UNCALIBRATED STEREO AND HAND–EYE COORDINATION

Nicholas John Hollinghurst
Trinity Hall
January 1997

dissertation submitted to the University of Cambridge
towards the degree of Doctor of Philosophy

Department of Engineering
University of Cambridge

Uncalibrated Stereo and Hand–Eye Coordination

Nicholas John Hollinghurst

Trinity Hall, Cambridge

January 1997

Summary

This dissertation describes new applications of uncalibrated and weakly calibrated stereo vision to facilitate pick-and-place operations by a robot manipulator.

A 'weakly calibrated' stereo rig is one for which only a small number of reference observations have been made (for instance, by observing the robot itself making deliberate motions) and which might be subject to vibrations and small movements during use. Thus the epipolar geometry and camera parameters will be known only approximately. In such an environment, it is shown that an approximate linear model (the affine camera) is well suited to estimating both the epipolar constraint, and the relation between image measurements and the robot's coordinate system (the hand–eye relation).

The stereo system is used to track a pointing hand, implementing a vision-based user interface which allows the operator to specify objects to be grasped and to guide the robot's motion around the workspace. By considering only the plane projectivities between the images and a ground plane, it is shown that points on the plane may be indicated without calibration.

A novel stereo algorithm is developed to match line segments in weakly calibrated views and recover a description of the planar surfaces of objects in the robot's workspace. These can then be reconstructed in an approximate metric frame for grasp planning.

The tracking system employed in this project is a novel type of edge-seeking active contour, based on a template which can deform only affinely in the images. This can be used for tracking the operator's hand, the robot's gripper, and planar facets of objects in the workspace.

By tracking the robot itself, visual feedback can be employed to align the robot's gripper accurately with the surface to be grasped, even in the face of disturbances to the stereo cameras or the robot's control systems. Visually guided grasping is implemented in real time on standard hardware.

Uncalibrated Stereo and Hand–Eye Coordination

Nicholas John Hollinghurst

Trinity Hall, Cambridge
January 1997

Declaration

- This dissertation is submitted to the University of Cambridge towards the requirements for the degree of Doctor of Philosophy.
- This dissertation was composed entirely by myself. Except where otherwise noted, it describes my own original research and contains nothing which is the result of work done in collaboration.
- No part of this dissertation has been or is currently being submitted for any other university degree or diploma.
- The dissertation contains 52 figures and approximately 33000 words.

Acknowledgements

I am most grateful to my supervisor, Roberto Cipolla, for many ideas and suggestions, sound advice, and his constant encouragement throughout the project.

Over the last 4 years I have gained insight from conversations with several other people at CUED, particularly Antranig Basman, Tat-Jen Cham, Andrew Gee, Gabriel Hamid, Jonathan Lawn, Joan Lasenby and Jun Sato. I thank them for many interesting discussions and helpful suggestions.

Thanks to Elizabeth Guild, who helped me plan a path around a number of obstacles.

I would also like to thank my parents, and Mark, for all their love and support whilst I was grappling with this dissertation.

This project was funded by an EPSRC studentship and research grant GR/K08635. The robot manipulator was kindly donated to the Speech, Vision and Robotics group by Olivetti Research Ltd.

Contents

List of Figures

List of Tables

Glossary of symbols

Chapter 2

X, Y, Z	World coordinates of a point in space
X_c, Y_c, Z_c	Camera-centred coordinates of a point in space
x_i, y_i	Coordinates of a point on the image plane
u, v	Image coordinates in pixels
$\mathbf{R}$	Orthogonal 3×3 matrix representing a rotation
$\mathbf{t}$	Vector representing a 3-D translation
$\mathbf{A}$	Coefficients of plane affine transformation (2×2)

k_u, k_v, k_{uv}	Intrinsic camera parameters relating (x_i, y_i) to (u, v)
u_0, v_0	Intrinsic camera parameters denoting pixel coordinates of the origin
$\mathbf{P}$	3×4 matrix of projective camera coefficients
$\mathbf{C}$	World coordinates of the camera's optical centre
$'$	Marks symbols relating to second camera in stereo pair
$\mathbf{F}$	Fundamental matrix (3×3) encoding epipolar constraint
$\mathbf{o}$, $\mathbf{o}'$	Homogeneous coordinates of the epipole in each image

$\mathbf{M}$	2×3 matrix of affine camera coefficients
$\mathbf{Q}$	4×3 matrix of affine stereo coefficients
$\mathbf{c}$, $\mathbf{c}'$	Vectors parallel to camera viewing directions in space
$\mathbf{e}$	4-vector of coefficients for the linear epipolar constraint
x, y	Rectified image coordinates so that $y = y'$ for corresponding points

Chapter 3

$\mathbf{X}_P$	World coordinates of a point on the robot gripper
$\mathbf{\Theta}$	Joint angles specifying robot configuration
$\mathcal{K}$	Kinematic relation: $\mathbf{X}_P = \mathcal{K}(\mathbf{\Theta})$
$\hat{\ }$	Denotes a modelled or estimated quantity

$\mathbf{J}_{\mathcal{K}}$	Jacobian of $\mathcal{K}$, or matrix of partial derivatives of $\mathbf{X}_P$ wrt Θ
$\mathbf{u}_P$	Image coordinates of a point on the robot gripper
$\mathbf{u}_S$	Image coordinates of set point
$\mathbf{Q}$	4×3 matrix relating world to image coordinates
$\mathbf{Q}^+$	Left pseudo-inverse of $\mathbf{Q}$, such that $\mathbf{Q}^+\mathbf{Q} = \mathbf{I}$
$\mathbf{X}_P^*$	'Controller space' coordinates sent to kinematic model and robot
$\mathbf{o}_P$	Vector encoding gripper orientation in image-based terms
$\mathbf{o}_S$	Vector encoding target orientation in image-based terms
$\mathcal{F}$	Function relating image-based orientation to controller's parameterisation

Chapter 4

u, v	Image coordinates in pixels
X, Y	Canonical frame coordinates of points on the plane
$\mathbf{T}$, $\mathbf{T}'$	Transformations between plane and image coordinates
l_i	Line in the image defined by one view of pointing hand
l_{gp},l'_{gp}	Constraint lines on the plane, which intersect at the indicated point

Chapter 5

u, v	Image coordinates in pixels
x, y	Approximately rectified image coordinates
$y_{i[0]}$, $y_{i[1]}$	The y coordinates of each endpoint of the ith line segment
L_i	Length of segment in a 'vertically stretched' frame, used for matching
θ_{ij}	Angle between segments in the 'vertically stretched' frame
δ	Disparity along the epipoplar lines, $(x' - x)$
σ_{ij}	Intrinsic support for the match between segments i and j
$\mathcal{O}(n^a)$	Proportional to n^a as n becomes large
n	Number of features (straight line segments) in each image
r	Number of figurally related features per feature $(r < n)$
m	Number of candidate matches per feature $(m \leq n)$

Chapter 1

Introduction

This chapter sets out the motivation for the project, and introduces robot hand–eye coordination with a survey of existing robot vision systems. The contributions of the dissertation are summarised.

1.1 Motivation

When humans grasp and manipulate objects, they almost invariably do so with the aid of vision. Visual information is used to locate and identify things, and to decide how they should be grasped. Visual feedback helps us guide our hands around obstacles and align them accurately with their goal. *Hand–Eye Coordination* gives us a flexibility and dexterity of movement that no machine can yet match.

Robot manipulators have traditionally been restricted to performing repetitive tasks in highly ordered environments. Reliable and flexible computer vision would enable them to operate in less structured environments containing displaced or unfamiliar objects; to overcome operational errors using visual tracking and feedback; and to be programmed more easily via novel user interfaces such as gestures and pointing.

Because most robots need to move in all three dimensions, we exploit *stereo vision*, the use of two (or more) cameras to obtain 3-D information about the robot and its workspace. The scope of stereo vision applications is generally limited by the need to *calibrate* the vision system — the camera geometry must be measured to a high level of precision [129]. A well-calibrated stereo rig can accurately determine the position and shape of things to be grasped; however, if calibration is erroneous or the cameras are disturbed, the system will often fail gracelessly.

Here we explore the use of robust algorithms for stereo vision and hand–eye coordination which require minimal calibration and can tolerate some uncertainty in camera and robot positions and orientations. We develop a novel **visual grasping system** which uses vision to help plan and execute grasps of unmodelled objects placed at unknown positions in its workspace. The user indicates an object by a pointing gesture, and uncalibrated stereo vision is used to reconstruct its surfaces. Finally, the object is grasped by the robot under visual control.

Two strategies are employed to reduce dependence on calibration: firstly, by the use of *invariant* cues and representations of scene structure which are independent of camera geometry; secondly by the use of *image-based feedback* to correct for errors and align the robot with a visible target. Implementation is based on monochrome CCD cameras and a standard workstation environment.

1.2 Robot vision hardware

1.2.1 Configurations

A number of systems have been proposed using machine vision to help robot manipulators perform pick-and-place operations. The vision hardware may consist of one or more monochrome or colour video cameras, or more sophisticated devices such as structured light or laser rangefinders [16, 116, 75]. Figure 1.1 shows two common configurations:

Eye-in-hand systems have a camera mounted on the last link of the robot manipulator. This gives a detailed view of objects to be grasped, and facilitates visual servoing to align the gripper prior to grasping [21, 31, 27]. It also permits dynamic inspection of the target object from multiple viewpoints, affording a 3-D reconstruction of their surfaces [22, 125].

Eye-in-hand cameras typically suffer from a limited field of view and depth of field, so do not provide an overall picture of the workspace. They require camera calibration and *a priori* knowledge of the camera pose relative to the gripper, since these parameters cannot be recovered by self-calibration [48]. The entire visual field also moves whenever the robot does, which can increase image-processing overheads [134].

External camera or ‘independent-eye’ systems view both the manipulator and its workspace using one or more distant cameras. By observing the manipulator making known motions, self-calibration is possible [61], and feedback may be used to drive the gripper to a visually-specified target configuration regardless of camera position [134, 48].

With static external cameras, objects in the workspace tend to be viewed at a lower resolution than with a robot-mounted camera. The cameras may instead be mounted on pan/tilt heads, or an integrated stereo head [89]; zoom lenses may be employed for detailed inspection of parts as well as a broader view of the workspace [124]. Additional flexibility may be gained by mounting one or more cameras on an independent robot arm to allow dynamic control of viewpoints [19, 90], although this is clearly more expensive to implement.

1.2.2 Experimental setup

For this project, an external camera system is employed, with a pair of monochrome CCD cameras arranged for stereo vision (figure 1.2). The cameras view a robot manipulator and its workspace from a distance of about 2m; their field of view can also accommodate an operator, who can communicate with the system by means of pointing gestures. The angle between the cameras is in the range of 15–30 degrees. There is some flexibility in the positioning of the cameras, which are mounted on free-standing tripods: since these tend to be disturbed frequently, accurate calibration data are not available.

The experimental system is based around a Sun SPARCstation 20 with a Data Cell S2200 frame grabber. The manipulator is a Scorbot ER-7 robot arm, which has 5 degrees of freedom and a parallel-jawed gripper. The robot has its own 68000-based controller which implements the low-level dynamic control loop and provides a Cartesian kinematic model: the computer controls the robot and supplies it with visual feedback by means of a serial interface.

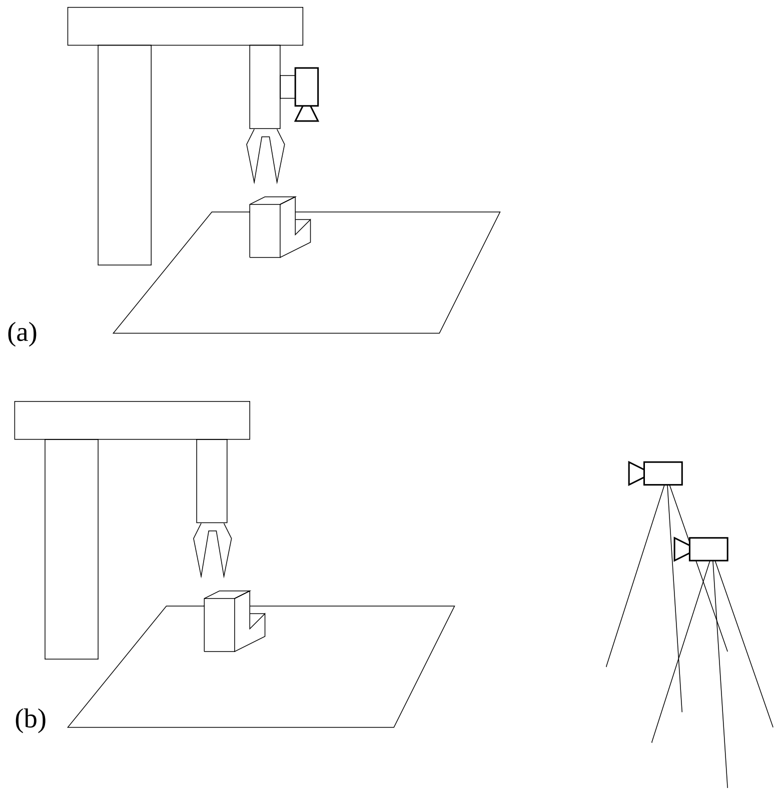


Figure 1.1: Robot/camera configurations: (a) eye-in-hand (b) external cameras

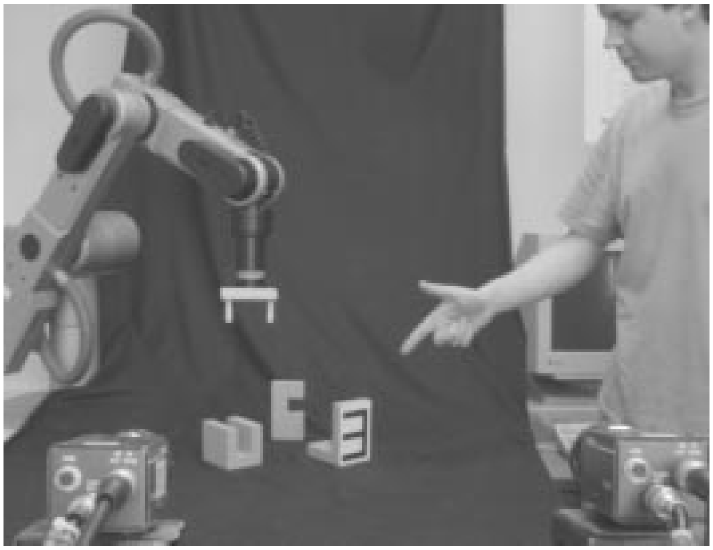


Figure 1.2: The experimental setup showing the robot, its workspace, stereo cameras and operator.

