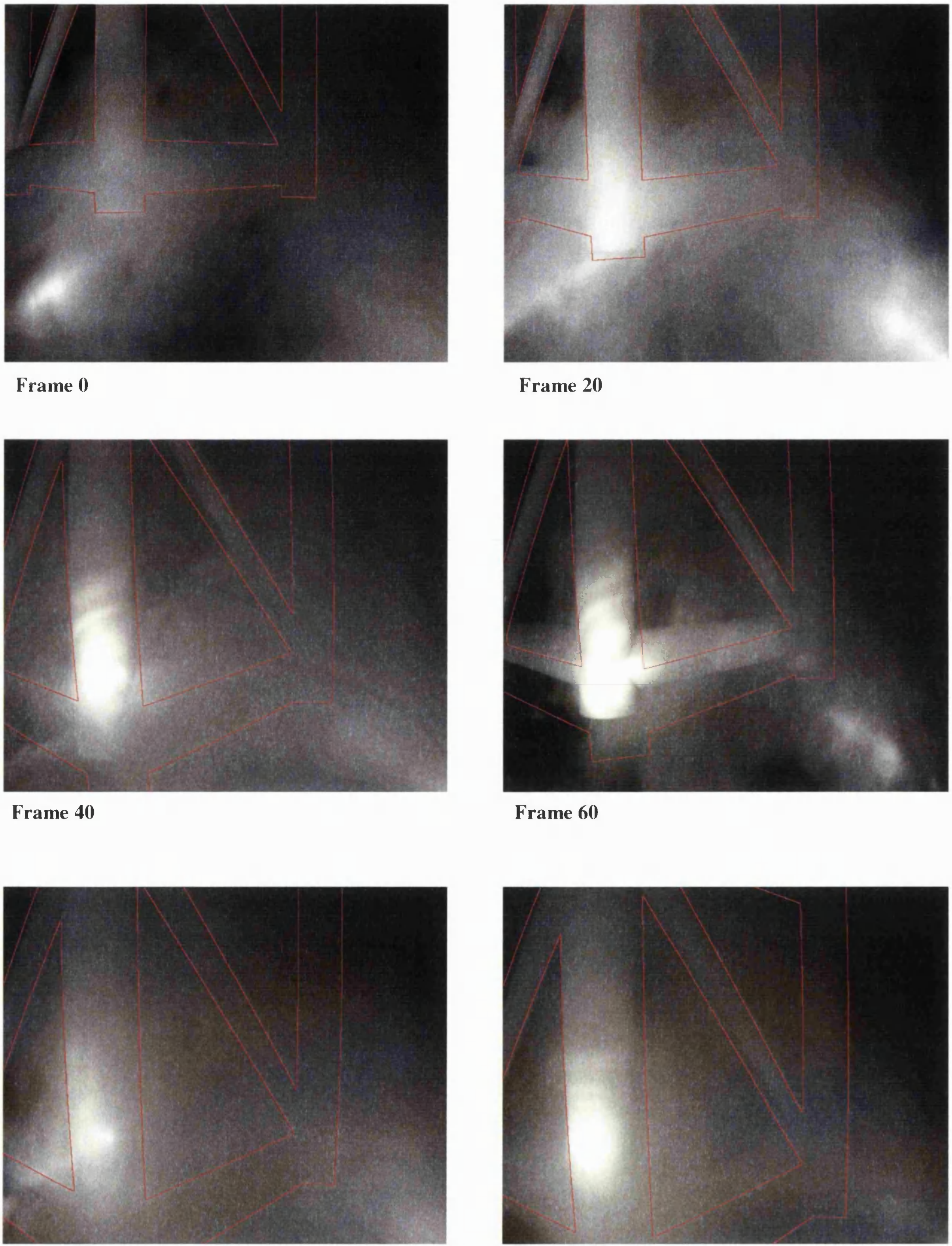


Figure 5.61 Rotational error at each frame.



Frame 0

Frame 20

Frame 40

Frame 60

Frame 80

Frame 100

Figure 5.62 **Vision system output superimposed over observed images. Sequence sampled once every 20 frames.**

5.4.6 Successful tracking with moderate prediction

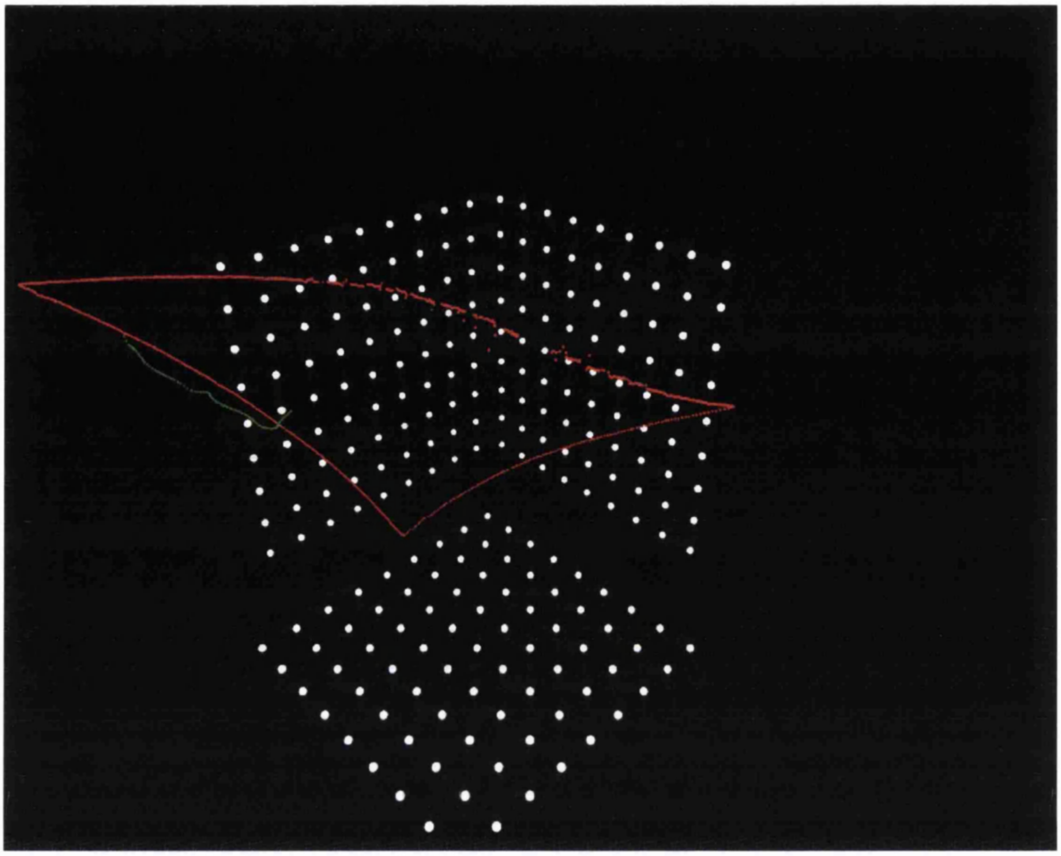


Figure 5.63 Measured trajectory output by the vision system (green) compared to ground truth trajectory (red) for a 101 frame poor visibility (dry ice fog and moving lights) image sequence along a smooth trajectory section.

For this experiment a u value of 0.5 and an S_2 value of 1.0 were used. Both of these values are mid range though they represent less prediction weighting than the values for the previous experiment (section 5.4.5). The rms translational error over this 101 frame sequence was 27.2mm and the rms rotational error was 3.6 degrees. These errors are very similar in magnitude to those of the previous experiment with a marginally larger translational error and smaller rotational error.

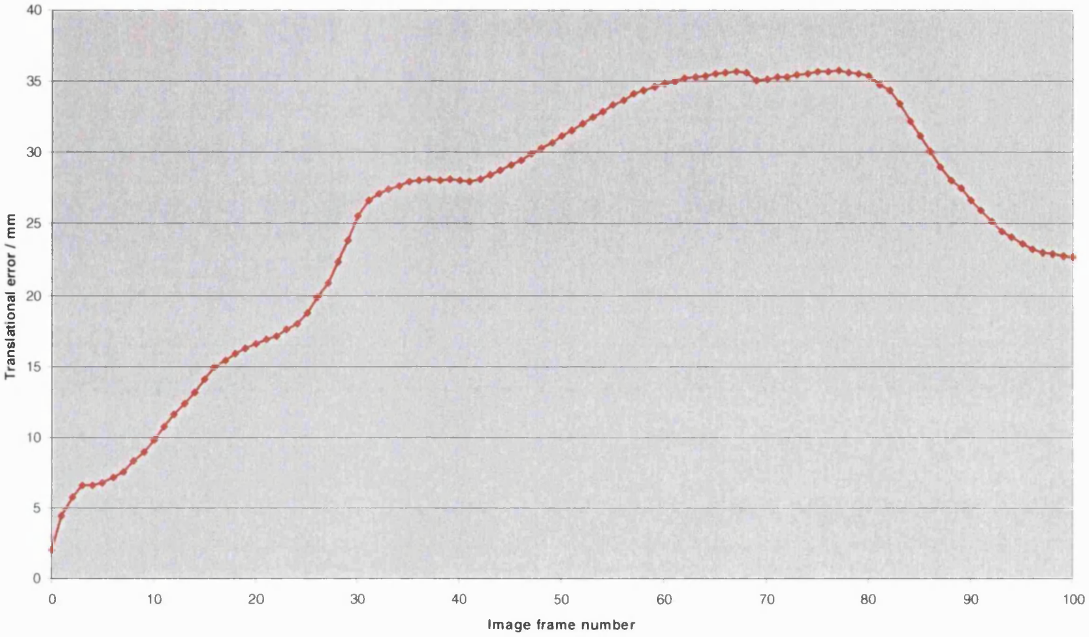


Figure 5.64 Translational error at each frame.