1.3 Existing systems

Here previous work in computer vision for robot manipulator guidance is reviewed (particular techniques and relevant theory will be surveyed in more detail in later chapters).

1.3.1 Look and move

The earliest paradigm for hand–eye coordination has become known as the *'look-and-move'* approach [48]. Vision is used only in the planning of motions, which are executed without visual assistance.

In two dimensions. Early robot vision systems [15, 97, 51], and many still in widespread use, have a single overhead camera to extract two-dimensional information about the positions of features on a part to be grasped, to recover its 2-D pose or to select different robot actions based on object recognition. This approach minimises the computation spent on vision, since the camera is used only once per operation, to analyse a static scene. Frequent recalibration is required to maintain satisfactory operation [15].

Binocular stereo vision. A recent hand–eye system [112] based on the Sheffield TINA stereo vision algorithms of Pollard et al. [103] uses a pair of calibrated cameras which view straight-edged objects taken from a modest repertoire. The system constructs a wire-frame model of the objects' edges [100] which is matched against stored models of the objects [99]. Objects are identified and picked up by an RTX robot using pre-determined grasps.

Other ranging techniques. An experimental system of Ikeuchi et al. [62] reconstructs the contents of a workspace using *photometric stereo*, in which a camera takes multiple images of the scene under different lighting conditions, to recover local surface orientation. This information is supplemented by range data provided by the PRISM stereo system [93], which projects random texture onto the scene and matches the resulting views by a multi-scale algorithm. These techniques allow it to reconstruct, recognise and grasp objects with smooth, featureless surfaces which would otherwise be difficult to see.

Robot planning systems have been proposed using other specialised sensors, such as *laser rangefinders* which reconstruct surface shapes from a single 'view'

[16, 75, 111]. Range imaging can recover the shapes of arbitrary surfaces more accurately than stereo [16]; however the sensors are expensive and require precise calibration.

Tracking and interception of moving targets. Vision can also be used to perform dynamic tasks involving moving objects. Allen et al. [1] describe a stereo vision system which can track a single target at frame rate, using a Kalman filter to estimate and predict its motion. The system is demonstrated using a robot arm to intercept and grasp a model train. Other high-speed stereo tracking systems have been used to perform tasks such as striking a ping-pong ball [2, 107]. As with all of the above systems, there is no visual feedback of the grasping operation, which is not robust to physical disturbances.

Open loop robot vision has become very sophisticated, and has been demonstrated successfully in pick-and-place and other applications. However, it requires accurate calibration so that the imaging and kinematic processes can be inverted without error, and demands high repeatability from the robot manipulator. The need for precision is most acute in the 3-D case, due to the increased number of parameters to be known and the added complexity of both robot and vision systems [119].

1.3.2 Visual feedback in two dimensions

In these systems, a single camera observes a manipulator from above, to guide the gripper's motion in two dimensions. The third dimension of movement is assumed to be constrained or controlled by an independent mechanism, as in many '2-and-a-half dimensional' robots which manipulate objects on a flat table, and whose vertical motion is limited and independent of the main X–Y motion.

Visual feedback for gripper alignment. The seminal work of Shirai and Inoue [117] reported the use of visual feedback to align a square prism over a box into which it is then fitted. The dimensions and heights of the objects were given, but the initial position of the box was unknown, and there was some uncertainty in the alignment of the prism within the gripper. Vision was used to estimate the two-dimensional position and orientation of the box, to place the prism over it. The system then observed the prism, estimated the error in its position and orientation, and made corrective motions.

A similar system was presented in [19] as a 'behavioural module' for an existing model-based manipulation system (the Edinburgh SOMASS system [77]). Here the manipulator moves across a horizontal *approach plane* to align itself vertically with a target, in preparation for grasping. Vision is used to track markers on the two fingers of the gripper, to provide visual feedback. It is noted that there is a 1-to-1 mapping between the approach and image planes, so that feedback can be based directly on the difference between observed and desired image positions.

Dynamic visual control. Most visual feedback systems use a hierarchy of two control loops: an inner one using joint sensors to control the robot's dynamics and an outer, slower loop incorporating vision. However, a few systems attempt to integrate the two using field-rate tracking of simple features on the manipulator [48]. In one experimental setup, a manipulator moves across a flat table and is viewed from above by a single camera [134]. The same camera is used to locate the target object during the planning phase. A point on the end-effector is marked by a beacon which allows it to be tracked at 50Hz to provide position-based feedback during execution. It is shown that the integration of visual feedback into the controller permits efficient operation and fast convergence despite significant errors in camera calibration or kinematic modelling.

Because of the simple 1-to-1 mapping between world and camera coordinates, visual feedback is an effective way to null positioning errors in two dimensions [136]. For fast, efficient operation, visual *tracking* of the end-effector (and/or its target) is required, to continuously update the estimate of the error between the manipulator's actual and desired pose.

1.3.3 Single camera feedback for 3-D tasks

These systems deal with the positioning of a robot in three dimensions under visual control, using either an eye-in-hand or external camera.

Hybrid system with 2-D vision. Harrel et al. [54] describe an eye-in-hand system to guide a fruit-picking robot. This system is notable by its use of *colour* vision to segment citrus fruits from the background and track them. The vision system provides two-dimensional feedback, controlling two degrees of freedom of the arm to keep the camera fixated on the fruit as it approaches; the distance is measured independently by ultrasonic ranging.

Single camera pose estimation. Espiau et al. [31] consider the use of visual feedback to place a calibrated camera in a given pose relative to visible features. They derive analytically the *image Jacobian* (that is, the matrix encoding the differential relation between camera motions and changes in image measurements) as a function of image feature positions. Inverting this relation allows the robot to make appropriate movements to bring the image features into a specified configuration, constraining the camera pose with respect to the target. This is demonstrated for the alignment of an 'eye-in-hand' camera with respect to a known target object.

Affine visual servoing. In the case where the target features are confined to a plane, the interaction between image and world motion is simplified. Colombo and Crowley [27] present a system which tracks features on a target surface and positions a camera at a given pose relative to the surface, deriving the gains for image-based control from a *weak perspective* [108] approximate pose estimation.

Spratling and Cipolla [121] present a similar system which requires no calibration but continuously re-estimates the image Jacobian from recent motions, to bring the camera into the pose corresponding to a goal image. They track the target surface using an active contour, and estimate the affine transformation between observed and goal configurations from area moments, making it correspondence-free [113].

The construction and attainment of an image-based goal requires a model of the camera and of the object to be manipulated [133, 31]; and pose estimation from a single view is ill-conditioned when the camera is distant [53]. Therefore, single-camera servoing is best suited to calibrated eye-in-hand systems.

1.3.4 Stereo visual feedback

Systems have also been proposed using *stereo* visual feedback to improve the accuracy of 3-D manipulation.

Image-specified manipulation. Skaar et al. [119, 18] consider the case in which known points on a manipulator are sporadically observed by two or more cameras, but continuous stereo tracking is not possible. They introduce a simple orthographic camera model and show that the estimated camera parameters also absorb linear errors in the kinematics: this allows the system to predict the configuration which will bring the gripper to a visually-specified target in two or more views. By appropriately weighting a set of observations, they are able to solve for the local hand–eye relation in any region of the workspace, allowing 6-DOF[1] alignment of a gripped object with a visually-specified target.

Stereo image-based feedback. Hollinghurst and Cipolla [61] demonstrated the use of stereo tracking of a robot manipulator whose kinematics are (approximately) known, using visual feedback to align it with a target. A linear camera model is assumed. An extension of this system is described in chapter 3.

Hager et al. [50] present a similar system, using stereo image-based feedback for 6-DOF positioning. Approximate camera calibration is used to estimate the image Jacobian, but the system is shown to be insensitive to calibration errors. Hager then considers the use of visual feedback to enforce one or more constraints (with 6 DOF or less) between the end-effector pose and that of another object, using least-squares solutions in both the underconstrained and overconstrained cases [49]. Visual constraints are used to assist dextrous tasks such as the insertion of a floppy disk into a drive.

Multiple cameras simplify the problems of setting and attaining visually-specified goals for 3-D positioning, and allows the manipulation of *unmodelled* objects (whose pose cannot be determined in a single view). Such systems are robust to small errors in the robot's kinematic model and allow precise manipulation tasks to be performed.

1.3.5 Learning systems for hand–eye coordination

Some systems deal with unknown robot kinematics as well as unknown camera parameters by considering the *visual kinematic relation* between actuator settings and parameters extracted from the image. Since robot kinematics are usually highly nonlinear, the structure of this relation must be *learnt* either before or during operation.

Mel's MURPHY. Mel [85, 86] took inspiration from human learning to devise a vision-guided control and planning system that *learns by doing*. It controls

[1] That is, control of both *position* and *orientation* in three dimensions.

a 3-DOF planar arm and guides it to a visible target whilst avoiding obstacles. MURPHY learns the forward visual kinematic relation, taking an unusual approach by learning to 'envisage' an entire 64×64 image of the arm in any configuration. It is this whole-image-based approach that is the key to its collision-avoiding behaviour. It also learns the inverse differential kinematic relation by observing how the gripper position responds to changes in actuator settings. Learning takes place in an initial 'random flailing' stage in which it views about 17000 of the 3 million legal arm configurations. These models are used by a path planner, which is based upon heuristic depth-first search. A trajectory is planned in joint space, to reach the target avoiding obstacles. Despite promising initial results and a refreshingly simple approach, MURPHY is slow, and scales badly to higher degrees of freedom. Its neural network architecture could not efficiently model the simple geometry underlying the camera and kinematic relation.

3-D visual kinematic learning. Hervé et al. [58, 59] take a qualitative approach to visual kinematic learning by identifying the *singularities* in the joint space / sensor space transformation (points where $|\mathbf{J}| = 0$, i.e. the inverse differential relation is not defined). Away from these singularities, the hand–eye relation is smooth and can be navigated using feedback. The robot makes experimental motions to determine the gradient of its *Perceptual Control Surface*. It builds up a qualitative model of the PCS by noting when it encounters a singularity in the Jacobian, and plans paths which avoid these singularities.

Visual memory-based control can be used to control manipulation by a multi-fingered hand, whose kinematics are difficult to model analytically, by tracking the position and orientation of a grasped object. The system of Jägersrand et al. [63] estimates the Jacobian of the *visual kinematics relation*; that is, the matrix of coefficients relating the movement of each joint to movements in the image of the grasped object [48], using exploratory movements to obtain its components in various directions. As it moves, it builds up a piecewise linear model of this relation, with uncertainty analysis used to ascertain the region of trust for each linear patch.

Learning-based control can be useful when controlling a redundant or multi-fingered manipulator (which would otherwise be difficult to model [63]), but in general this is unnecessary and inefficient. Qualitative modelling of the hand–eye relation can also be used in conjunction with visual feedback [59].

1.4 The approach

Traditional robot vision systems have attempted accurate reconstruction, using metric information to plan and execute motions in an open loop [112, 1, 107]; but these require calibration and are not robust to disturbances. Systems have also been proposed using image-based feedback with varying calibration requirements [117, 134, 31]. Sometimes the characteristics of the robot itself are learnt along with the parameters of the vision system [59, 133, 63].

Here we address the case in which the cameras are uncalibrated but the robot's kinematics are known (perhaps imperfectly), allowing the end-effector to be controlled in terms of Cartesian coordinates with a small, smooth error function. In the absence of accurate calibration, it is reasonable to resort to an approximate linear model of stereo vision. The Cartesian hand-eye relation is monotonic and can be modelled by a linear relation.

The use of a kinematic model simplifies the learning of the hand–eye relation (a linear estimator will suffice), whilst the use of visual feedback retains robustness against small kinematic errors and even non-stationary camera parameters. Such an approach has been used very successfully for visual robot control in two dimensions. Here it is applied to stereo vision for three-dimensional control of position and orientation. We track the robot's gripper in stereo with active contours, and use visual feedback to servo its image position in the two views.

To exploit visual feedback in grasping operations, the manipulator's goal configuration must be specified in terms of *image* measurements. Indication of the target object must therefore be image-based, and this can be achieved using a visual user interface. We use stereo vision to form an affine reconstruction of the facets of the target object in an image-based coordinate frame. This representation is used in conjunction with visual feedback to align the robot gripper with a suitable facet of the target object, so that it may be grasped.

Thus the entire grasping operation is to be facilitated by uncalibrated stereo vision.

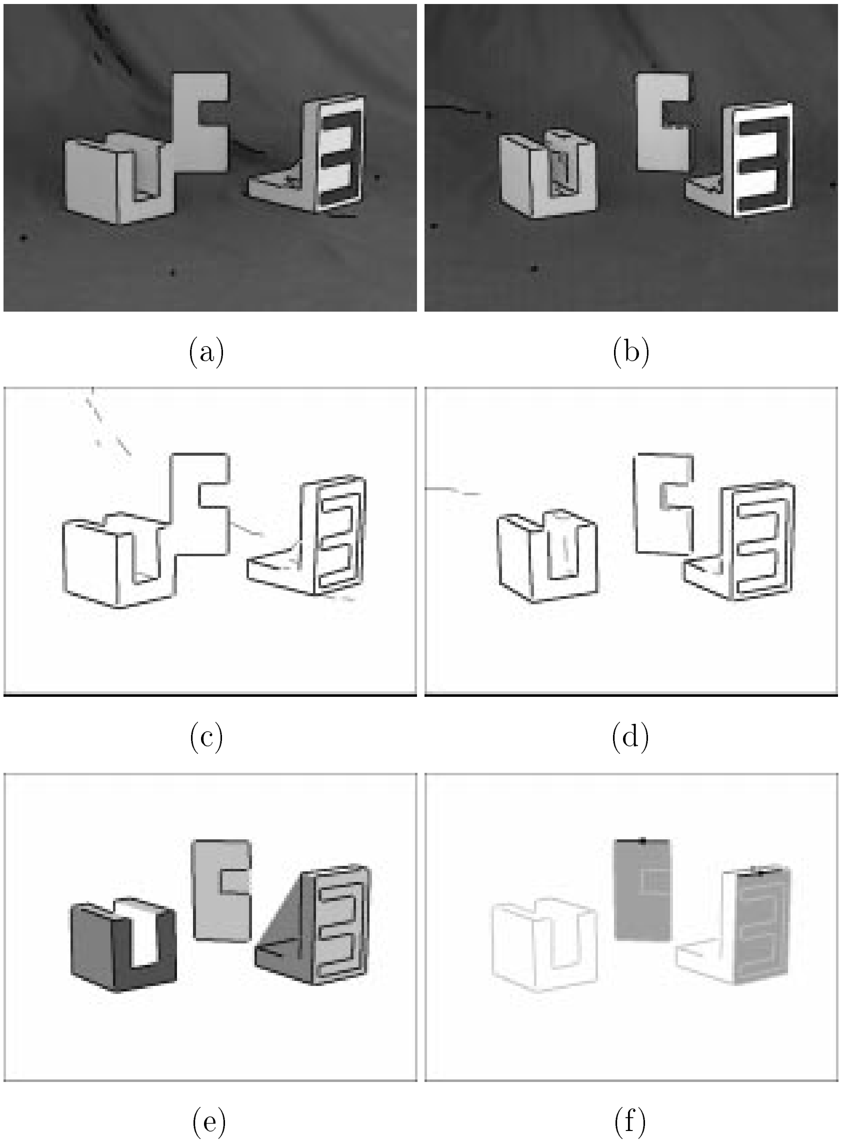


Figure 1.3: Surface reconstruction for grasping: (a,b) stereo images of the workspace with edges superimposed; (c,d) unmatched *(light)* and matched *(dark)* line segments; (e) cyclopean view with planar facets identified; (f) proposed grasping sites.

1.5 Contributions

1.5.1 Affine stereo

In this dissertation, it is argued that a linear approximate camera model is well suited to practical uncalibrated stereo, both for solving the correspondence problem and for modelling the relation between world and image motions, whenever the camera configuration resembles the typical ‘parallel’ arrangement with equidistant cameras fixating on a compact scene. It is noted that the epipolar geometry of a stereo rig is qualitatively different from that found in many navigation/structure-from-motion applications in which the camera motion is largely along the optical axis. The restricted form of the *affine camera* makes it easier to compute approximate camera parameters from a small number of measurements, than the projective camera model estimated in the traditional manner. Affine stereo is shown to be more robust to image coordinate noise and disturbances to the cameras.

1.5.2 Pointing interface

A novel form of human–robot interface is presented, based on real time stereo vision tracking of the operator’s pointing hand. We do not use a full 3-D reconstruction of the hand in space, but consider only plane projectivities between a ground plane and the images. This formulation allows objects on a plane to be indicated by pointing, without the need for camera calibration. Simulations and experiments measure the accuracy of the system, both in open loop and as a means for the operator to servo the position of the robot’s gripper.

1.5.3 Weakly calibrated stereo reconstruction

A new stereo matching algorithm is developed for matching line segment images in weakly calibrated stereo pairs (in which the epipolar geometry is only approximately known because only a few reference correspondences have been observed) under weak perspective. Integrated into the system is the grouping of line segments into planar facets. This provides a model of the scene which is suitable for grasp planning with a parallel gripper (figure 1.3). It also allows the visible surfaces to be reconstructed accurately despite uncertainty in the epipolar geometry, which is not generally possible for individual line segments. The reconstruction is used to select a suitable grasp of the target object.

1.5.4 Visual feedback for grasping

The linear approximation to the 'hand–eye relation' between the robot's movements and motion in the images is used as the basis of a visual feedback control loop, allowing the robot to be guided in three dimensions towards a visually specified target. The robot is aligned so that a surface of its gripper is near to and coplanar with a given surface of the object; then rotated into the grasping configuration. It is shown that such an approach is robust to calibration errors of either the robot or vision system, and even to disturbances to the system occurring during operation.

1.6 Overview of the dissertation

Chapter 2 gives a general introduction to the geometry and modelling of stereo vision systems, and derives the camera models which are referred to later in the dissertation.

- Conventional projective and affine models of video camera imaging are introduced. The theory of camera calibration and the epipolar geometry of stereo vision are reviewed for each model.
- The affine and projective camera models are compared in the context of parallel-camera stereo vision, and it is concluded that the affine camera is more robust to errors and more easily calibrated.
- This is supported by experiments and simulations comparing projective and linear models degraded by noisy data. It is shown that the systematic error due to the linear approximation is of comparable magnitude to other errors, e.g. noise in image in feature localisation.

Chapter 3 describes the use of the affine stereo formulation developed above to achieve alignment of the robot with a visually-specified target, using stereo visual feedback.

- A visual feedback scheme is developed for the affine stereo formulation. It is noted that, for point alignment, image-based and position-based servoing are equivalent under this model.
- We extend the feedback scheme to align both the *position* and *orientation* of planar features on the robot and target object.

- The system is implemented using affine active contours to track a surface of the robot's gripper. By tracking the target facet as well as the robot, we close the visual control loop and enable the system to track and grasp objects despite movements and disturbances to the cameras.
- Experiments show that this system is robust to camera motions and small errors in the robot's kinematic model.

Chapter 4 describes a novel human–robot interface based on pointing. This is the proposed means for indicating to the system which object is to be grasped.

- The geometry of pointing at a ground plane is analysed, and it is shown that this does not require camera calibration, apart from 4 matching reference points on the plane.
- It is shown how this method may be used to indicate points on a single plane or in an environment containing multiple planes.
- Methods for tracking a pointing hand are summarised, and a novel implementation is developed using a pair of affine active contours to track the thumb and index finger.
- Experimental results and accuracy evaluation are presented.

Chapter 5 discusses the stereo correspondence and reconstruction of a scene composed mainly of straight edges and planar surfaces.

- Previous approaches to solving the correspondence problem in stereo vision are reviewed, and their shortcomings discussed in the context of uncalibrated or weakly calibrated setups of the kind used in this project.
- A stereo matching algorithm is developed, based on existing work for line segment matching but explicitly allowing for a bounded error in epipolar constraint estimation.
- Uncalibrated plane grouping is incorporated into the system, exploiting the geometry of weak perspective views of coplanar features.
- The groupings are used to extract a description of the planar surfaces of objects for reconstruction, and to improve the accuracy of uncalibrated reconstruction.

Chapter 6 is concerned with the implementation of the complete visual grasping system.

- The theory of grasping is briefly reviewed, with particular emphasis on grasp synthesis for a parallel-jawed gripper.
- A scheme is devised for choosing grasping sites on a stereo reconstruction of the surfaces of the target object, and demonstrated on real images of 'blocks world' scenes.
- The algorithms described in the dissertation are integrated to form a complete system.

Chapter 7 reviews the findings and contributions of the dissertation, and concludes with an outline of future work.

Appendix A describes the novel type of active contours used in the project. These are based on a template and are able to deform only affinely. They are suitable for the real time tracking of planar objects or facets under weak perspective, as well as for tracking the index finger and thumb of a pointing hand.

Chapter 2

Perspective and Affine Stereo

In this chapter we review the geometry of monocular and stereo cameras, and show that an approximate linear model of stereo vision is robust and well-suited to uncalibrated and weakly calibrated systems.

2.1 Introduction

In order to make geometrical use of stereo vision we must model the relation between the three-dimensional world and two-dimensional images. Specifically, we will need to use stereo to reconstruct the shapes of objects in the robot's workspace in order to grasp them successfully, and to associate relative image positions with 3-D motions to drive the robot to its target configuration.

This chapter reviews the geometrical modelling of the perspective camera; its generalisation, when full calibration data are not available, to the projective camera; and a useful linear approximation, the affine camera. Essential theory for stereo vision is summarised in each case, describing the relation between a stereo pair of views, and the use of calibrated and uncalibrated stereo systems to reconstruct points and surfaces.

Numerical experiments will be used to demonstrate the superiority of the affine camera for the estimation of the epipolar constraint and the reconstruction of relative positions in three dimensions, when calibration data are noisy and few — it is this *weakly calibrated stereo* model that is used in chapter 3 to control a robot and in chapter 5 to facilitate stereo correspondence and the reconstruction of planar facets.

2.2 The perspective camera

2.2.1 Pinhole camera

Video cameras are conventionally analysed using the *pinhole camera model*, in which an image is projected onto a retinal plane by rays passing through a single point called the *optical centre* [32]. This point forms the origin of a camera-centred coordinate frame, (X_c, Y_c, Z_c) such that the retinal plane has the equation $Z_c = f$, where f is a constant, the *focal length*. Image coordinates (x_i, y_i) on the retina are ratios of world coordinates (X_c, Y_c, Z_c) thus: $x_i = fX_c/Z_c \ \ y_i = fY_c/Z_c$. This simple model is a good approximation to the optics of most types of camera, although it neglects effects such as lens distortion which are significant in some high-accuracy applications such as aerial photogrammetry [128].

The relation between the camera-centred and some other world frame (such as that defined by a robot or another camera) is a rigid motion, encoding the camera's orientation and position. It can be represented by an orthogonal rotation matrix $\mathbf{R}$, and a translation vector $\mathbf{t}$. Using homogeneous coordinates [8] with a tilde to symbolise equivalence up to a scale factor,

$$\begin{bmatrix} x_i \\ y_i \\ f \end{bmatrix} \sim \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \tag{2.1}$$

2.2.2 Projective camera

Measurements on the image plane are not made directly, because the image is sampled into *pixels*. The relation between retinal positions (x_i, y_i) and pixel addresses (u, v) is modelled by an affine transformation (to represent offsets, scaling and shearing) [32]. Aligning the pixel and retinal coordinate systems so that the v and y_i directions coincide,

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \sim \begin{bmatrix} fk_u & fk_{uv} & u_0 \\ 0 & fk_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \tag{2.2}$$

The 5 coefficients[1] fk_u, fk_v, fk_{uv}, u_0 and v_0 are the camera's *intrinsic parameters*, and the $\mathbf{R}$ and $\mathbf{t}$ components can be expressed in terms of 6 *extrinsic parameters*. Combining these relations, we obtain the *direct linear transformation* (DLT) form of the camera model [128]:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \sim \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \tag{2.3}$$

This is the usual camera model for many vision systems where the camera intrinsics and pose are not initially known [32]. The transformation matrix is defined up to a scale factor, thus there are 11 degrees of freedom.

2.2.3 Camera calibration

Calibration of the camera is necessary to fix the 11 unknowns in the 12 parameters p_{ij}. This can be done by observing at least 6 points of known position, not all coplanar. Each observation generates two homogeneous equations in terms of p_{ij}. The system is homogeneous, so we can constrain $p_{34} = 1$ and solve using linear least squares estimation. If image positions are noisy, the results can be improved by observing more than 6 points using a recursive linear estimator. Often a special *calibration object* with very accurate grids is used [6].

In practice, the linear method is somewhat ill-conditioned, and a large number of reference points are needed, which must be localised to sub-pixel accuracy [129]. This is because the error measure, when formulated linearly in p_{ij}, is not geometrically meaningful; the last row and column have different numerical dimensions and play different roles in the model. A number of calibration methods have been proposed based on nonlinear (iterative) optimisation and reparameterisations of $\mathbf{P}$, and these give somewhat better results [37, 130].

Having obtained the DLT form, the intrinsic and extrinsic parameters can be extracted if required. For any 3×4 matrix of rank 3, scaled so that $||p_{31}\ p_{32}\ p_{33}|| = 1$, it can be shown [32] that there exist four sets of camera parameters satisfying equation (2.2), the four solutions being trivially related by changes of sign. The optical centre can be recovered directly, by solving $\mathbf{PC} = \mathbf{0}$.

[1] Often, k_{uv} is taken to be zero [32]. This assumes rectangular pixels in the camera and complete decoupling of horizontal and vertical coordinates in the frame capture hardware. Thus there will be only 4 intrinsic parameters, and an additional constraint will be imposed on camera calibration.

Once the intrinsic parameters of the camera are known, pixel coordinates can be converted back to *normalised* image coordinates $(x_i/f, y_i/f, 1)$: these would be the image-plane coordinates for a pinhole camera of unit focal length, and are equivalent up to a scale factor to the camera-centred world coordinates. Hence directions and angles may be measured at the optical centre.

Camera calibration must be repeated whenever the camera lens is replaced, zoomed or refocused (change of f), or the camera position is disturbed (change of $\mathbf{R}$, $\mathbf{t}$). This project was motivated by a desire to avoid full camera calibration, and explores the use of formulations that work satisfactorily with few or no calibration measurements.

2.2.4 Viewing a plane

Consider the case in which several observed points line on a single plane. Thus in some world coordinate system they will all have $Z = 0$, and equation 2.2 loses one column of the camera transformation to become:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \sim \begin{bmatrix} fk_u & fk_{uv} & u_0 \\ 0 & fk_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{32} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} . \tag{2.4}$$

We see that the relation between plane and camera coordinates is a 2-D projectivity, preserving projective invariants of features on the plane [88]. If the intrinsic parameters are known, $\mathbf{R}$ and $\mathbf{t}$ (the camera pose relative to the plane) can be computed up to a two-fold ambiguity from just 3 known points, by exploiting the nonlinear constraints among elements of the rotation matrix [53]. If camera parameters are not known, a minimum of 4 reference points are needed to fix the 2-D projective relation between the world plane and the image coordinate system [109], allowing points and structures on the plane to be reconstructed from a single view.

2.3 Full perspective stereo

In general, a single camera gives only two-dimensional information about scene structure. In the absence of other constraints, two or more views are required for reconstruction.

2.3.1 The epipolar constraint

The image coordinates of a world feature in two images are not independent, but are related by an *epipolar constraint*. This comes about from the fact that 4 image coordinates are derived from only 3 degrees of freedom in world positions. Consider a family of planes passing through the optical centres of both cameras. These project to a family of *epipolar lines* in each image (figure 2.1). If a feature lies upon a particular line in the left image, the corresponding feature must lie upon the line in the right image, which is the projection of the same plane. Most stereo systems exploit this constraint, which reduces the search for matching features to a single dimension [98].