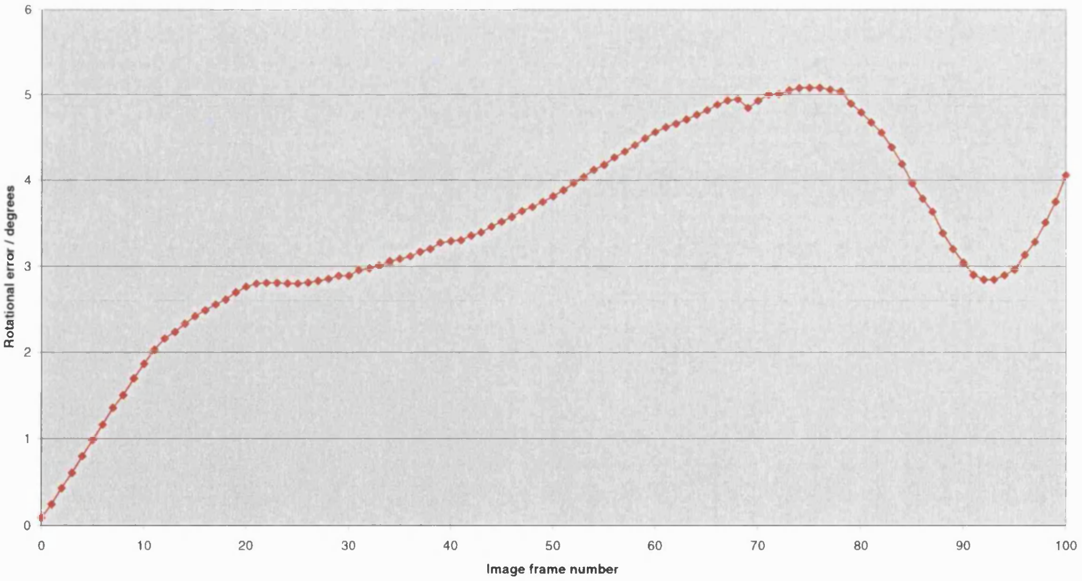
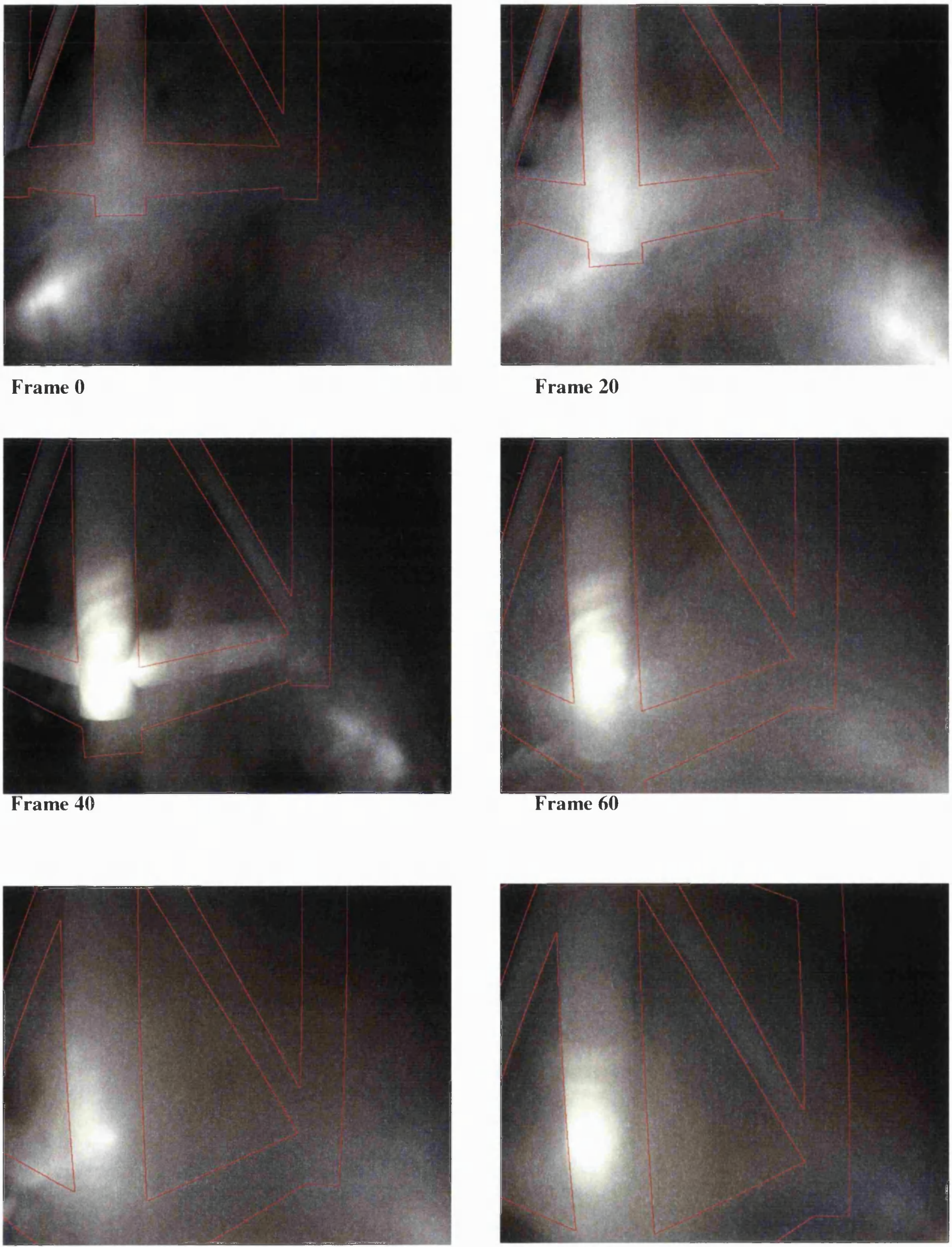


Figure 5.65 Rotational error at each frame.



Frame 0

Frame 20

Frame 40

Frame 60

Frame 80

Frame 100

Figure 5.66 Vision system output superimposed over observed images. Sequence sampled once every 20 frames.

5.4.7 Tracking problems with different objects

The EM/E-MRF algorithm was also tested on image sequences featuring another object (the "block" object, see chapter 4). Two experimental test sequences (including a sharp corner in the camera trajectory) are shown featuring the results obtained using firstly, a u value of 0.8 and an S_2 value of 3.1 (figures 5.68 to 5.71) and secondly, a u value of 0.7 and an S_2 value of 4.0 (figures 5.72 to 5.75).

In both cases the algorithm is able to successfully track the 2D position of the object in each image throughout the image sequence (although this accuracy does deteriorate towards the end of one sequence, the algorithm still manages to broadly identify the correct portion of each image as being "object"). Additionally, in both cases the 3D position and orientation of the camera is tracked accurately during the smooth segment of the camera trajectory (first five frames of the first example and first 25 frames of the second example). However, in both cases the estimated 3D camera trajectory becomes highly erroneous once the true trajectory encounters a sharp corner, despite the fact that the algorithm continues to accurately track the 2D position of the object in each image.

It is interesting that the vision algorithm may produce seemingly sensible interpretations of 2D image content whilst producing significantly erroneous 3D camera position estimates. This may be because the geometry of the object (a cuboid) means that views from different directions are not sufficiently different to be distinguishable, especially in poor visibility conditions in which parts of the object are obscured by clouds of fog. The problems of tracking a tall, thin block in poor visibility are similar to those of observing a cylinder (figure 5.67) in good visibility-it can appear similar when viewed from many different directions. It is observed that the vision system manages to track the object reasonably successfully from image to image, but the system's estimate of the orientation of the object (with respect to its own vertical axis) deteriorates with time.

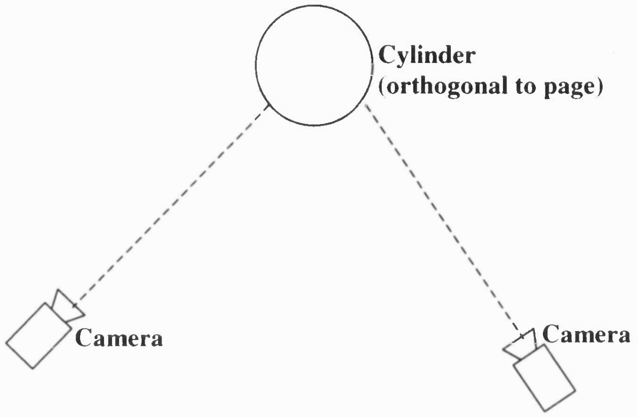


Figure 5.67 **Two possible views of a cylinder that will result in indistinguishable images**

5.4.7.1 First experiment with "block" object

This experiment (figures 5.68 to 5.71) uses a 51 frame test sequence based on a ground-truth camera trajectory that features a smoothly curved section (approximately first five frames) followed by a sharp corner leading to another relatively smooth portion of trajectory. The vision algorithm parameters used were a u value of 0.8 and an S_2 value of 3.1.

The algorithm was able to accurately and consistently track the 2D position of the object in each image frame over the entire sequence, even after the sharp corner in the camera trajectory (figure 5.69). However, after the corner (at approximately the fifth frame), the estimated 3D position and orientation of the camera deteriorate linearly with time (figures 5.70 and 5.71). The estimated camera path is observed to smoothly drift away from the ground-truth path after the corner event (figure 5.68).

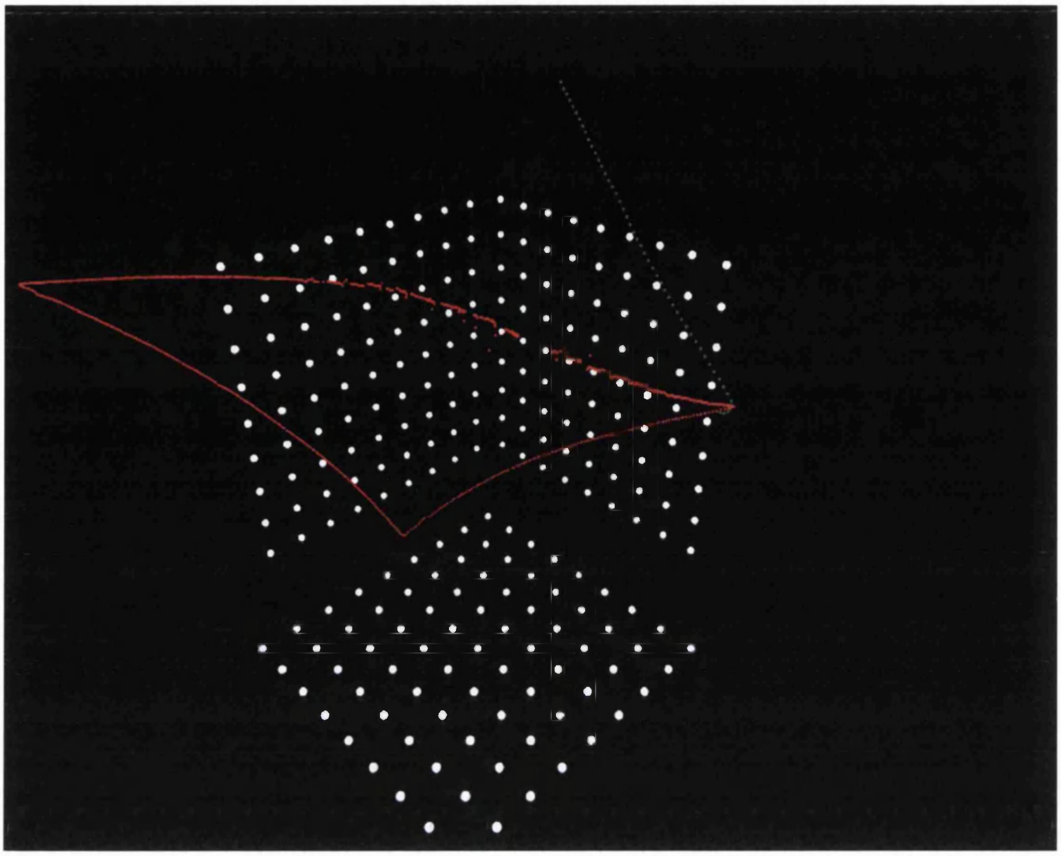
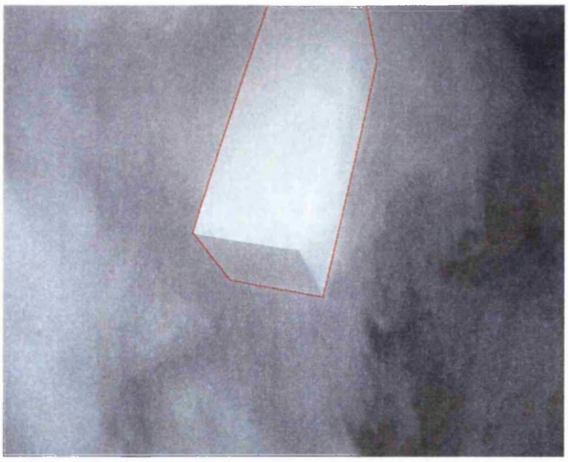
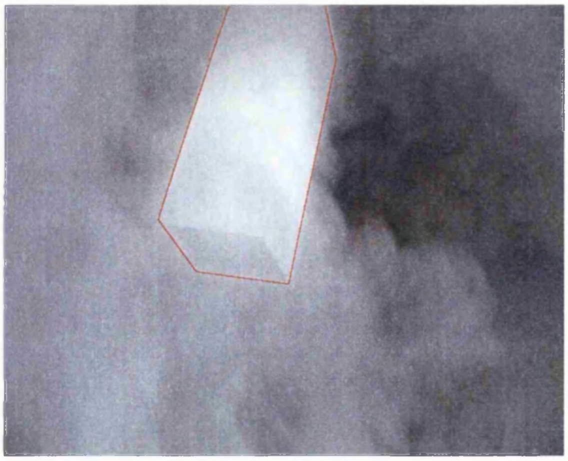