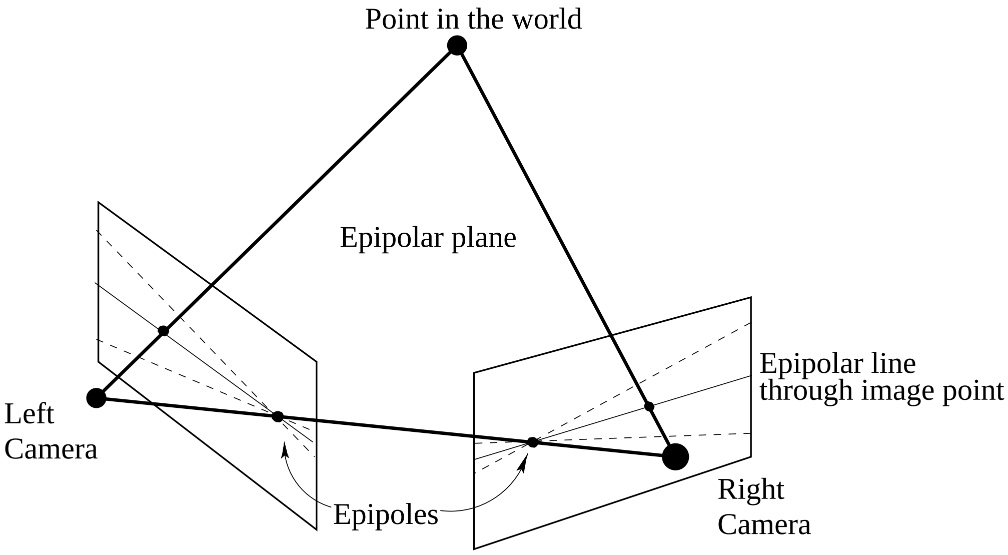


Figure 2.1: The epipolar geometry of stereo vision

Fundamental matrix

The epipolar constraint is represented algebraically by a 3×3 matrix $\mathbf{F}$ called the *fundamental matrix* [32] such that corresponding points (u, v) and (u', v') satisfy:

$$\begin{bmatrix} u' & v' & 1 \end{bmatrix} \mathbf{F} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = 0. \tag{2.5}$$

This is a generalisation of Longuet-Higgins' *essential matrix* [72], which encoded the relation between camera-centred coordinates in two views, to the case where intrinsic parameters are not known. $\mathbf{F}$ has rank 2 and is defined up to a scale factor, i.e. the constraint has 7 degrees of freedom.[2] The epipole in each view is the image of the other camera's optical centre (i.e. $\mathbf{o} \sim \mathbf{PC}'$, using homogeneous coordinates for $\mathbf{o}$ and $\mathbf{C}'$). The epipoles can be extracted from the fundamental matrix itself: $\mathbf{Fo} = \mathbf{0}$ and $\mathbf{F}^T\mathbf{o}' = \mathbf{0}$; that is, they are in the right and left nullspaces of $\mathbf{F}$.

In calibrated systems, $\mathbf{F}$ can be recovered from the camera matrices [32], otherwise it may be obtained up to a threefold ambiguity by observing 7 corresponding points [127], or estimated by linear least squares given 8 corresponding points [72]. Epipolar geometry can be estimated from image coordinates alone without reference to world coordinates; however, degeneracy occurs when the points all lie on a *critical surface* such as a plane, cone or cylinder [73, 36]. As with camera calibration, the solution is sensitive to errors and may require more than 8 points and/or nonlinear optimisation [76].

Linear form

In the general case, epipolar lines will meet at a single point in each image plane, the *epipole*, which is the image of the other camera's optical centre [32]. However, if the cameras' *focal planes* ($Z_c = 0$, $Z'_c = 0$) coincide, the epipoles will be points at infinity and the epipolar lines parallel. In this case, the first 2×2 elements of $\mathbf{F}$ become zero, and the epipolar constraint is a single linear equation in u, v, u', v' and a constant term [115].

In practical stereo rigs each camera is usually far outside the other camera's field of view, and the linear form of the epipolar constraint is often valid (at least as a first approximation when only a small number of correspondences have been found).

[2] The loss of rank can be explained by considering $\mathbf{F}$ as a projective *correlation* between points in one image and lines in the other. To be an epipolar constraint, all points on an epipolar line must yield the same line when multiplied by $\mathbf{F}$: the matrix is therefore singular.

The linear approximation to the epipolar constraint can be recovered from just 4 corresponding points in uncalibrated stereo. This form of the constraint is incorporated into the *affine stereo* model introduced in section 2.5.

Image rectification

The simplest possible form of the epipolar constraint occurs when the cameras have the same intrinsic parameters and are separated by a pure translation in the X_c direction (*parallel cameras*). The constraint becomes: $v' = v$, i.e. corresponding points must lie on the same horizontal scan line in each image, and object depth is encoded by horizontal *disparities* along the scan lines. This simplifies the problems of stereo correspondence and reconstruction, and many stereo vision algorithms require images in this form [103, 6]. The purpose of *image rectification* is to transform images (or the image coordinates of features) so that the epipolar constraint takes this form, even when they were taken through non-parallel cameras.

If the cameras are calibrated, rectification is achieved by projective transformations of image points into a new coordinate system (x, y) so that $y' = y$ for all matching points. The rectification transformations simulate the rotation of each camera until they are parallel, and the scaling and shifting of one image to bring the scan lines into agreement [32]. If the epipolar geometry is known but not the camera intrinsics, the rectification transformations are defined up to 9 free parameters,[3] usually chosen for numerical convenience [6].

2.3.2 Reconstruction

With calibration

Assume that both cameras have been calibrated for the same world coordinate frame. By rearrangement of (2.3), each measurement of (u, v) yields two simultaneous linear equations in (X, Y, Z), which represent the line of sight from a camera to a point in the world. Two views of the same point give us four linear equations which can be solved, e.g. by a least squares method. Numerical optimization can be used to improve robustness to noise (at the expense of speed), by minimising the offsets in image coordinates between observed and backprojected features [32].

[3] The epipolar constraint has 7 DOF, but a general pair of projective transformations on the two images would have $8 + 8 = 16$ DOF.

Intrinsically calibrated cameras

For cameras with calibrated optics but unknown pose, the rotation and direction of translation between the views may be estimated. The *essential matrix* (which is the fundamental matrix defined in terms of normalized image coordinates [72]) is computed from the intrinsic parameters and the image coordinates of at least 7 correspondences. It can then be decomposed [32] into the product of an antisymmetric matrix **T** and an orthogonal rotation matrix **R**. **T** encodes the translation and is defined up to a scale factor; thus the scene may be reconstructed up to a similarity.

Uncalibrated cameras

The extraction of non-metric and viewpoint-invariant information from completely uncalibrated cameras is a rapidly developing field in machine vision [88, 3, 10].

For instance, given two uncalibrated views of 8 corresponding points (from which the fundamental matrix can be recovered), it is possible to reconstruct the scene up to a 3-D projective transformation[4] [35, 33]. 5 of the points are used as a projective basis in space, i.e. they are assigned the coordinates $(1, 0, 0, 0)$, $(0, 1, 0, 0)$, $(0, 0, 1, 0)$, $(0, 0, 0, 1)$ and $(1, 1, 1, 1)$. Likewise, 4 of them form a projective basis in each of the images. Using these coordinate systems, each camera transformation matrix takes the form:

$$\begin{bmatrix} \mu a - \nu & 0 & 0 & \nu \\ 0 & \mu b - \nu & 0 & \nu \\ 0 & 0 & \mu c - \nu & \nu \end{bmatrix}$$

where (a, b, c) are the projective image coordinates of the fifth point. The coordinates of the optical centre can also be expressed in terms of a b, c, μ and ν. Thus, each camera model is fixed up to one degree of freedom, the ratio $\mu : \nu$. Faugeras shows how this may be eliminated using the epipolar constraint between views [35], exploiting the relation between the epipoles and the optical centres of the cameras.

This result can be extended by noting that we are within 3 degrees of freedom of an *affine* reconstruction of the scene. The 3 missing parameters encode a representation of the plane at infinity within the above projective basis, and these may be recovered by observing e.g. 3 vanishing points of parallel lines [105, 33].

[4] A projective representation of a 3-D scene is 9 DOF from Euclidean structure, allowing quite serious distortions of the reconstructed scene. It is therefore most useful for applications such as *recognition* of objects based on projective invariants [109].

2.4 Weak perspective and the affine camera

The projective camera model has many parameters and is nonlinear in form, making it difficult to calibrate. We now consider a simpler first-order approximation, the affine camera, as an alternative camera model for stereo vision.

2.4.1 Weak perspective

Let us assume that, within some region of the scene, the relative depth $|\Delta Z_c / Z_c|$ is bounded by a small value (*weak perspective* [108]). Equation (2.1) becomes:

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \frac{f}{h} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \tag{2.6}$$

where $h = \mathbf{r}_3 {\cdot} \mathbf{p} + t_3$, the normal distance between the focal plane ($Z_c = 0$) and a point $\mathbf{p}$ in the region of interest. We assume that h is constant across this region, i.e. that the relation between world and image coordinates is linear. With a camera whose intrinsic parameters are known, the X_c and Y_c components of feature positions can be recovered up to scale from a single view; and the camera pose can be estimated from ≥ 3 points in known configuration [53]. This approximation to the camera model is useful in *tracking* applications, where a compact object is observed moving around a three-dimensional space [56]. Under weak perspective, any image of a planar facet will be an *affine transformation* of the plane, encoding its depth and orientation relative to the camera, and images of planes will deform affinely under motion [67, 21].

2.4.2 Affine camera

Now if the depth of the entire scene is small compared to the camera distance, h can be assumed constant. Consider the images in terms of pixel coordinates (u, v). Without a knowledge of the intrinsic parameters, camera pose cannot be determined, but the camera model is further simplified:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} + \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}, \tag{2.7}$$

where (u_0, v_0) is the image of the world origin. This is equivalent to parallel projection followed by an arbitrary affine transformation in the image. It is known as the *affine camera* model [88].

The affine camera can be calibrated by observing just 4 reference points. All 8 coefficients are independent. Its linear form makes it less sensitive to calibration noise, since it can be optimised to minimise errors in the image coordinates themselves. Where the assumption of weak perspective throughout the scene can be made, it allows a more accurate camera model to be constructed from limited calibration data [23].

2.5 Affine stereo

With the affine camera model, image coordinates are linear functions (plus a constant offset) of the 3-D coordinates of points in the world. This simplifies the epipolar constraint, as well as calibrated and uncalibrated stereo reconstruction.

2.5.1 The affine stereo formulation

Combining information from a pair of images, we have four image coordinates (u, v), (u', v') for each point, all linear functions of the three world coordinates (X, Y, Z):

$$\begin{bmatrix} u \\ v \\ u' \\ v' \end{bmatrix} = \begin{bmatrix} u_0 \\ v_0 \\ u'_0 \\ v'_0 \end{bmatrix} + \mathbf{Q} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} . \tag{2.8}$$

$\mathbf{Q}$ is a 4×3 matrix formed from the m_{ij} coefficients of (2.7) for the two cameras. It should be noted that the integration of information from more than two cameras is easily accommodated within this framework: each additional view generates two extra linear equations which can be represented by extra columns to $\mathbf{Q}$.

2.5.2 The epipolar constraint in affine stereo

When a point is viewed in stereo, there are 4 image coordinates, all linear functions of 3 world coordinates. These cannot be independent, but are related by a single linear constraint: the epipolar constraint thus takes the linear form [115], and can be estimated from a minimum of 4 corresponding points.

To analyse the constraint, consider the 4-vector $\mathbf{e}$ satisfying $\mathbf{Q}^T\mathbf{e} = \mathbf{0}$, i.e. the direction orthogonal to the three *rows* of $\mathbf{Q}$. This is the annihilator for vectors of the form $\mathbf{Q}[XYZ]^T$. Thus the epipolar constraint may be written:

$$\mathbf{e} \cdot \begin{bmatrix} u - u_0 \\ v - v_0 \\ u' - u'_0 \\ v' - v'_0 \end{bmatrix} = 0. \tag{2.9}$$

Geometrically, the epipolar planes are the family of planes parallel to both viewing directions $\mathbf{c}$ and $\mathbf{c}'$ (the nullspace vectors of $\mathbf{M}$ and $\mathbf{M}'$), so that the epipolar line direction in the first image is parallel to $\mathbf{Mc}'$, and in the second image to $\mathbf{M}'\mathbf{c}$. Furthermore, it follows that $[e_1\, e_2]{\cdot}\mathbf{Mc}' = 0$ and $[e_3\, e_4]{\cdot}\mathbf{M}'\mathbf{c} = 0,$ since motion along the epipolar lines does not violate the constraint.

Image rectification

To rectify a pair of images, each point must be represented in terms of linearly independent coordinates (x, y) such that $y = y'$ for all matching points. This condition is satisfied when:

$$\begin{aligned} y &= -Ae_1(u - u_0) - Ae_2(v - v_0) + B, \\ y' &= Ae_3(u' - u'_0) + Ae_4(v' - v'_0) + B \end{aligned} \tag{2.10}$$

for some scale factor A and offset B. We can then use y and y' values to find or test for matching features in stereo. The rectified x coordinate is most conveniently defined as the component parallel to the epipolar lines in each image, so that basis vectors $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$ are orthogonal: thus rectification may be achieved using plane similarity transformations (rotation, translation and scaling) in each image.

2.5.3 Reconstruction

Calibrated cameras

If all the coefficients are known, world coordinates can be obtained by inverting (2.8). Since the model is linear in both the world and image coordinates, least-squares minimisation gives an optimal solution from (uncorrelated) noisy image data. Errors in calibration will manifest themselves as an affine distortion of the perceived coordinate frame [68].

In hand–eye applications, it might instead be convenient to calibrate the vision system in the coordinate space in which the manipulator is controlled (assuming this maps approximately linearly to Cartesian coordinates). This can be done by tracking the position of a robot gripper as it visits four predefined reference points [61].

Uncalibrated cameras

In the absence of camera calibration, any four (non-coplanar) points may be given arbitrary world coordinates (such as the canonical affine basis $(0, 0, 0)$, $(0, 0, 1)$, $(0, 1, 0)$ and $(1, 0, 0)$). The appropriate solution for $\mathbf{Q}$ yields an *affine* reconstruction of the scene, which preserves affine shape properties such as collinearity, coplanarity and ratios of parallel lengths. This is in accordance with Koenderink and van Doorn's *Affine Structure-from-Motion Theorem* [68].

2.5.4 Recovery of surface orientation from two views

Any two views of the same planar surface will be affine-equivalent: there will exist an affine transformation that maps one image to the other. This transformation can be used to recover surface orientation [21]. Let the linear mapping between the views be represented by transformation matrix $\mathbf{A}$ and a 2-D translation vector.

It is the $\mathbf{A}$ component which encodes orientation. Consider the standard basis vectors $\hat{\mathbf{u}}$ and $\hat{\mathbf{v}}$ in one image and suppose they were the projections of some vectors tangent to the surface. The columns of $\mathbf{A}$ itself will be the corresponding vectors in the second image. By inspection, the epipolar constraint requires that:

$$\begin{aligned} e_1 + e_3 a_{11} + e_4 a_{21} &= 0, \\ e_2 + e_3 a_{12} + e_4 a_{22} &= 0. \end{aligned} \tag{2.11}$$

Two degrees of freedom remain. For purposes of visual servoing on surface orientation, such transformations can simply be parameterised by the pair (a_{11}, a_{12}). For reconstruction, we can form a surface normal vector $\mathbf{n}$ from the cross product of two world-space vectors on the plane:

$$\mathbf{n} = \mathbf{Q}^{+} \begin{bmatrix} 1 \\ 0 \\ a_{11} \\ a_{21} \end{bmatrix} \wedge \mathbf{Q}^{+} \begin{bmatrix} 0 \\ 1 \\ a_{12} \\ a_{22} \end{bmatrix} \tag{2.12}$$

where $\mathbf{Q}^{+}$ is the pseudo-inverse $(\mathbf{Q}^T\mathbf{Q})^{-1}\mathbf{Q}^T$ [122].

2.6 Comparison of perspective and affine stereo

A series of experiments and simulations were performed to compare the accuracy of perspective and affine stereo models in cases where only a small number of calibration measurements were available, or the camera positions were perturbed after calibration. Two tasks were considered:

- recovery of the *epipolar constraint*, to facilitate stereo correspondence of two images of an unknown object;
- estimation of the *relative positions* of points, to facilitate reconstruction of an object and visual servoing to align a manipulator with it.

For the numerical simulations, two ideal pinhole cameras were simulated, facing the origin from a distance of 3–24 units, displaced by a rotation of 20° about a vertical axis (figure 2.2). They observed reference and test points within a unit cube centred about the origin. The focal length of the cameras varied with distance, so as to keep a constant image size (for a vertical unit vector at the origin) of 320 pel. Figure 2.3 shows the appearance of the unit cube for camera distances of 3, 8 and 24 units.

2.6.1 Epipolar constraint recovery

These experiments measure the accuracy with which the epipolar constraint may be estimated from a small number of reference points, in both the linear and fundamental matrix forms.

I. Accuracy of the linear model in noiseless simulations

With noiseless images, the fundamental matrix could be calculated with complete accuracy from 8 corresponding points. The linear approximate epipolar constraint was estimated, from 4 correspondences within the cube.[5]

For any point in the left view, an epipolar line may be predicted in the right. The normal distance between this line and the corresponding point gives us a measure of the error in the epipolar geometry model. Figure 2.4 shows the maximum and RMS error for a grid of points filling the unit cube. It can be seen that the errors due to the linear model decrease with increasing camera distance.

[5] Coordinates (-.3,-.3,-.3), (-.3,.3,.3), (.3,-.3,.3), (.3,.3,-.3): a regular tetrahedron.

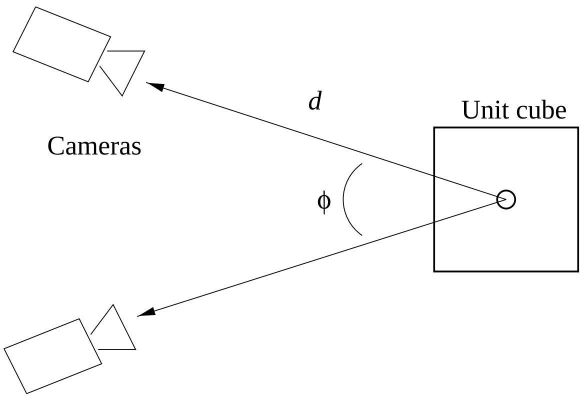


Figure 2.2: The camera geometry used in the numerical simulations. $\phi = 20^{\circ}$ and d varies from 3 to 24 units.

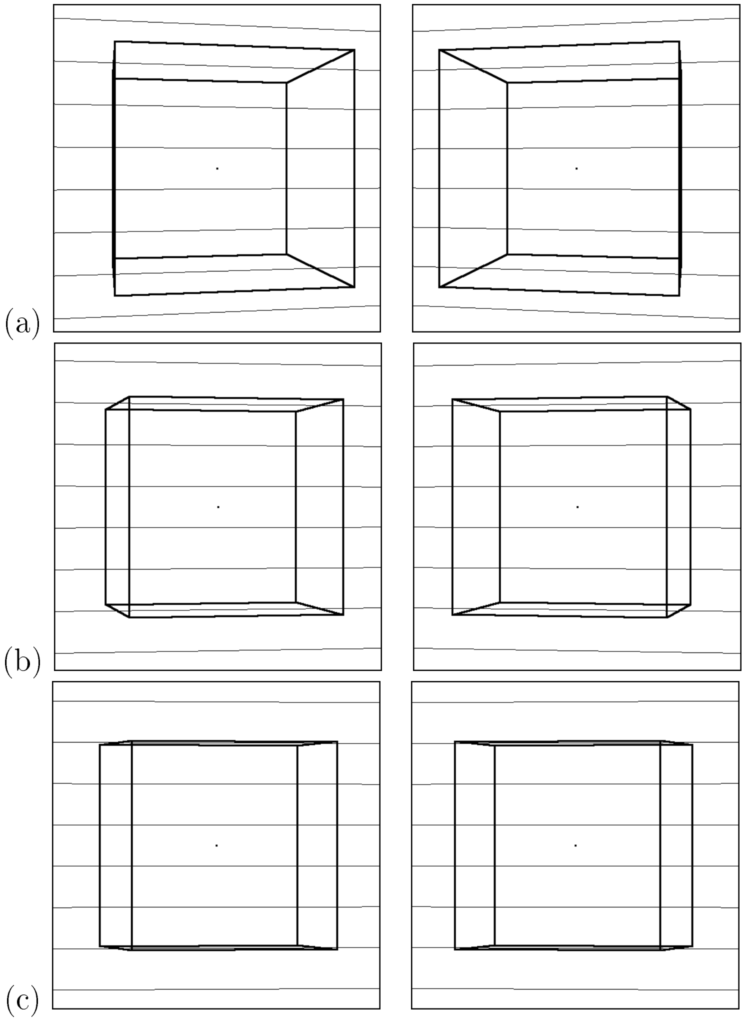


Figure 2.3: The appearance of the unit cube and epipolar lines viewed with the simulated cameras from a distance of (a) 3 (b) 8 (c) 24 units. Image size 512 pel.

II. Linear and fundamental-matrix models from noisy images

We now consider the case in which the epipolar geometry is estimated from noisy correspondences. Image coordinates of the reference points had Gaussian noise ($\sigma = 2.0$ pel) added to each axis. Linear epipolar constraints were estimated from 4 and 8 points, and a fundamental matrix from 8 points,[6] which is the minimum number for an unambiguous solution. Their accuracy was measured as above, using a grid of noiseless correspondences. Figure 2.5 shows the RMS error over 512 trials, for camera distances ranging from 3 to 24. By constraining the epipolar geometry to the linear form, greater robustness to noise is achieved.

III. From noisy image points and known world coordinates

If the world coordinates of the reference points are known, epipolar geometry may be estimated more accurately by first solving for a pair of camera models (calibration).

Reference point image coordinates had 2.0 pel noise as before, but accurate world coordinates were also available. This allowed affine and projective camera models to be estimated from 4 and 6 points respectively. The models were then rearranged to recover epipolar constraints, which take the fundamental-matrix and linear forms respectively. Figure 2.6 shows the RMS error over 512 trials, for camera distances ranging from 3 to 24. The use of world coordinates improves the estimate of the fundamental matrix, but makes no difference to the linear form in the 4-point case.

IV. Real data

For this experiment we used images of a robot to define 8 corresponding points, whose world coordinates were also known. Affine and projective camera models were estimated using linear least squares. A real scene was them observed in stereo, and a number of points of interest selected by hand in the left image. Figure 2.7 compares the epipolar line structure predicted by both affine and full perspective stereo models for matching these points. In this setup, in which the camera distances are about 2 metres, both models gave comparable accuracy — the RMS perpendicular error of the points in the right image from their predicted epipolar lines was 3.6 pel in each case. Furthermore, the affine model can predict epipolar lines using just 4 reference points with sufficient accuracy to allow matching (RMS error 4.4 pel); perspective stereo requires a minimum of 6 points.

[6] (-.3,-.3,-.3), (-.3,.3,.3), (.3,-.3,.3), (.3,.3,-.3), (.3,-.3,-.3), (.3,.3,.3), (-.3,-.3,.5), (.1, -.4, -.2).

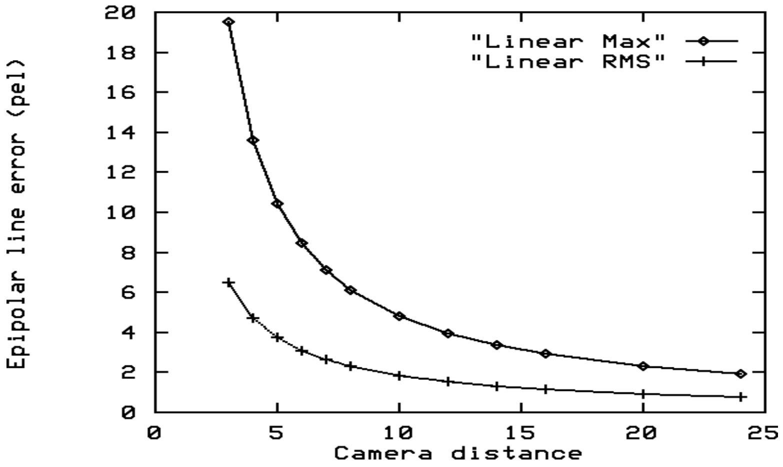


Figure 2.4: Worst-case and RMS error for the linear epipolar constraint.

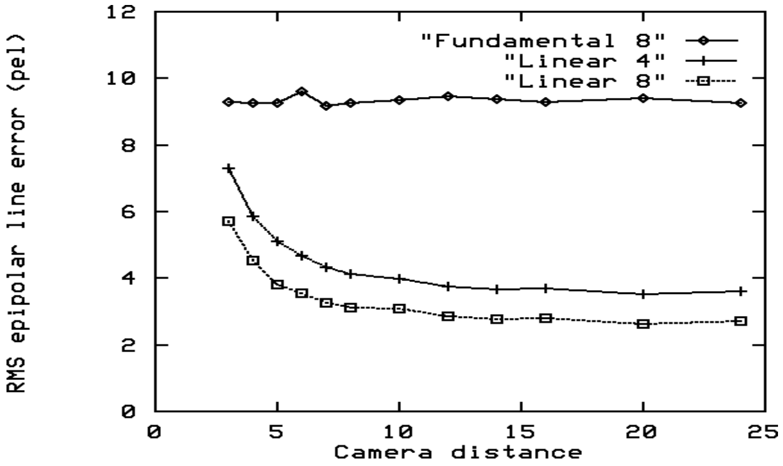


Figure 2.5: RMS error for linear and fundamental-matrix constraints, estimated from noisy correspondences ($\sigma = 2.0$ pel) by least squares.

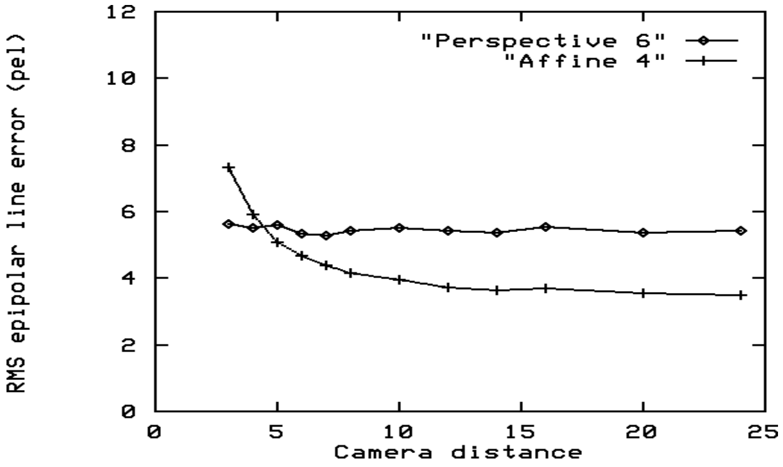


Figure 2.6: RMS error for linear and fundamental-matrix constraints, after noisy calibration of affine and perspective camera models.