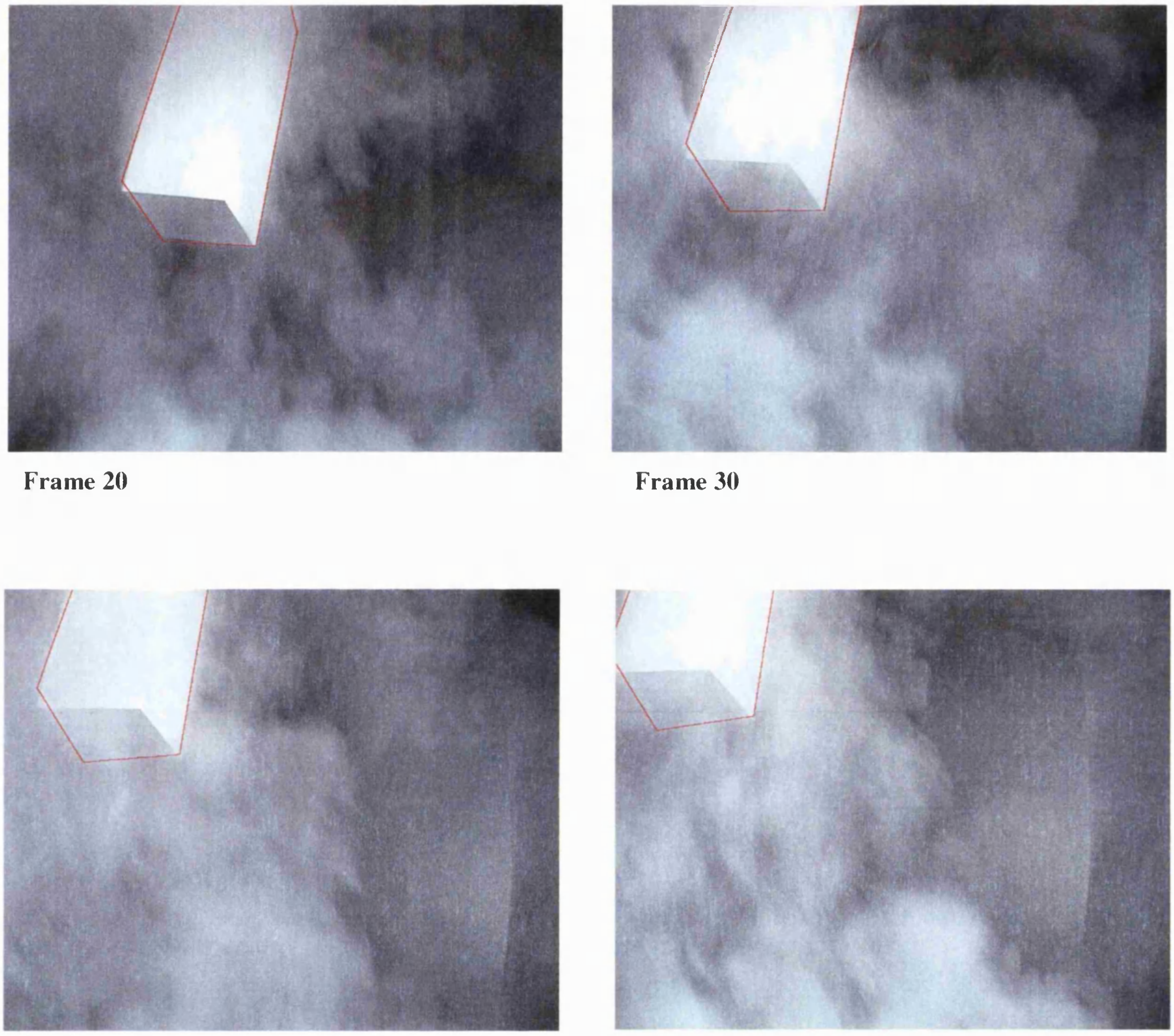
Figure 5.68 Measured trajectory output by the vision system (green) compared to ground truth trajectory (red) for a 51 frame poor visibility (dry ice fog and moving lights) image sequence along a trajectory section including a sharp corner.



Frame 0



Frame 10



Frame 20

Frame 30

Frame 40

Frame 50

Figure 5.69 **Vision system output superimposed over observed images. Sequence sampled once every ten frames. The object is tracked (2D) successfully from frame to frame throughout the sequence despite a deterioration in the 3D measurement of camera position.**

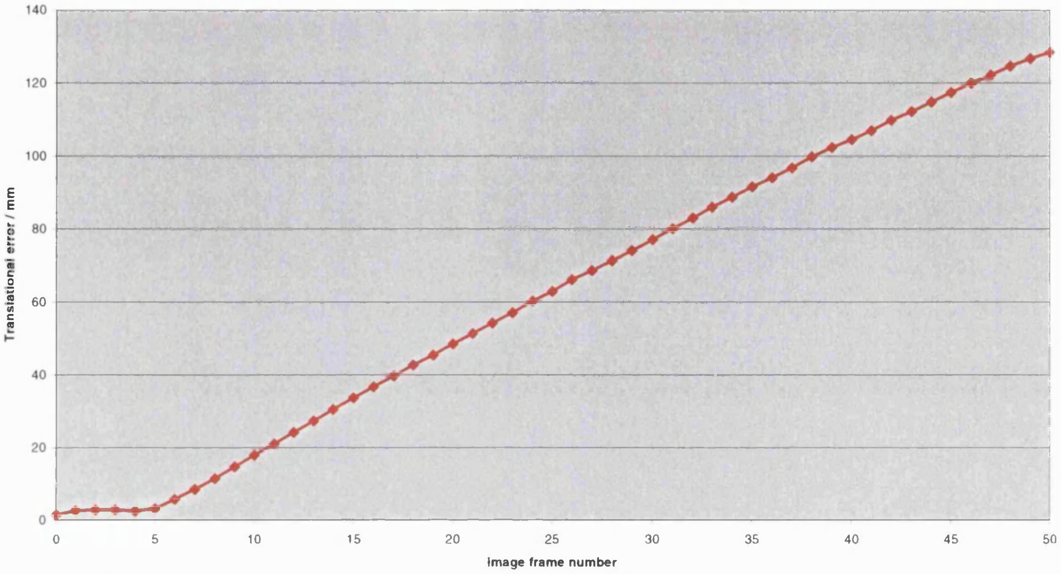


Figure 5.70 Translational error at each frame. The algorithm tracks 3D camera position very accurately for the first five frames before encountering the corner event in the camera trajectory at around the sixth frame of the sequence, afterwhich the error increases linearly with time.

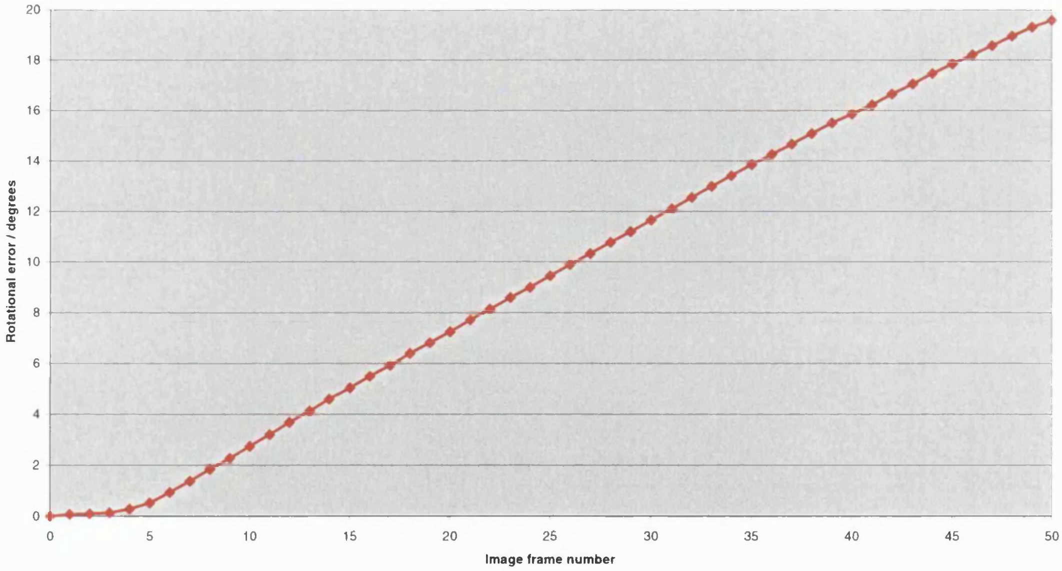


Figure 5.71 Rotational error at each frame. The algorithm tracks 3D camera position very accurately for the first five frames before encountering the corner event in the camera trajectory at around the sixth frame of the sequence, afterwhich the error increases linearly with time.

5.4.7.2 Second experiment with "block" object

This experiment (figures 5.72 to 5.75) uses a 51 frame test sequence based on a ground-truth camera trajectory that features a smoothly curved section (approximately first 25 frames) followed by a sharp corner leading to another relatively smooth portion of trajectory. The vision algorithm parameters used were a u value of 0.7 and an S_2 value of 4.0.