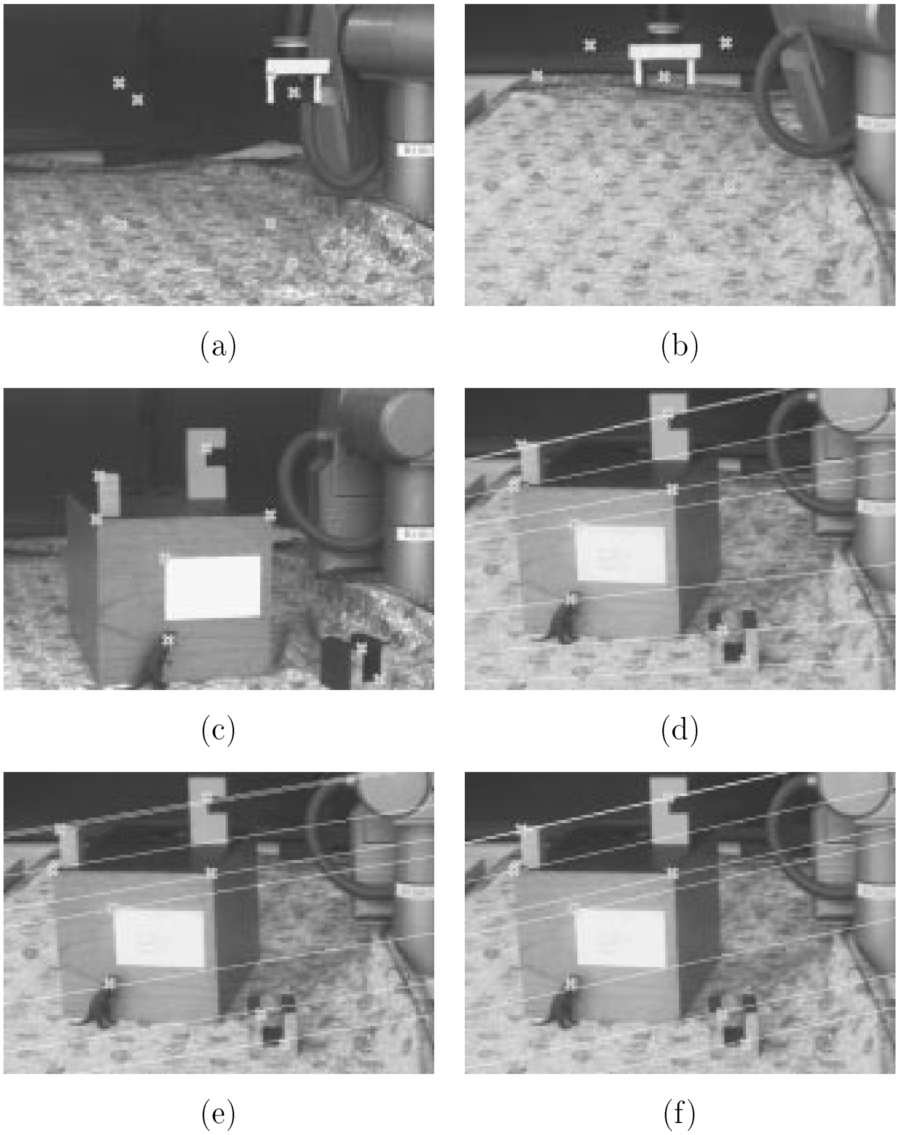


Figure 2.7: Estimation of epipolar lines: (a,b) two views of 8 reference points defined by the robot; (c) selected points in the left image; (d) epipolar lines estimated by the projective camera model after calibration with 8 points; (e,f) epipolar lines estimated by affine camera model with 8 points and 4 points respectively.

2.6.2 Accuracy of reconstruction

To compare affine and full perspective stereo reconstruction, simulations were performed measuring their ability to estimate the *relative* positions of points within the unit cube.

I. Under ideal conditions

Without noise or other disturbances, perspective stereo estimates absolute and relative positions with complete accuracy (in our ‘pinhole camera’ simulations, at least). An affine stereo model was calibrated using 6 reference points. At close range it performs poorly due to strong perspective distortion, but the error decreases in inverse proportion to camera distance. Figure 2.8 shows the RMS error for estimating the vector between a random pair of points within the unit cube (the average length of such a vector is 0.707).

II. With noisy calibration

Adding 2.0 pel noise to the image coordinates of the reference points causes both stereo models to lose accuracy (figure 2.9). Perspective stereo is more sensitive to noise because of its nonlinearity and greater degrees of freedom, and is *less* accurate than the affine stereo approximation at larger camera distances (viewing an increased number of reference points reduces the effects of noise and restores the accuracy of perspective stereo).

III. With noisy image coordinates after calibration

When Gaussian noise is added to the image coordinates of the points whose relative position is to be estimated (after accurate calibration), the effect is comparable on both systems. The two models converge for camera distances above ≈ 10 units (figure 2.10).

IV. Camera disturbances after calibration

In a laboratory or industrial environment it is possible for cameras to be disturbed from time to time and subject to small rotations and translations. If this happens after calibration, it will give rise to a corresponding error in stereo reconstruction.

Table 2.1 shows the average change in perceived relative position when one camera is rotated or translated a small distance around/along each principle axis.

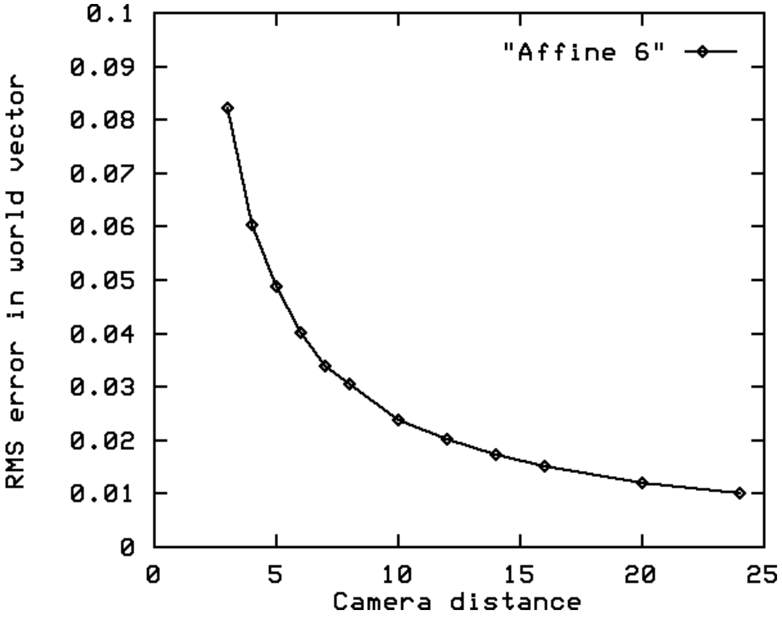


Figure 2.8: RMS relative positioning error (for random point pairs in the unit cube) as a function of camera distance, for the affine stereo model.

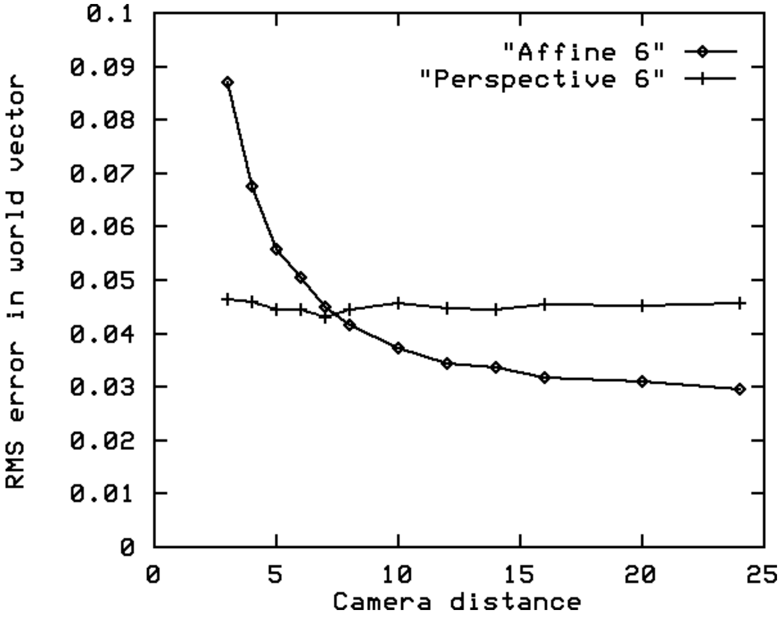


Figure 2.9: RMS relative positioning error as a function of camera distance, after calibration with 6 noisy reference points ($\sigma = 2.0$ pel).

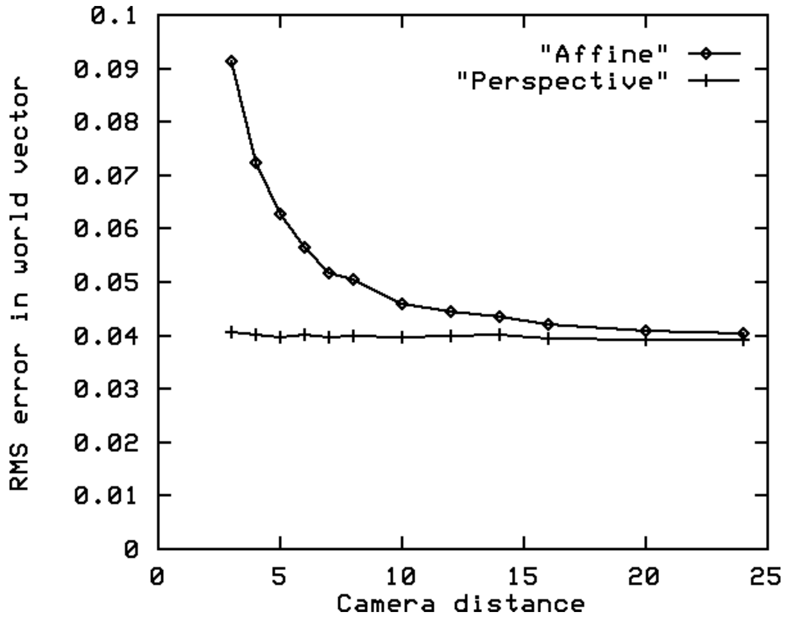


Figure 2.10: RMS relative positioning error from noisy images ($\sigma = 2.0$ pel) of world points after accurate calibration with 8 points.

Disturbance	Change (Affine)	Change (Perspective)
$X_c : Y_c$ (roll) rotation 1°	.0214	.0214
$X_c : Z_c$ (pan) rotation 1°	.0007	.0468
$Y_c : Z_c$ (tilt) rotation 1°	.0006	.0049
$X_c : Y_c$ (roll) rotation 5°	.1069	.1068
$X_c : Z_c$ (pan) rotation 5°	.0095	.1867
$Y_c : Z_c$ (tilt) rotation 5°	.0056	.0769
X_c (epipolar) translation 0.1	.0119	.0207
Y_c (vertical) translation 0.1	.0020	.0007
Z_c (distance) translation 0.1	.0119	.0119
X_c (epipolar) translation 0.5	.0596	.1168
Y_c (vertical) translation 0.5	.0102	.0139
Z_c (distance) translation 0.5	.0574	.0572

Table 2.1: RMS *change* to relative position estimates of world points, caused by disturbing one of the cameras after calibration. Camera distance 10 units.

The two models are affected similarly by small movements, the worst of which is $X_c\!:\!Y_c$ rotation about the optical axis (this is the only motion which, to first order, changes the $\mathbf{Q}$ matrix of world–image coefficients).

Perspective stereo is more sensitive to larger movements, and to rotations and translations in the epipolar plane (in which a small error can induce large changes of perceived depth), because it distorts nonlinearly.

2.7 Discussion

For a typical stereo setup with two cameras fixating on a compact scene, perspective effects are small, and the epipoles will be far outside the image frames. In this case, a linear model of the epipolar constraint is valid, and the errors due to the linear approximation become comparable to other sources of error such as 'noisy' image measurements from trackers or feature detectors. It should be noted that the conditions required for linear epipolar geometry are *weaker* than those for the affine stereo model itself, which is accurate for camera distances more than ≈ 10 times the size of the scene.

Calibration is easier with affine stereo because the system has fewer parameters and is amenable to solution by linear techniques. Even if it could be calibrated accurately, the projective model is still more sensitive to errors and unexpected camera movements after calibration. The linear form of the affine stereo model makes it quite robust to calibration errors and changes. Even without calibration, it affords an approximate affine reconstruction of any scene with more than 4 corresponding points.

We do not attempt to use affine stereo to reconstruct *absolute* positions of points in the scene (as would be used by a look-and-move manipulation system). That would require accurately calibrated perspective camera models. Instead, we propose to use the affine model to match and reconstruct small objects in the scene, and to estimate the relative positions of nearby structures.

In chapter 3 the affine stereo formulation introduced here is used at the heart of a visual feedback controller for executing a grasp operation specified in terms of a pair of images of a graspable surface.

Chapter 3

Uncalibrated Stereo Visual Feedback

The core task in hand–eye coordination is to align a robot with a visually specified target. This chapter describes the use of visual feedback of gripper position and orientation to align it with the target object. The system does not require calibration, but estimates the affine stereo coefficients by making three deliberate motions. It is even robust to small camera motions during operation.

3.1 Introduction

If a stereo vision system were calibrated precisely, then the robot's gripper could be sent directly to the coordinates of a visually-specified target. However, this open-loop approach is sensitive to errors in calibration and kinematic modelling. Instead, we *track* the robot's gripper as it approaches the target, using *visual feedback* to correct the errors in its trajectory.

Affine stereo is a simplified stereo vision formulation that is very easily calibrated, but it is of limited open-loop accuracy. Nevertheless, it gives reliable *qualitative* information about the relative positions of points and can, of course, indicate when they are in precisely the same place. We therefore use it as part of a visual feedback loop to align the robot gripper with its target, which is a planar facet of the object to be grasped. Image-based feedback is used to null the error in the images, so as to align their position and orientation despite camera modelling errors.

3.2 Theory

3.2.1 Point to point alignment

First, we consider aligning a point attached to the robot (or defined in terms of an affine coordinate system based on the robot) with a point specified in the images.

Let the point on the robot be P. Its position is determined by a vector of (at least 3) joint settings, $\mathbf{\Theta}$, which are related to Cartesian coordinates by the *kinematic* function $\mathcal{K}$:

$$\mathbf{X}_P = \mathcal{K}(\mathbf{\Theta}). \tag{3.1}$$

$\mathbf{X}_P = [X_P\, Y_P\, Z_P]^T$, its world coordinates in a Euclidean frame. We wish to align the robot with a visually-specified 'set point' S, specified by image coordinates $\mathbf{u}_S = [u_S\, v_S\, u'_S\, v'_S]^T$. Using the affine stereo model, we estimate its position:

$$\hat{\mathbf{X}}_S = \hat{\mathbf{Q}}^+(\mathbf{u}_S - \hat{\mathbf{u}}_0), \tag{3.2}$$

This is the inverse of equation (2.8), where $\hat{\mathbf{Q}}^+$ models the left pseudo-inverse of $\mathbf{Q}$. Suppose that we also have an inverse model $\hat{\mathcal{K}}^{-1}$ of the robot's kinematic function, (if there are more than 3 joints, assume that the redundant degrees of $\mathbf{\Theta}$ are constrained in an appropriate way). We could attempt to send the robot directly to the configuration corresponding to $\mathbf{u}_S$:

$$\mathbf{\Theta}_{OL} = \hat{\mathcal{K}}^{-1}(\hat{\mathbf{X}}_S) \tag{3.3}$$

This is the 'look-and-move' approach. It fails to compensate for inaccuracies in the inverse kinematic model $\hat{\mathcal{K}}^{-1}$ and in the camera model ($\hat{\mathbf{Q}}$, $\hat{\mathbf{u}}_0$) as well as for errors due to strong perspective distortion.