The algorithm exhibits partial success in tracking the 2D position of the object in each image frame over the entire sequence, although with less accuracy than in the previous example. Again, after the corner (at approximately the 25th frame), the estimated 3D position and orientation of the camera deteriorates. Whereas the errors increased linearly with time in the previous example (figures 5.70 and 5.71), in this case the estimated camera positions appear to scatter seemingly randomly even though the 2D object position is tracked (albeit somewhat clumsily) right through to the end of the sequence.

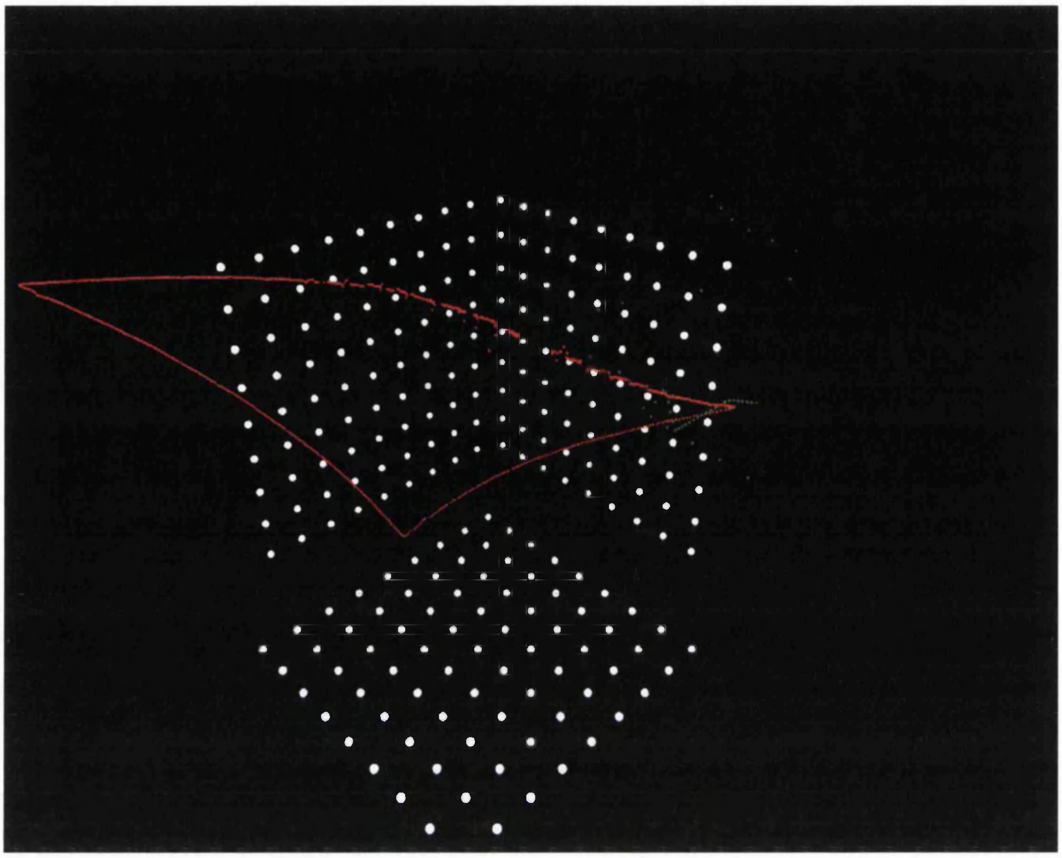
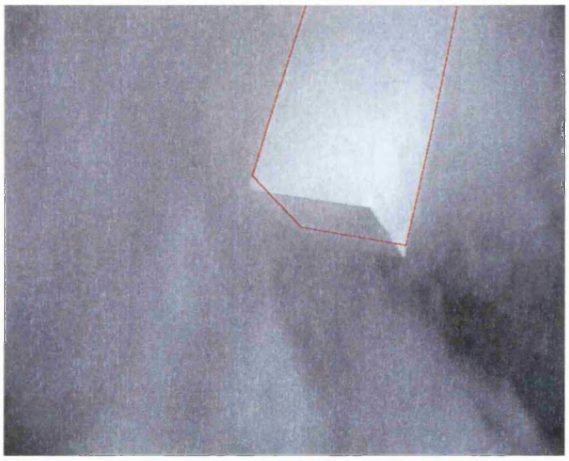
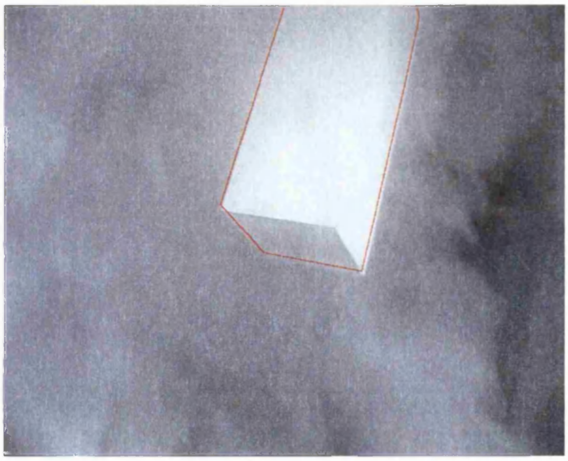


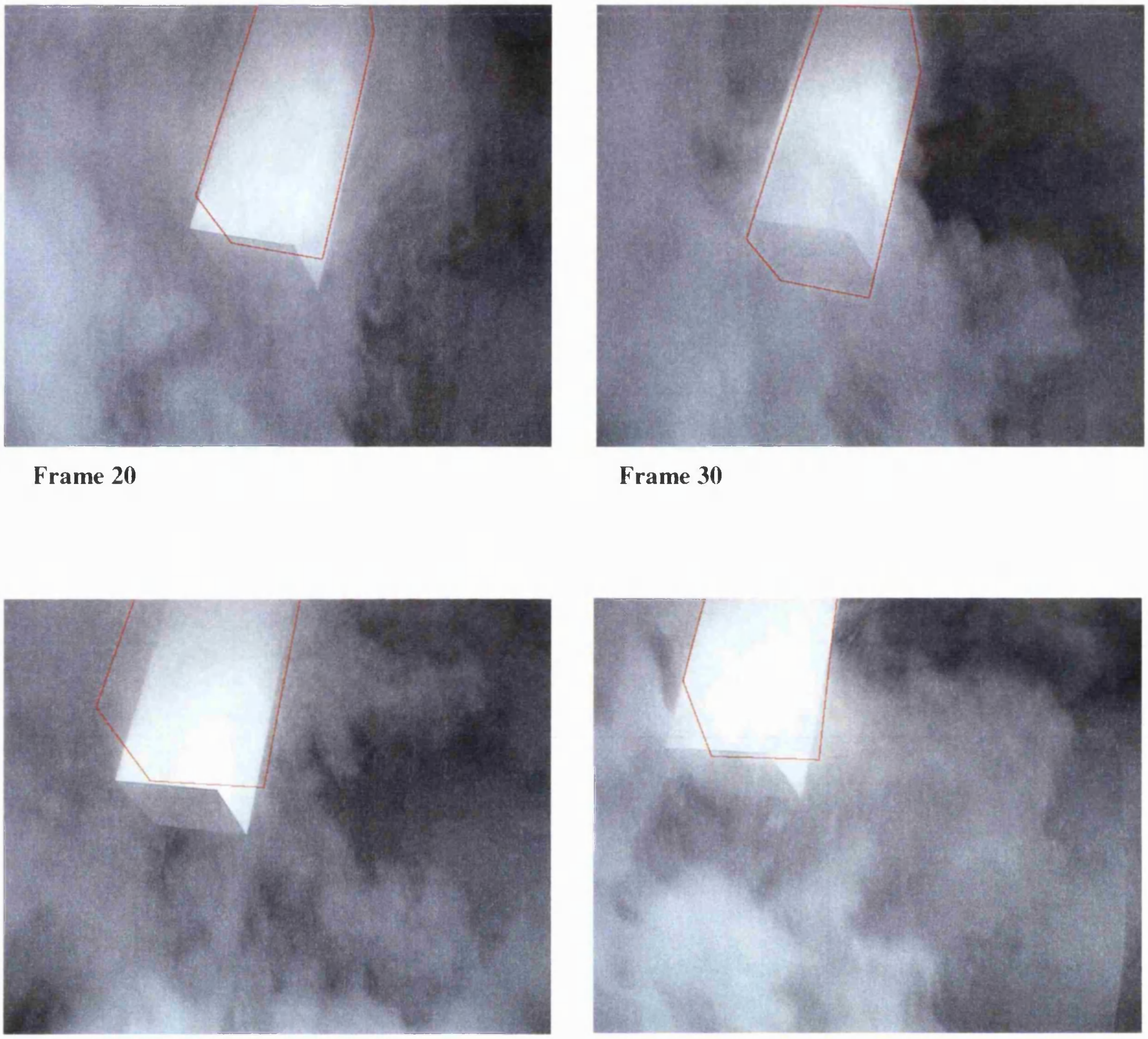
Figure 5.72 **Measured trajectory output by the vision system (green) compared to ground truth trajectory (red) for a 51 frame poor visibility (dry ice fog and moving lights) image sequence along a trajectory section including a sharp corner. The vision system tracks the 3D trajectory with reasonable success for the first 25 frames until the corner event is encountered.**



Frame 0



Frame 10



Frame 20

Frame 30

Frame 40

Frame 50

Figure 5.73 **Vision system output superimposed over observed images. Sequence sampled once every ten frames. The "object" region of the image is tracked (2D) successfully from frame to frame throughout the sequence despite a deterioration in the 3D measurement of camera position. The errors in 2D tracking of the "object" image region deteriorate towards the end of the sequence.**

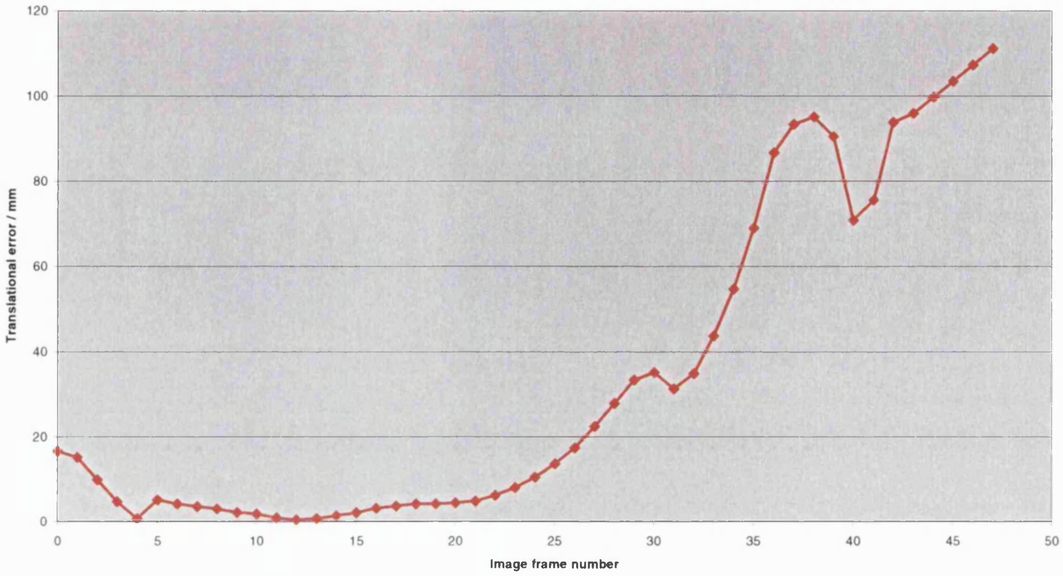


Figure 5.74 Translational error at each frame. The algorithm tracks 3D camera position relatively accurately for approximately the first 25 frames before encountering the corner event in the camera trajectory, afterwhich the error increases towards the end of the sequence.

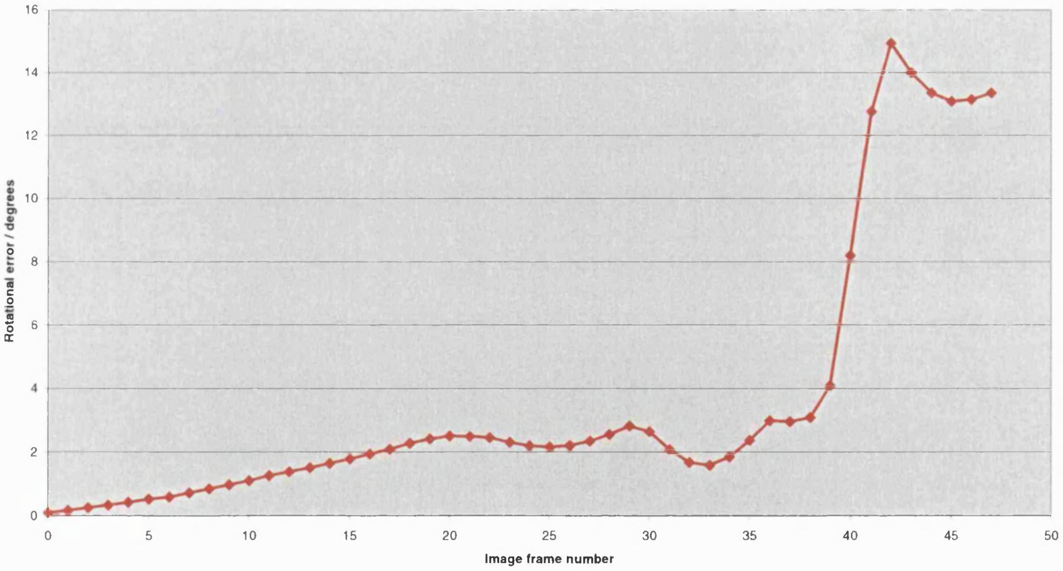


Figure 5.75 Rotational error at each frame. The algorithm tracks 3D camera orientation relatively accurately at first, deteriorating towards the end of the sequence and after the trajectory corner, despite the fact that the 2D location of the object in each image is tracked (albeit imprecisely) throughout the sequence.

5.5 Recap

Starting error	No. iterations before convergence	Improves image segmentation?	Improves position estimate?
28.4mm	3	Yes	Yes
56.8mm	4	Yes	Yes
85.2mm	0	No	No
4.2 degrees	5	Yes	No
9.0 degrees	0	No	No

Figure 5.76 Summary of performance of the EM/E-MRF algorithm on a single image frame when subjected to various different starting errors. This table summarises section 5.3.

Visibility	Trajectory	Object	S_1	S_2	U	2D tracking of object region in image?	3D tracking of camera positions?
Good	Corner	Oil rig	1.0	1.0	0.7	Very accurate.	Very accurate.
Bad	Corner	Oil rig	1.0	4.0	0.3	Initially accurate. Deteriorates with time.	Estimated trajectory drifts away from ground-truth with time.
Bad	Smooth curve	Oil rig	1.0	1.0	1.0	Fails after short period.	Fails after short period.
Bad	Smooth curve	Oil rig	1.0	3.0	1.0	Fails.	Fails.
Bad	Smooth curve	Oil rig	1.0	1.5	0.6	Tracks throughout sequence.	Tracks throughout sequence.
Bad	Smooth curve	Oil rig	1.0	1.0	0.5	Tracks throughout sequence.	Tracks throughout sequence.
Bad	Corner	Block	1.0	3.1	0.8	Very accurate throughout.	Very accurate at first, then deteriorates with time after corner.
Bad	Corner	Block	1.0	4.0	0.7	Tracks object region throughout. Accuracy deteriorates towards end of sequence.	Accurate until corner event.

Figure 5.77 Summary of performance of the EM/E-MRF algorithm on extended image sequences involving various trajectories, visibility conditions, algorithm parameters and observed objects. This table summarises section 5.4.

6 Discussion

6.1 Discussion of results

6.1.1 Quantity and quality of testing

The choice of examples, presented in chapter 5, might appear somewhat arbitrary and limited to some readers. It should be possible (although prohibitively time consuming in this case due to the slow running speed of the algorithm in its present implementation) to run the vision algorithm very large numbers of times on each image sequence and chart the variation in performance with a wide range of variations in each parameter, thus determining optimum values. However, the usefulness of this approach is limited since different sets of parameter values will be required for different video sequences under different visibility conditions. Extensive fine tuning of the algorithm for a particular data set is of little help to other researchers who may wish to apply ideas from this thesis to future engineering problems.

Similarly, when testing individual images with varying starting errors (section 5.3), it should be possible to determine the threshold error beyond which the algorithm can no longer converge on an improved position estimate. Again, this information is of little use since thresholds will be different for all images, varying with the object viewed, visibility conditions and the particular view as well as the types (orientation or translation) and directions of the starting errors.

Instead, the examples have been carefully selected in order to illustrate the various significant kinds of behaviour of the algorithm and the conditions which can cause them. In particular, the results chapter has explored the various conditions which cause the algorithm to fail, and has demonstrated the different modes of failure which result.

6.1.2 Algorithm performance on single images

It has been clearly demonstrated (sections 5.2.2. and 5.2.3) that both thresholding and conventional MRF techniques are unable to adequately segment the poor visibility test images. In contrast, segmentation by Extended-Markov Random Field produces useful results. The quality of the E-MRF segmentation is shown (section 5.3) to improve with successive iterations of the EM/E-MRF algorithm.

Section 5.2.5 illustrates how the object model is fitted to the segmented image, producing an improved camera position and orientation estimate.

Section 5.3 demonstrates that the EM/E-MRF algorithm is capable of accurately locating the (2D) position of objects in poor visibility images. It is also able to improve camera position (3D) estimates when subjected to various initial translational and rotational errors.

Occasionally (section 5.3.4) the algorithm is observed to accurately locate the object in an image (significantly improving the initial estimate of 2D object location in the image) while failing to improve or even worsening the 3D camera position estimate. This is a result of combinations of rotational and translational errors producing similar predicted images to those projected from the true camera location. This explanation for the errors is supported by the strong correlation observed between rotational and translational errors (see section 5.4.1, figures 5.48 and 5.49). These errors may be regarded as a fundamental limitation of vision systems which are based on fitting an object model to an observed image. This source of error is reduced when using objects with complex geometry and many distinctive features.

For each image frame, there will be a limiting size of starting error from which the algorithm cannot recover. This limit will vary with the geometry of the scene being viewed, the directions of the errors and the level of visibility. During experiments the algorithm was able to recover good estimates of both the object position in the image and the 3D camera position when subjected to a translational error of 56.8 mm. When subjected to a rotational error of 4.7 degrees, the algorithm was able to accurately locate (2D) the object in the image but was unable to improve the error in the 3D camera position estimate. The algorithm was unable to improve an initial position estimate with a translational error of 85.2mm or a rotational error of 9.4 degrees.

Where the initial error was too big for the algorithm to recover from, the area of overlap between the portions of the observed and predicted images containing the object were small. This caused the means of the class conditional distribution estimates to be very similar resulting in the failure of the algorithm to distinguish between object and background classes in the observed image.

6.1.3 Algorithm performance on image sequences

The algorithm was able to accurately track a 201 frame artificially created perfect visibility sequence for which the camera trajectory contained a sharp corner. The algorithm failed to negotiate sharp trajectory corners with real, poor visibility image sequences, although in two examples (section 5.4.7) the algorithm was able to continue tracking the 2D position of the object in the images even though the 3D estimate of camera trajectory deteriorated following the corner event.

The algorithm successfully tracked a 101 frame image sequence. The image sequence was filmed along a smooth trajectory in extremely poor visibility, produced by dry ice fog and moving, focussed beam lighting. The rms translational and rotational errors while tracking the sequence were 24.9mm, 3.6 degrees and 27.2mm, 3.6 degrees respectively with two different settings of algorithm parameters.

Parameters of the algorithm can be altered in order to vary the degree to which the algorithm utilises predicted data. Both the significance of predicted pixel class values during segmentation, and also the weighting during combination of predicted (by trajectory extrapolation) and observed (by the vision algorithm) camera positions, can be varied.

Excessive use of prediction results in smooth trajectories which deviate from the ground truth increasingly with time (section 5.4.2). Too little use of prediction results in instability with the estimated trajectory prone to scatter, seemingly randomly, as the algorithm fails (section 5.4.3).

The algorithm was tested with an object (the "block" object) the geometry of which is not sufficiently complex to provide distinctly different images when viewed from different directions in poor visibility. The algorithm was partially successful as a "blob" tracker, satisfactorily identifying and tracking the approximate position (2D) of the object in the image. The algorithm was partially successful at tracking the 3D position and orientation of the camera during the early parts of these sequences for which the camera trajectory was relatively smooth. The 3D tracking aspect of the algorithm failed once the camera trajectory encountered a sharp corner although the algorithm continued to track the 2D position of the observed object in each image.

6.2 Limitations of the algorithm

Currently the vision system needs to be manually initialised. Good estimates of the camera position at two successive image frames at the start of each image sequence need to be entered into the system by hand. This suggests an application to situations in which good visibility conditions suddenly deteriorate. In such a scenario, the EM/E-MRF vision system could be initialised by the position estimates derived from a conventional, good visibility vision system for the frames immediately prior to the onset of poor visibility conditions.

The EM/E-MRF algorithm will only work for images in which the mean grey-levels of object pixels and background pixels are significantly different (see sections 5.3.3 and 5.3.5). If these class means are approximately equal, the algorithm is unable to compensate for erroneous predicted data by accurately distinguishing between classes in the observed data. This makes the algorithm unsuitable for practical problems involving tracking an object with a similar colour or texture to the background. It might also have consequences for attempts to extend the algorithm to track one of several similar objects or an object against certain types of clutter.

The EM/E-MRF algorithm will only work if the initial camera position predicted for each frame is sufficiently accurate that the position (2D) of the object in the predicted image significantly overlaps true object position in the observed image (when the observed and predicted images are superimposed). A common consequence of this condition not being met is that the above condition of distinct class means is not met either (see sections 5.3.3 and 5.3.5).