Visual feedback

By tracking the robot's gripper, we can also obtain from its image position $\mathbf{u}_P$ an estimate $\hat{\mathbf{X}}_P$ of its world coordinates,

$$\hat{\mathbf{X}}_P = \hat{\mathbf{Q}}^+(\mathbf{u}_P - \hat{\mathbf{u}}_0). \tag{3.4}$$

Feeding back the relative position term $\hat{\mathbf{X}}_P - \hat{\mathbf{X}}_S$, a simple proportional control law [120] may be devised to null the error:

$$\dot{\mathbf{\Theta}} = -g\hat{\mathbf{J}}_{\mathcal{K}}^{-1}(\hat{\mathbf{X}}_P - \hat{\mathbf{X}}_S). \tag{3.5}$$

$\hat{\mathbf{J}}_{\mathcal{K}}^{-1}$ models the inverse differential kinematic relation [48] at the current robot configuration, and g is an appropriate gain constant. The use of a term such as $\hat{\mathbf{X}}_P - \hat{\mathbf{X}}_S$ is known as *position-based feedback*. We can also express the control law in terms of the image coordinate error term $\mathbf{u}_P - \mathbf{u}_S$ (*image-based feedback*). We use our estimates of the camera and kinematic models to provide a suitable gain:

$$\dot{\boldsymbol{\Theta}} = -g\hat{\mathbf{J}}_{\mathcal{K}}^{-1}\hat{\mathbf{Q}}^{+}(\mathbf{u}_P - \mathbf{u}_S). \tag{3.6}$$

We note that, according to the affine stereo model, position-based and image-based feedback are equivalent. This is because the world–image relation is modelled as linear (cf. [49]). The combined kinematic-and-vision relation $\mathbf{Q}\mathbf{J}_{\mathcal{K}}$ (inverted in (3.6)) is sometimes called the *image Jacobian* [48].

Discrete implementation

In practice, due to the limited bandwidth between the computer vision system and the robot controller, visual feedback is implemented as a *discrete* series of relative motions of the gripper:

$$\mathbf{X}_P^*|_{t+1} = \mathbf{X}_P^*|_t - k\hat{\mathbf{Q}}^{+}(\mathbf{u}_P - \mathbf{u}_S), \tag{3.7}$$

where $\mathbf{X}_P^*$ is the vector of world coordinates passed to the inverse kinematic model; that is, $\boldsymbol{\Theta} = \hat{\mathcal{K}}^{-1}(\mathbf{X}_P^*)$. The gain term, k, governs the rate of convergence.

Convergence criteria

When does the visual feedback loop converge to the set point and when is it unstable? Define $\mathbf{X}_{err}^* = \mathbf{X}_P^* - \mathbf{X}_S^*$ where $\mathbf{X}_S = \mathcal{K}(\hat{\mathcal{K}}^{-1}(\mathbf{X}_S^*))$, and suppose that the error is small, so that a first order model of $\mathcal{K}$ may be used. Equation (3.7) becomes:

$$\mathbf{X}_{err}^*|_{t+1} = (\mathbf{I} - k\hat{\mathbf{Q}}^{+}\mathbf{Q}\mathbf{J}_{\mathcal{K}}\hat{\mathbf{J}}_{\mathcal{K}}^{-1})\ \mathbf{X}_{err}^*|_t\,. \tag{3.8}$$

The error term will vanish and the system converge to the set point[1] only if all the eigenvalues of $(\mathbf{I} - k\hat{\mathbf{Q}}^{+}\mathbf{Q}\mathbf{J}_{\mathcal{K}}\hat{\mathbf{J}}_{\mathcal{K}}^{-1})$ have absolute magnitude below unity [122]. For a perfectly modelled system, $\hat{\mathbf{Q}}^{+}\mathbf{Q}\mathbf{J}_{\mathcal{K}}\hat{\mathbf{J}}_{\mathcal{K}}^{-1} = \mathbf{I}$ and the set point is reached in one step by setting $k = 1$.

Setting $0 < k < 2$ also leads to convergence, but values above 1 will cause overshoots and ringing (which, in a robotic application, could lead to collisions!) To prevent this whilst allowing for some inaccuracy in kinematic and camera modelling, k should be set significantly below unity, e.g. $k = 0.75$.

[1] We assume here that the set point is stationary.

3.2.2 Position and orientation alignment

Suppose now that we wish to align a planar surface on the robot's end effector with one specified in the image. Alignment of position and surface orientation is a 5-DOF constraint; additionally, if a vector on the robot is to be aligned with a distinguished image direction, there are constraints on all 6 components of robot pose.

Recall from section 2.5, that the orientation of a surface is encoded by its *affine transformation between views*, $\mathbf{A}$. This has only two degrees of freedom and may be represented by two components (a_{11}, a_{12}); the other two components can be obtained using the epipolar constraint. An image-based representation of surface orientation[2] is thus the vector $\mathbf{o} = [a_{11}\, a_{12}]^T$. The surface normal direction itself may easily be obtained from $\mathbf{o}$ and an estimate of the $\mathbf{Q}$ matrix.

Image-based feedback of surface orientation

Let the robot now be controlled in terms of a desired position and orientation, where the orientation is expressed in image-based terms:

$$\mathbf{\Theta} = \hat{\mathcal{K}}^{-1}(\mathbf{X}^*_P,\, \hat{\mathcal{F}}(\mathbf{o}^*_P)), \tag{3.9}$$

where $\hat{\mathcal{K}}^{-1}$ is an inverse kinematic model for both position and orientation control, and $\hat{\mathcal{F}}$ is a function to convert image-based orientations into some other parameterisation used by the robot.[3]

A suitable control law to align the gripper with a target is:

$$\dot{\mathbf{\Theta}} = -g(\frac{\partial \mathbf{\Theta}}{\partial \mathbf{X^*_P}} \hat{\mathbf{Q}}^+(\mathbf{u}_P - \mathbf{u}_S) + \frac{\partial \mathbf{\Theta}}{\partial \mathbf{o^*_P}}(\mathbf{o}_P - \mathbf{o}_S)). \tag{3.10}$$

Again, in practice it is convenient to use a discrete implementation, in which a sequence of position and orientation demands are made:

$$\begin{aligned} \mathbf{X}^*_P|_{t+1} &= \mathbf{X}^*_P|_t - k_1\hat{\mathbf{Q}}^+(\mathbf{u}_P - \mathbf{u}_S), \\ \mathbf{o}^*_P|_{t+1} &= \mathbf{o}^*_P|_t - k_2(\mathbf{o}_P - \mathbf{o}_S). \end{aligned} \tag{3.11}$$

As before, the gain parameters k_1, k_2 should be set to values between 0 and 1.

[2] A third image-based parameter may be added to this vector to specify all 3 DOF of orientation.

[3] $\hat{\mathcal{F}}$ depends on an estimate of $\mathbf{Q}$.

3.3 Simulations

A simple articulated robot was simulated, its origin at coordinates (−2.0, 0, −.5), with two links of length 1.5 units (figure 3.1). The position of the end-effector was governed by three angles: *waist* (θ_1), *shoulder* (θ_2) and *elbow* (θ_3); the kinematic function was:

$$\begin{aligned} X_P &= 1.5\cos\theta_1(\cos\theta_2 + \cos(\theta_2 - \theta_3)) - 2.0, \\ Y_P &= 1.5\sin\theta_1(\cos\theta_2 + \cos(\theta_2 - \theta_3)), \\ Z_P &= 1.5(\sin\theta_2 + \sin(\theta_2 - \theta_3)) - 0.5. \end{aligned} \tag{3.12}$$

The same simulated cameras were used here as in section 2.6, facing the origin from a distance of 4.0 units. The inter-camera angle was 20°. Affine stereo coefficients were estimated by observing 4 known points in a tetrahedron within the unit cube.

Setpoints were enumerated on a dense grid of points within the unit square and the robot aligned with those points:

- **Open loop**, by inverting kinematic and camera models;
- **Closed loop**, using visual feedback with $k = 1$;
- **Closed loop**, using visual feedback with $k = 0.5$.

When using visual feedback, the initial position of the end effector in each trial was the world origin in the centre of the unit cube.

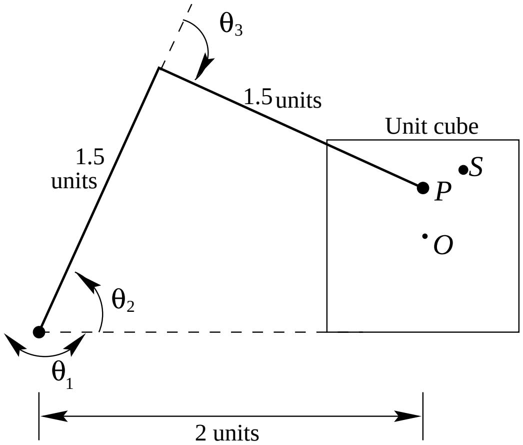


Figure 3.1: Articulated robot model used in the simulations

I. Ideal case

An inverse kinematic model [120] was derived analytically from equation (3.12), and the camera coefficients estimated using noiseless reference points. In open loop, the RMS positioning error for a point within the cube was .068 units, and the maximum error .157 units. These errors are due to perspective distortion. With visual feedback, however, the errors are reduced practically to zero (results are summarised in table 3.1). Figure 3.2(a) shows the trajectory of the robot when the set point is $(.5, .5, .5)$ with $k = 0.5$. It is almost a straight line.

II. With erroneous kinematic model

The simulations were repeated, using a modified inverse kinematic model which moved θ_1 through 1.5 times the desired angle and added a 10° offset to θ_3. This seriously degraded open-loop positioning accuracy; however visual feedback with $k = 0.5$ was able to correct the errors. In this case, better performance was obtained with $k = 0.5$ than with $k = 1$, which lead to 'ringing' and failure to converge in some regions of the robot's configuration space. See figure 3.2(b).

III. After camera disturbances

This time the correct kinematic model was used, but the camera pose was changed between observation of the reference points and alignment with the set points. One of the cameras was translated 0.25 units upwards, the other rotated 10° about its optical axis. Again, visual feedback was able to null the errors. See figure 3.2(c).

	Open loop		$k = 1$		$k = 0.5$	
	RMS	Max	RMS	Max	RMS	Max
No disturbance	.068	.157	.0001	.0007	.013	.034
Kinematic error	.263	.401	.036	.162	.012	.026
Camera disturbance	.197	.361	.003	.023	.025	.071

Table 3.1: Results of simulations in which the end-effector was aligned with points on a dense grid within the unit cube. RMS and maximum positioning errors after 6 iterations are shown.

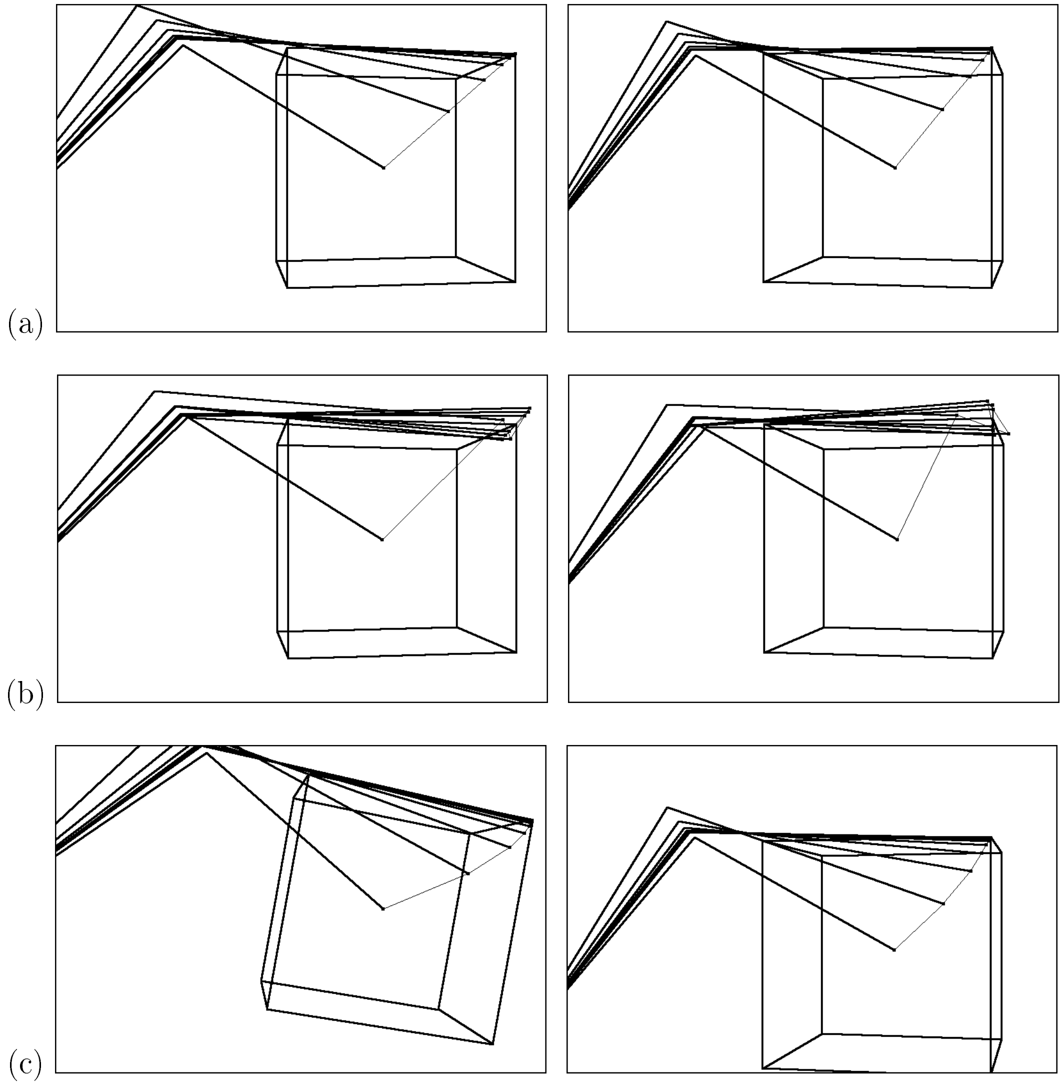


Figure 3.2: Simulated robot trajectories under visual feedback. The end effector is converging on one corner of the unit cube: (a) ideal case ($k = 0.5$); (b) with erroneous kinematic model ($k = 1$) showing 'ringing' behaviour; (c) after camera disturbances ($k = 0.5$).