The geometry of the object being viewed should be sufficiently complex that its segmented silhouette appears significantly different from different viewpoints even under poor visibility conditions. When objects are viewed that do not adequately satisfy this condition, the vision system is often unable to correctly extract 3D camera positions, even though the approximate location (2D) of the object within each image can often still be detected with some success (see section 5.4.7).

This work is limited to the case of scenes containing only a single known object of interest, for which the vision system possesses an accurate model. The problems of distinguishing between a number of different known objects or classifying a variety of unknown objects are not addressed.

6.3 Suggestions for tuning the algorithm parameters

Two important parameters of the vision system are u and S_2 (see chapters 3 and 5). These control the relative weightings between predicted data and observed data during image segmentation and also the relative levels of confidence associated with camera position estimates derived from fitting the object model to the segmented image and those derived from extrapolating the recent camera trajectory.

Both these parameters must be fine tuned to each specific application of the algorithm in order to achieve optimum performance. It is necessary to consider, not only the visibility conditions which are likely to be encountered, but also the expected nature of any camera motion. Poor visibility levels require high S_2 values (large predicted class value weighting) and low u values (low confidence in observed camera position and high reliance on extrapolated camera position), however if the camera trajectory is expected to be highly erratic, with large accelerations and rapid direction changes, then a low u value will be undesirable since extrapolated camera positions are unlikely to be correct. Note that $0 \leq u \leq 1$ and $0 \leq \mathrm{S}_2 \leq \infty$, but useful values for S_2 often lie in the range $1.0 \leq \mathrm{S}_2 \leq 4.0$.

One way to optimise these parameters would be to generate video test sequences with known ground-truth, for which both the motion and visibility conditions closely resemble those expected in the intended application. With a large number of experiments the parameters can then be modified in order to minimise, over the set of test sequences, the errors between the vision system outputs and the known ground-truth.

In practice, however, it either may not be feasible to create known ground-truth test sequences which realistically match the intended application conditions, or alternatively both the visibility and motion conditions may be highly variable. In such cases it would be desirable to enable the vision system to automatically adjust these parameters in response to varying conditions. This idea is discussed as further work in section 6.6.

6.4 Real-time issues

The EM/E-MRF vision system does not at present run in real time. Currently the software, implemented in JAVA and running on an off-the-shelf 3GHz PC, takes between several minutes and an hour to analyse a single image, depending on how many EM iterations take place before convergence. This work therefore serves as proof of principle and is not yet ready for application as a useful working system.

During this work, the computer code was designed primarily for clarity and simplicity and no attempt was made to optimise the code for speed of operation. It is likely that some rearrangement and optimisation of the code would result in improved speed.

The algorithm has so far been implemented in Java (Borland J-Builder). It is generally accepted that other languages are better suited to speed critical applications. It is likely that implementing the software in C++ would lead to a significant improvement in speed.

By far the most time consuming part of the algorithm is the non-linear optimisation of camera position estimate when best fitting the object model to the segmented image. This is computationally expensive since every increment in camera position during successive iterations of the non-linear optimisation requires a corresponding predicted image to be projected for comparison with the segmented image. During this work, Powell's method was used primarily for convenience since it was already available in a compatible coded form. Alternative optimisation strategies might usefully sacrifice accuracy of fitting and robustness for increased speed, especially since further refinement can take place during successive EM iterations.

Most algorithms can be speeded up by using dedicated hardware including programmable logic chips or "hardwiring" algorithms directly into specialised chips. Such technology could be expected to significantly increase the speed of the system.

Moore's law (Denning [1997]) expects that the speed of computers will double every eighteen months. This rate of improvement is expected to continue over the next two decades. The algorithm in its present, un-optimised form, can therefore be expected to operate in real time on a conventional PC within 20 years.

It should also be noted that speed is relative and application specific. A slowly operating algorithm performs as well with a slowly moving camera as a high speed algorithm performs with a fast moving camera. It is possible that applications for this work might arise which involve relatively slowly changing scenes.

Finally, this work has usefully demonstrated the principle that predicted data can be combined with observed data in order to enable machine vision in poor visibility conditions. There may be alternative or modified methods by which these two kinds of data can be combined which will prove better suited to real time applications.

6.5 Original contributions in this work

- The use of a predicted image (projected from an initial estimate of camera position) to estimate class conditional probability distributions and a discriminating threshold value during image segmentation.

- A method by which class condition distributions are progressively re-learned and refined, both with successive iterations during the analysis of each image and also from image to image over an extended image sequence, thus exhibiting machine learning and response to variable visibility conditions.

- Use of the E-MRF segmentation technique within an iterated feedback loop. The E-MRF is used as part of a process which outputs an improved estimate of camera position based on a less good initial position estimate. The output of this process is then recycled resulting in an Expectation Maximisation feedback process.

- Creation of test image sequences with known ground truth. These are extended real image sequences of several hundred frames, filmed along six degree of freedom camera trajectories, featuring a variety of known objects, filmed in poor visibility with known ground truth in terms of camera position and orientation at each frame in addition to camera intrinsic calibration data and a lens distortion model.

- Use of synchronised good visibility calibration sequences and poor visibility test sequences to provide ground truth. The camera positions extracted from the calibration sequence at each frame are used to provide ground truth for the corresponding images in the poor visibility test sequence.

6.6 Further work

6.6.1 Improving the vision algorithm

The image model used in the vision system is overly simplistic for two reasons:

- Class conditional distributions are modelled as uni-modal normal distributions. This is a useful approximation since the true class-conditional histograms are often uni-modal and bell shaped (see figure 5.3, section 5.2.2). However, in the presence of focussed beam spotlights and severe back-scattering, both the object being viewed and the background may at times become multi-modal since regions of the (mostly bright) object can appear very dark and regions of the (mostly dark) background can appear very bright.

- The same class conditional distributions are assumed for all regions in the image. This assumption is not always valid since both lighting and visibility conditions can vary with position in the image.

It is therefore suggested that, firstly, class conditional distributions be modelled as multi-modal, possibly using histograms or some form of Gaussian mixture model, and secondly, that an adaptive method be adopted such that these distributions can vary with image position. Ideally, each pixel should be modelled with a unique pair of class conditional distributions which are based on data from a local region surrounding that pixel. The size of this region will be crucial in its effect on generality and over-fitting or under-fitting of the image model to the observed data.

It is also apparent that more use can be made of predicted data. Not only can the position of the camera at the next frame be estimated, but it should also be possible to estimate the forms of the class conditional distributions at the next frame. It may prove useful to allow the form of class conditional distributions to be influenced by those of the previous frame. This might help from the point of view of reducing noise and also might improve computation speed by providing a good starting point for any process that refines distribution estimates.

Two important parameters of the vision system are u and S_2 (see chapters 3 and 5). These control the relative weightings between predicted data and observed data and the confidence associated with extrapolated camera positions. Ideally these values should be self tuning and automatically adjusted as visibility conditions vary (see section 6.3). Such a system would require a method for detecting and quantifying “goodness of visibility”. Different kinds of image degradation (attenuation, occlusion, back-scatter) may require different settings. This is a complex problem without obvious solutions. One possible approach would be to measure the difference or separability of object from background for each frame after segmentation. This measurement could then be used to determine the levels of prediction used to tackle the next frame in the sequence. Perhaps Fisher’s discriminant ratio (the square of the difference between class means divided by the sum of class variances) could be used as a separability measure. Similarly, a measure for “trajectory smoothness” should be considered, and this could be computed from recently measured camera positions.

Another important parameter is S_1 (see chapters 3 and 5), which determines the significance of the spatial portion of the MRF neighbourhood (i.e. significance of the class of nearest neighbour pixels). During this work, attention was focussed on investigating different values of S_2 which determines the significance of a predicted pixel class. Comparatively little attention was paid to S_1, which was set to 1.0 on the basis of a few experiments with a small number of images. Further work needs to be undertaken to determine optimum relative weightings between S_1 (spatial), S_2 (predictive) and the class conditional components during segmentation.

As has been mentioned previously (section 6.3), the speed of the algorithm might be improved by using an alternative to Powell's method for the non-linear optimisation involved in best-fitting the object model to the segmented image. An alternative optimisation strategy might profitably sacrifice accuracy of fitting for improved speed, especially since further refinement is always possible during later EM iterations. Possible modifications might include limiting the number of iterations of the optimisation algorithm and enlarging the minimum step size that the algorithm can move along each parameter.

6.6.2 Improvements to data set development

In chapter 4 (section 4.4.8) several suggestions were made as to how the practical process of generating test sequences might be improved. These are summarised as follows:

- The robot trajectory should be programmed such that the camera has a good view of all three targets in every frame. This reduces the errors caused by translation/rotation equivalence.

- The camera should be calibrated from a set of images filmed at a variety of different ranges from the targets. This prevents over-fitting to points lying in the target planes and under-fitting to points in the space outside of those planes.

- The objects to be viewed should be constructed such that they largely fill the volume of space within the calibration targets. This prevents error magnification when the camera moves in from a wide view of the targets to a close-up view of a relatively small object.

Additionally, it would be useful to define a measure of visibility and a more systematic way to classify and compare different kinds of image degradation. Ideally image sequences should be filmed at a range of specific and consistent visibility levels. It might be possible to maintain relatively consistent visibility conditions by means of a fog chamber. An image sequence in which visibility gradually degenerates with time would also be a useful way of observing the visibility level at which the vision system begins to fail.

6.6.3 Improvements to testing and analysis

In chapter 5, errors were presented in terms of translational and rotational components relative to a world co-ordinate system. It might also be useful to visualise errors with respect to a co-ordinate frame set in the camera itself. Thus errors could be expressed in terms of role, pitch and yaw of the camera and translations parallel to the image plane (up-down, left-right) and the optical axis of the camera (forwards-backwards or "range"). The error source identified in section 5.3.4 (see figure 5.44) could be better investigated and understood by seeking correlation between specific directions of rotational and translational error relative to the camera. Errors in terms of a co-ordinate system set in the camera should be computable from existing data.

As mentioned in the previous section, it would be desirable to create new test sequences featuring various different consistent and distinct visibility levels. If this could be achieved, the algorithm could be systematically tested to determine at what level of poor visibility it fails. These experiments could also cover various different specific and distinct kinds of image degradation.

6.7 Summary

Throughout the history of computer vision research, object recognition and tracking algorithms have been developed predominantly for good visibility applications. These algorithms typically rely on detecting edges, lines and corners of the object being observed. Such systems, dependent on identifying detailed features, are unsuitable for conditions of extremely poor visibility which are often encountered in the real world and under which the human visual system is often capable of functioning successfully.

This thesis presents a novel algorithm for the interpretation of scene content and camera position from extremely poor visibility images. The algorithm is capable of tracking camera trajectories over extended image sequences under conditions of extremely poor visibility.

The algorithm combines observed data (the current image) with predicted data derived from prior knowledge of the object being viewed and an estimate of the camera's motion.

It has been shown that an Extended-Markov Random Field technique can be used to combine these two kinds of data. The E-MRF extends Markov dependency to include contributions from corresponding pixels in a predicted image. It has also been shown how interpretations of scene content and camera position can be mutually improved using Expectation-Maximisation.

The resulting algorithm exhibits elements of continuous machine learning. Statistical image models are continuously relearned, both during the analysis of each frame and also with successive frames over an entire image sequence. The algorithm is therefore able to adapt to changing visibility conditions.

Suggestions have been made for ways in which the algorithm might be improved by increasing the generality of the statistical image model and by allowing certain parameters of the algorithm to vary automatically with changing visibility conditions.

Poor visibility image sequences of known objects, filmed along pre-measured trajectories with a calibrated camera have been constructed in order to provide test data with underlying ground truth. Using this data, the EM/E-MRF algorithm has been tested on a large number of images, over a range of visibility conditions, camera trajectories, algorithm parameters and observed objects.

The algorithm has been shown to accurately segment poor visibility images given a range of errors in the initial camera position estimate for those images. The camera position for these images is recovered. Various sources of error have been identified and explored and some important failure behaviours of the algorithm have been illustrated.

The algorithm has been tested on extended image sequences including examples for which the camera moved on both smooth trajectories and trajectories containing abrupt changes of direction. Both poor visibility real image sequences and artificially created good visibility sequences have been tested. Sequences containing objects of both distinctive, complicated geometry and also overly simple geometry were used. The performance of the algorithm has been investigated both in response to these different kinds of image sequence and also in response to varying key parameters of the algorithm.

Calibration data

The following calibration data were computed from the calibration sequence described in chapter 4 and apply to all the test sequences used in the thesis.

Intrinsic camera parameters

$$\begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 790.18 & 0 & 361.1 \\ 0 & 869.81 & 313.13 \\ 0 & 0 & 1 \end{bmatrix}$$

u_0, v_0 are co-ordinates of the principal point (in pixels).

α , β are equal to focal length multiplied by number of pixels per unit length (in u and v directions respectively, units in pixels).

γ is a measure of skewness between u and v directions in the pixel array, which in this work was assumed to be square.

Lens distortion parameters

k_1 = -3.475×10^{-7}

k_2 = 2.0335×10^{-13}

such that:

$$\hat{u} = u + (u - u_0)(k_1 r^2 + k_2 r^4)$$

and

$$\hat{v} = v + (v - v_0)(k_1 r^2 + k_2 r^4)$$

where

$$r^2 = (u - u_0)^2 + (v - v_0)^2$$

and (u,v) and $(\hat{u},\hat{v})$ are the pixel co-ordinates on a true pinhole image and a radially distorted image respectively.

Target relations transformations

These are rigid body transformations which map the spot co-ordinates from one target onto another.

Transformations are expressed in the form $(x \quad y \quad z \quad \omega_1 \quad \omega_2 \quad \omega_3)$ where the first three numbers represent translation (in mm) and the second three numbers are a vector whose direction is that of the axis of rotation and whose magnitude is the angle of rotation (in radians) about that axis.

All transformations are relative to the world co-ordinate system, chosen to be that of the base target (target 1). See figure 4.8, section 4.2.4 for target designation, layout and co-ordinate axes.

Transformation from target 2 to target 1
(251.2, -38.2, -5.3, -0.024, 2.221, 2.222)

Transformation from target 3 to target 1
(-38.7, -52.2, -4.2, 1.199, 1.205, 1.212)

References

Agapito [2001] L. Agapito, E. Hayman, I. Reid. Self-Calibration of Rotating and Zooming Cameras. *International Journal of Computer Vision.* Vol. 45(2), pages 107-127. 2001.

Barun [1999] V. Barun, A. Ivanov. Nontraditional features in active vision through a turbid medium: evaluation and optimization on the base of modern radiative transfer approaches. *Proc. SPIE International Society of Optical Engineering.* Vol. 3837, pages 414-425. 1999.

Besag [1974] J. Besag. Spatial Interaction and the Statistical Analysis of Lattice Systems. *Journal of the Royal Statistical Society, series B.* Vol. 36, pages 192-236. 1974.

Besag [1986] J. Besag. On the Statistical Analysis of Dirty Pictures. *Journal of the Royal Statistical Society, series B.* Vol. 48, pages 259-302. 1986.

Besl [1985] P. Besl, R. Jain. Three-Dimensional Object recognition. *ACM Computing Surveys.* Vol. 17(1). 1985.

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

Bishop [1995] C. Bishop. Neural Networks for Pattern Recognition. Oxford University Press. 1995.

Blake [2000] A. Blake. Probabilistic Inference and Learning in Computer Vision. *11th British Machine Vision Conference.* Pre-Conference Tutorial. 2000.

Bouthemy [1988] P. Bouthemy, P. Lalande. Determination of apparent mobile areas in an image sequence for underwater robot navigation. *Proceedings of IAPR Workshop on Computer Vision: Special Hardware and Industrial Applications.* Pages 409-412. 1988.

Bouthemy [1989] P. Bouthemy, P. Lalande. Motion detection in an image sequence using Gibbs distributions. *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing.* Vol. 3, pages 1651-1654. 1989.

Braud [1994] Modelled object pose estimation and tracking by a multi-cameras system. *Proc. IEEE Compuer Society Conference on Computer Vision and Pattern Recognition.* Pages 976-979. 1994.

Brown [1971] D. Brown. Close-range camera calibration. *Photogrammetric Engineering.* Vol. 37(8), pages 855-866. 1971.

Chow [1972] C. Chow, T. Kaneko. Automatic boundary detection of the left ventricle from cineangiograms. *Computing in Biomedical Research.* Vol. 5, pages 338-410. 1972.

Christmas [1996] W. Christmas, J. Kittler, M. Petrou. Error propagation for 2D-to-3D matching with application to underwater navigation. *Proc 7th British Machine Vision Conference.* Pages 555-564. 1996.

Cootes [1997] T. Cootes, C. Taylor. A Mixture Model for Representing Shape Variation. *Proc 8^{th} British Machine Vision Conference.* Pages 110-119. 1997.

Dempster [1977] A. Dempster, N. Laird, D. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society.* Vol. B(39), pages 1-38. 1977.

Denning [1997] P. Denning, R. Metcalfe. Beyond Calculation: The Next 50 Years in Computing. Copernicus Publishing. ISBN 0-387-98588-3. 1997.

Derin [1985] H. Derin. The use of Gibbs distributions in image processing. *Communications and Networks: A Survey of Recent Advances.* I. Blake, V. Poor, Eds. Springer-Verlag. 1985.

Derin [1986] H. Derin, W. Cole. Segmentation of textured images using Gibbs random fields. *Computer Vision, Graphics and Image Processing.* Vol. 35, pages 72-98. 1986.

Deutscher [2000] J. Deutscher, A. Blake, I. Reid. Articulated body motion capture by annealed particle filtering. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition.* Vol. 2, pages 126-133. 2000.

Deutscher [2001] J. Deutscher, A. Davison, I. Reid. Automatic partitioning of high dimensional search spaces associated with articulated body motion. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition.* Vol. 2, pages 11669-11676. 2001.

Drummond [1999] T. Drummond, R. Cipolla. Real-Time tracking of complex structures with on-line camera calibration. *Proc. 10th British Machine Vision Conference,* 1999.

Drummond [2000] T. Drummond, R. Cipolla. Real-Time Tracking of Multiple Articulated Structures in Multiple Views. *European Conference on Computer Vision.* 2000.

Dubes [1990] R. Dubes, A. Jain, S. Nadabar, C. Chen. MRF Model-Based Algorithms For Image Segmentation. *Proceedings IEEE 10th International Conference on Pattern Recognition.* Pages 808-814. 1990.

Faig [1975] W. Faig. Calibration of close-range photogrammetry systems: Mathematical formulation. *Photogrammetric Engineering and Remote Sensing.* Vol. 41(12), pages 1479-1486. 1975.

Fairweather [1997a] A. Fairweather. Robust Interpretation of Underwater Image Sequences. *PhD Thesis, University of London.* 1997.

Fairweather [1997b] A. Fairweather, M. Hodgetts, A. Greig. Robust Interpretation of Underwater Image Sequences. *Image Processing and its Applications.* Pages 660-664. 1997.

Faugeras [1986a] O. Faugeras, G. Toscani. The calibration problem for stereo. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* Pages 15-20. 1986.

Faugeras [1986b] O. Faugeras, M. Hebert. The representation, Recognition, and Locating of 3-D Objects. *The International Journal of Robotics Research.* Vol.5(3), 1986.

Faugeras [1992] O. Faugeras, T. Luong, S. Maybank. Camera self-calibration: theory and experiments. *Proceedings of the 2nd European Conference on Computer Vision* in *Lecture Notes in Computer Science*. Springer-Verlag. Pages 321-334. 1992.

Fitzgibbon [1998] A. Fitzgibbon, A. Zisserman. Automatic camera recovery for closed or open image sequences. *Lecture Notes in Computer Science*. Vol. 1406, p 311. 1998.

Foresti [2001] G. Foresti. Visual Inspection of Sea Bottom Structures by an Autonomous Underwater Vehicle. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics.* Vol. 31, No. 5, 2001.

Fu [1981] K. Fu, J. Mui. A survey on image segmentation. *Pattern Recognition.* Vol. 13, pages 3-16. 1981.

Ganapathy [1984] S. Ganapathy. Decomposition of transformation matrices for robot vision. *Pattern Recognition Letters.* Vol 2, pages 401-412. 1984.

Geman [1984] S. Geman, D. Geman. Stochastic Relaxation: Gibbs Distributions and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* Vol. 9, pages 721-741. 1984.

Gennery [1979] D. Gennery. Stereo-camera calibration. *Proceedings of the 10th Image Understanding Workshop*. Pages 101-108. 1979.

Gennery [1981] D. Gennery. A feature-based scene matcher. *Proc. 7th International Joint Conference on Artificial Intelligence.* Pages 667-673. 1981.

Gennery [1982] D. Gennery. Tracking known three-dimensional objects. *Proc. 2nd National Conference on Artificial Intelligence.* Pages 13-17. 1982.

Ghosh [1991] A. Ghosh, N. Pal, S. Pal. Image segmentation using a neural network. *Biological Cybernetics.* Vol. 66, pages 151-158. 1991.

Gibbs [1902] W. Gibbs. Elementary Principles of Statistical Mechanics. Yale University Press. 1902.

Ginhoux [2001] R. Ginhoux, J. Gutmann. Model-based object tracking using stereo vision. *Proc. IEEE International Conference on Robotics and Automation.* Vol. 2, pages 1226-1232. 2001.

Gracias [2001] N. Gracias, J. Santos-Victor. Trajectory reconstruction with uncertainty estimation using mosaic registration. *Robotics and Autonomous Systems.* Vol 35, pages 163-177. 2001.

Greig [1996] A. Greig, S. Lovell. Improvement of the accuracy of a PUMA 560 industrial manipulator by calibration. *Journal of Engineering Manufacture, Proc. I.Mech.E.*, Part B. Vol. 210, pp. 55 - 67. 1996.

Grimson [1998] E. Grimson, C. Stauffer, R. Romano, L. Lee. Using adaptive tracking to classify and monitor activities in a site. *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition.* Pages 22-29. 1998.

Grimson [2000] E. Grimson, M. Leventon, O. Faugeras, W. Wells. Computer Vision Methods for Image Guided Surgery. *Proc. 11th British Machine Vision Conference.* Pages 1-12. 2000.

Hall [1992] L. Hall, A. Bensaid, L. Clarke, R. Velthuzien, M. Silbiger, J. Bezdek. A Comparison of Neural Network and Fuzzy Clustering Techniques in Segmenting Magnetic Resonance Images of the Brain. *IEEE Transactions on Neural Networks.* Vol. 3(5). 1992.

Haralick [1985] R. Haralick, L. Shapiro. Survey, image segmentation techniques. *Computer Vision, Graphics and Image Processing.* Vol. 29, pages 100-132. 1985.

Harkness [2000] M. Harkness, P. Green. Parallel Chains, Delayed Rejection and Reversible Jump MCMC for Object Recognition. *Proc.11th British Machine Vision Conference.* 2000.

Harris [1992] C. Harris. Tracking with Rigid Bodies. *Active Vision.* A. Blake ed. Chapter 4, pages 59-73, MIT press. 1992.

Hartley [2000] R. Hartley, A. Zisserman. Multiple View Geometry in Computer Vision. *Cambridge University Press.* 2000.

Hilton [2000] A. Hilton, B. Daniel, G. Thomas, R. Smith, S. Wei, J. Illingworth. Whole-body modelling of people from multiview images to populate virtual worlds. *Visual Computer.* Vol. 16 (7), pages 411-436. 2000.

Hodgetts [1999] M. Hodgetts, A. Greig, A. Fairweather. Underwater Imaging Using Markov Random Fields with Feed Forward Prediction. *Journal of the Society for Underwater Technology.* Vol. 23(4), pages 157-167. 1999.

Ioffe [1999] S. Ioffe, D. Forsyth. Finding people by sampling. *Proceedings of the IEEE International Conference on Computer Vision.* Vol. 2, pages 1092-1097. 1999.

Isard [1996] M. Isard, A. Blake. Visual tracking by stochastic propagation of conditional density. *Proc. 4th European Conference on Computer Vision.* Pages 343-356. 1996.

Isard [1998] M. Isard, A. Blake. CONDENSTION-Conditional Density Propagation for Visual Tracking. *International Journal of Computer Vision.* Vol. 29(1), pages 5-28. 1998.

Ising [1925] E. Ising. *Zeitschrift Physik.* (In German). Vol. 31. 1925.

Jurie [1997] F. Jurie. Model-based object tracking in cluttered scenes with occlusions. *IEEE International Conference on Intelligent Robots and Systems.* Vol. 2, pages 886-892. 1997.

Kalman [1960] R. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME-Journal of Basic Engineering.* Pages 35-45, 1960.

Kamber [1992] M. Kamber, D. Collins, R. Shinghal, G. Francis, A. Evans. Model-based 3D segmentation of multiple sclerosis lesions in dual-echo MRI data. *Visualisation in Biomedical Computing.* Pages 590-600, SPIE Vol. 1808. 1992.

Kaneda [1991] K. Kaneda, T. Okamoto, E. Nakamae, T. Nishita. Photorealistic image synthesis for outdoor scenery. *The Visual Computer.* Vol. 7, pages 247-258. 1991.

Keller [1990] J. Keller, C. Carpenter. Image Segmentation in the Presence of Uncertainty. *International Journal of Intelligent Systems.* Vol. 5, pages 193-208. 1990.

Kittler [1985] J. Kittler, J. Illingworth, J. Foglein. Threshold selection based on a simple image statistic. *Computer Vision, Graphics and Image Processing.* Vol. 30, pages 125-147, 1985.

Kosaka [1995] A. Kosaka, G. Nakazawa. Vision-based motion tracking of rigid objects using prediction of uncertainties. *Proc. IEEE International Conference on Robotics and Automation.* Vol. 3, pages 2637-2644. 1995.

Lacey [2001] A. Lacey, N. Thacker, P. Courtney, S. Pollard. TINA 2001: The Closed Loop 3D Model Matcher. *Proceedings of the 12th British Machine Vision Conference.* Pages 203-212. 2001.

Lowe [1992] D. Lowe. Robust model-based motion tracking through the integration of search and estimation. *International Journal of Computer Vision.* Vol. 8(2), pages 441-450, 1992.

Markov [1906] A. Markov. Extension of the law of large numbers to dependent events (in Russian), *Bull. Soc. Phys. Math. Kazan.* Vol. 2 (15), pages 155-156. 1906.

Maybank [1992] S. Maybank, O. Faugeras. A theory of self-calibration of a moving camera. *The International Journal of Computer Vision.* Vol. 8(2), pages 123-152. 1992.

McCane [2001] B. McCane, K. Novins, D. Crannitch, B. Galvin. On Benchmarking Optical Flow. *Computer Vision and Image Understanding.* Vol. 84, pages 126-143. 2001.

Mokhtarian [2000] F. Mokhtarian, N. Khalili, P. yuen. Free-form 3-D Object Recognition at Multiple Scales. *Proc. 11th British Machine Vision Conference.* 2000.

Nakagawa [1979] Y. Nakagawa, A. Rosenfeld. Some experiments on variable thresholding. *Pattern Recognition.* Vol. 11, pages 191-204. 1979.

Narasimhan [2002] S. Narasimhan, S. Nayar. Vision and the Atmosphere. *International Journal of Computer Vision.* Vol. 48(3), pages 233-254, 2002.

Neal [1993] R. Neal, G. Hinton. A New View of the EM Algorithm that Justifies Incremental and Other Variants. *Biometrika.* 1993.

Nishita [1987] T. Nishita, Y. Miyawaki, E. Nakamae. A shading model for atmospheric scattering considering luminous intensity distribution of light sources. *Siggraph '87.* 1987.

Nocedal [1999] J. Nocedal, S. Wright. Numerical Optimization. Springer. 1999.

North [1997] B. North, A. Blake. Using Expectation-Maximisation to Learn Dynamical Models from Visual Data. *Proc 8th British Machine Vision Conference.* Pages 669-679. 1997.

Otte [1994] M. Otte, H.-H. Nagel. Optical Flow estimation: Advances and Comparisons. *Proc. 3rd European Conference on Computer Vision.* Pages 51-60. 1994.

Otte [1995] M. Otte, H.-H. Nagel, Estimation of optical flow based on higher-order spatiotemporal derivatives in interlaced and non-interlaced image sequences. *Artificial Intelligence.* Vol. 78, pages 5-43. 1995.

Pal [1993] N. Pal, S. Pal. A review on image segmentation techniques. *Pattern Recognition.* Vol. 26(9), pages 1277-1294. 1993.

Paul [1981] R. Paul. Robot Manipulators: Mathematics, Programming and Control. MIT Press. ISBN 0-262-16082-X. 1981.

Press [1992] W. Press, S. Teukolsky, W. Vetterling, B. Flannery. Numerical Recipes in C. 2nd Edition. Cambridge University Press. ISBN 0 521 43108 5. 1992.

Ripley [1996] B. Ripley. Pattern Recognition and Neural Networks. Cambridge University Press. 1996.

Rokita [1997] P. Rokita. Simulating Poor Visibility Conditions Using Image Processing. *Real-Time Imaging,* 3, pages 275-281. 1997.

Sahoo [1988] P. Sahoo, S. Soltani, A. Wong, Y. Chen. A survey of thresholding techniques. *Computer Vision, Graphics and Image Processing.* Vol 41, pages 233-260. 1988.

Smith [1997] S. Smith and J. Brady. SUSAN - a new approach to low-level image processing. *International Journal of Computer Vision.* Vol. 23(1), pages. 45-78. 1997.

Stolkin [2000] R. Stolkin, M. Hodgetts, A. Greig. An EM/E-MRF Strategy for Underwater Navigation. *Proc.11th British Machine Vision Conference.* 2000.

Tarassenko [1998] L. Tarassenko. A Guide to Neural Computing Applications. Butterworth-Heinemann. 1998.

Torr [1999] P. Torr, A. Fitzgibbon, A. Zisserman. Problem of degeneracy in structure and motion recovery from uncalibrated image sequences. *International Journal of Computer Vision*. Vol. 32 (1), pages 27-44. 1999.

Tsai [1987] R. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf tv camera and lenses. *IEEE Journal of Robotics and Automation*. Vol. 3(4), pages 323-344. 1987.

Ullman [1996] S. Ullman. High-level Vision. The MIT Press. 1996.

Verghese [1988] G. Verghese, C. Dryer. Real–time model-based tracking of three-dimensional objects. *Technical report 806, University of Wisconsin, Computer Sciences*. 1988.

Verghese [1990] G. Verghese, K. Gale, C. Dryer. Real-time, parallel-motion tracking of three-dimensional objects from spatiotemporal image sequences. *Parallel Algorithms for Machine Intelligence and Vision*. Kumar et al. eds., Springer-Verlag. 1990.

Watkins [2000] W. Watkins, D. Tofsted, V. CuQlock-Knopp. Navigation through fog using stereoscopic active imaging. *Proc. SPIE International Society of Optical Engineering*. Vol. 4023, pages 20-28. 2000.

Watt [1992] A. Watt, M. Watt. Advanced Animation and Rendering Techniques Theory and Practice. Academic Press. 1992.

Wei [1993] G. Wei, S. Ma. A complete two-plane camera calibration method and experimental comparisons. *Proceedings of the*

Fourth International Conference on Computer Vision. Pages 439-446. 1993.

Welch [2002] G. Welch, G. Bishop. An Introduction to the Kalman Filter. *Technical Report, Department of Computer Science, University of North Carolina at Chapel Hill.* Updated: March 11, 2002.

Wells [1996] W. Wells, W. Grimson, R. Kikinis, F. Jolesz. Adaptive Segmentation of MRI Data. *IEEE Transactions on medical imaging*. Vol. 15 (4). 1996.

Weng [1992] J. Weng, P. Cohen, M. Herniou. Camera calibration with distortion models and accuracy evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 14(10), pages 965-980. 1992.

Wunsch [1996] P. Wunsch, G. Hirzinger. Registration of CAD-Models to Images by Iterative Inverse Perspective Matching. *Proceedings of the 13th International Conference on Pattern Recognition*. Pages 77-83. 1996.

Zhang [1998] Z. Zhang. A Flexible New Technique for Camera Calibration. *Microsoft Research Technical Report,* MSR-TR-98-71. 1998.

Zhang [2000] J. Zhang, P. Fieguth, D. Wang. Random Field Models. *Handbook of Image and Video Processing*. Pages 301-312, Academic Press. 2000.

Zisserman [1999] A. Zisserman, A. Fitzgibbon, G. Cross. VHS to VRML: 3D graphical models from video sequences. *International*

Conference on Multimedia Computing and Systems-Proceedings. Vol. 1, pages 51-57. 1999.