3.4 Experiment

3.4.1 Setup

When the system was started up, it began by opening and closing the jaws of the robot's gripper. By observing the image difference, it was able to locate the gripper and set up a pair of affine trackers as instances of a hand-made 2-D template. The trackers could then follow the gripper's movements continuously. Stereo tracking was implemented on the Sun at over 10 Hz. The robot then made a series of deliberate motions, moving to four preset points to estimate the coefficients matrix $\mathbf{Q}$.

Since the reference points used to self-calibrate were specified in the *controller's* coordinate space ($\mathbf{X}^*$), linear errors in the kinematic model were effectively bypassed. The system must still cope with any nonlinearities in control, as well as those caused by strong perspective effects.

A target object was located by similar means — by observing the image changes when it was placed in the manipulator's workspace. Alternatively it could be selected from a monitor screen using the mouse. There was no pre-defined model of the target shape, so a pair of 'exploding' B-spline snakes [21] were used to automatically locate the contours delimiting the target surface in each of the images. The snakes were converted into a pair of affine trackers, by re-expressing their sampling points in terms of an affine basis (see appendix).

The target surface was then tracked along with the gripper, to compensate for unexpected motions of either the target or the cameras during operation.

3.4.2 Visual feedback loop

The orientation of the gripper of a 5-DOF manipulator is constrained by its lack of a 'yaw' axis, and the constraint changes continuously as it moves. To avoid this problem, the test implementation kept the gripper vertical, reducing the number of degrees of freedom to four. Its orientation could then be described by a single *roll angle*. It was assumed that the target plane was also vertical. Their image orientations were therefore described by a single quantity, a_{11}.[4]

The gains for position and orientation control are set well below unity at 0.75, to prevent instability, even when the vision system is miscalibrated. The control structure of the system is shown in figure 3.4.

[4] It is assumed that the camera baseline is roughly horizontal, so that a_{11} varies with roll angle.

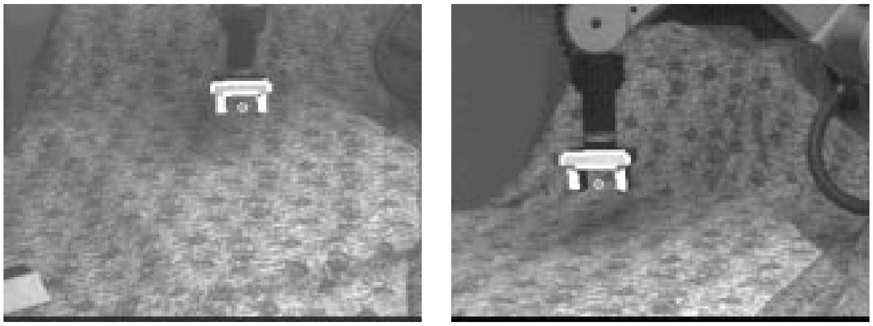


Figure 3.3: A stereo pair showing the robot gripper at one of the four reference points used for calibration. Active contour models are overlaid in white.

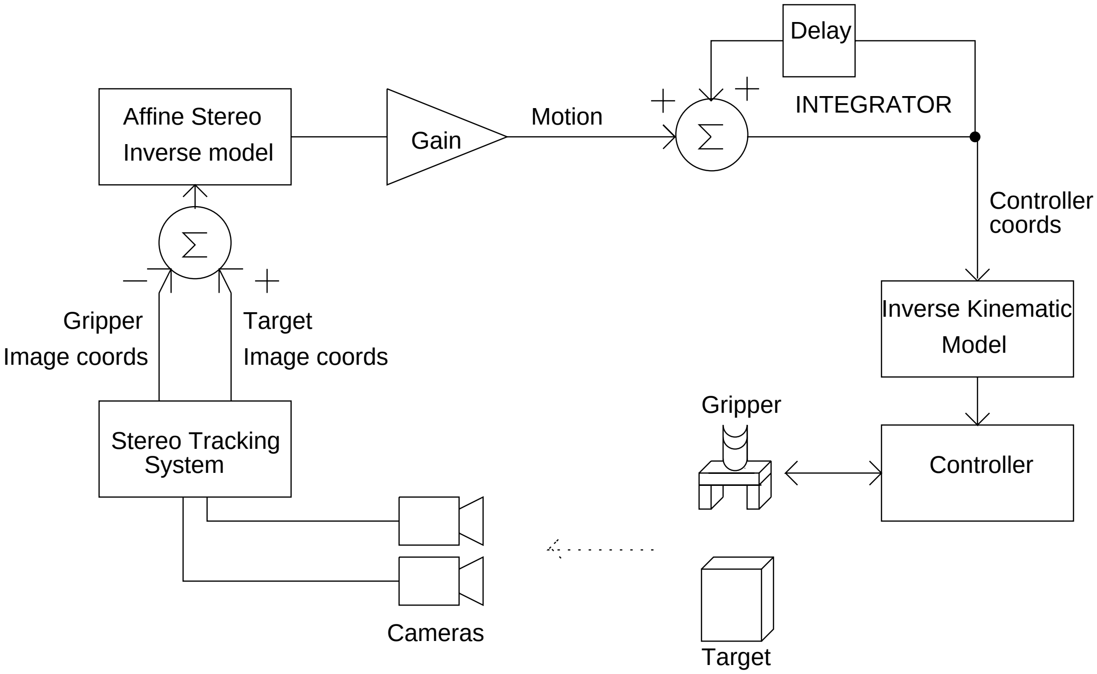


Figure 3.4: The control structure of the system, showing the use of visual feedback.

3.4.3 Tracking and grasping behaviours

Without modification, the visual feedback loop would attempt to superimpose the robot gripper and target object in the images. By offsetting $\mathbf{u}_P$ from the gripper's centre, we introduce a constant offset between gripper and target in space; the offset is defined in terms of a coordinate system attached to the gripper (in fact, the affine basis of the tracking mechanism), so that it will be invariant to motions of the cameras. We set the offset so that the robot tracks the target object continuously, hovering a few centimetres above a point on its top surface (figure 3.5).

Once this *pre-grasp* position has been achieved, the object may be grasped reliably using a pre-programmed motion, which consists of rotating the gripper through 90° and translating downwards (figure 3.6). Depending on the type and shape of object to be grasped, some other grasping motion could be substituted here.

3.4.4 Results

Without feedback control, the robot locates its target only approximately (typically to within 5cm in a 50cm workspace). With a feedback gain of 0.75 the gripper converges on its target in three or four control iterations. If the system is not disturbed it will take a straight-line path. The system has demonstrated its robustness by continuing to track and grasp objects despite:

Kinematic errors. Linear offsets or scalings of the controller's coordinate system are absorbed by the self-calibration process with complete transparency. Slight nonlinear distortions to the kinematics are corrected for by the visual feedback loop, though large errors introduce a risk of ringing and instability unless the gain is reduced.

Camera disturbances. The system continues to function when its cameras are subjected to small translations, rotations and zooms, even after it has self-calibrated. Large disturbances to camera geometry cause the gripper to take a curved path towards the target, and require more control iterations to get there.

Strong perspective. The condition of weak perspective throughout the robot's workspace does not seem to be essential for image-based control and the system can function when the cameras are as close as 1.5 metres (the robot's reach is

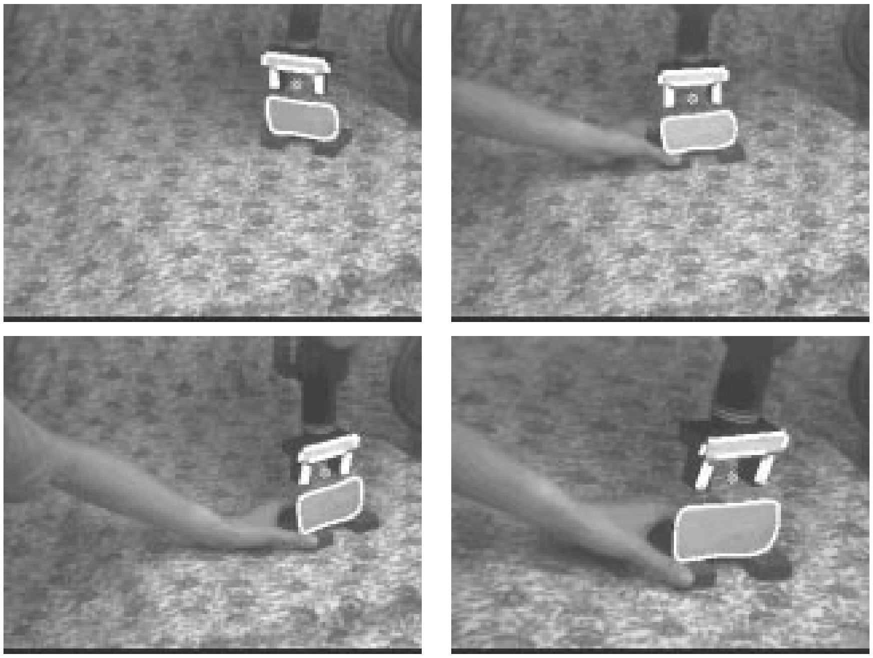


Figure 3.5: The robot is tracking its quarry, guided by the position and orientation of the target contour (view through left camera). On the target surface is an *affine snake* — an affine tracker obtained by 'exploding' a B-spline snake from the centre of the object. Last frame: one of the cameras has been rotated and zoomed, but the system continues to operate successfully with visual feedback.

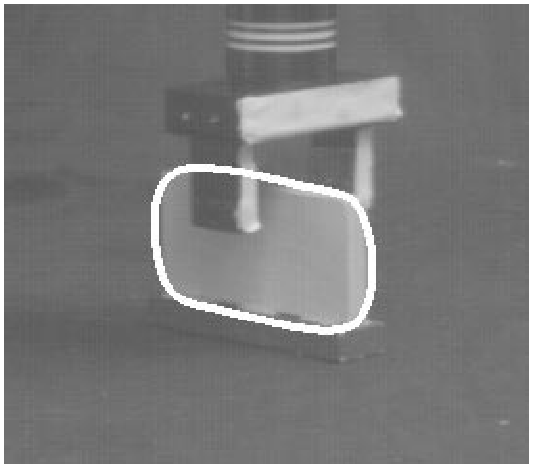


Figure 3.6: Robot grasping a planar target, using an active contour to recover its size and orientation. The gripper is not tracked during the grasping manoeuvre.

a little under 1 metre). However the feedback gain must be reduced to below 0.5, or the system will overshoot on motions towards the cameras.

Figure 3.5 shows four frames from a tracking sequence (all taken through the same camera). The cameras are about two metres from the workspace. Tracking of position and orientation is maintained even when one of the cameras is rotated about its optical axis and zoomed.

3.4.5 Why not track Q?

Since the visual feedback system has been designed to be robust to changes in the camera parameters (caused by movement of the cameras) during operation, an obvious question is whether or not efficiency can be improved by tracking these changes. This was attempted in a version of the above experiment, using a Kalman filter [42] whose state vector encodes the camera model $(\hat{\mathbf{Q}}, \hat{\mathbf{u}}_0)$, which is updated from subsequent observations of the robot gripper. However, this conferred little or no detectable benefit to the performance of the system.[5] This is because it is impossible, from a single observation of the gripper, to determine if an error in the image location of the gripper is due to:

1. Strong perspective (temporary change in $\mathbf{Q}$, $\mathbf{u}_0$),
2. Change in $\mathbf{u}_0$ caused by small camera rotations or translations,
3. Change in $\mathbf{Q}$ caused by large camera translations, zooming, or rotation about the optical axis.

Only the last of these warrants tracking, and this was the least frequent change to be observed. The errors due to perspective could to some extent be modelled by ‘observation noise,’ but there were no obvious values for ‘process noise’ to enable the other parameter changes to be distinguished. It was concluded that attempting to track camera motions was not only ill-conditioned but also unnecessary.

[5] Except for a validation gate on the gripper’s image coordinates, which was very useful for detecting failures of the trackers and reinitialising them.

3.5 Discussion

Here the effectiveness of affine stereo has been demonstrated for the task of aligning a robot with a visually specified target, in both position and orientation. In a discrete-time implementation, rapid convergence is achieved with a gain of unity; though if the system is disturbed from its initial configuration, the gain should be reduced to maintain stability and prevent overshoots which could lead to collisions.

The visual servoing system does not require camera calibration, but makes a small number of deliberate motions to actively estimate the relation between hand and eye. Even these are not always necessary, for instance if the cameras have been rotated and then realigned by hand, the previous estimate of $\mathbf{Q}$ will normally still be valid. It is not necessary, or even practical, to track these coefficients over time.

By defining the working coordinate system in terms of the robot's abilities, linear errors in its kinematics are bypassed. The remaining nonlinearities can be handled using visual feedback. We have shown that this can be achieved cheaply and effectively using a novel form of active contour to track planar features on the gripper and target.

Such a system has been implemented and found to be highly robust, without unduly sacrificing performance (in terms of speed to converge on the target).

Chapter 4

Indicating the Target Object

This chapter describes a human–computer interface which tracks a pointing hand, in order to specify objects and locations for robotic pick and place operations. The system is implemented using uncalibrated stereo vision.

4.1 Introduction

In order to make use of visual feedback in uncalibrated stereo, the target object must be indicated to the system in terms of *image* measurements. If there is more than one object visible in the scene, some means must be chosen to select the desired object for grasping, and to indicate the place to which it is to be moved.

This could be accomplished using a mouse to indicate points in one or both images. This is reliable if somewhat inelegant, and requires a workstation, or similar user-interface hardware, in close proximity to the work area. Alternatively, the operator could interact with the cameras already in place to indicate the target directly. The latter approach is explored here. An interface based on *pointing* is developed, to select objects on a planar table top.

We use a pair of monochrome cameras to observe the robot's work space and pointing hand in stereo. Active contours are employed to track the hand in real time. Using a simple result from projective geometry, the system can calculate where the hand is pointing to on the plane, without camera calibration, to an accuracy of about 10mm.

4.2 Geometrical framework

A single view of a pointing hand is ambiguous: its distance from the camera cannot be determined, and the 'slant' of its orientation cannot be measured with any accuracy. This means that the 'piercing point', where the line defined by the hand intersects the work surface, is constrained to a line, which is the projection of the hand's line in the image. A second view is needed to fix its position in two dimensions [106].

4.2.1 Viewing the plane

Consider a pinhole camera viewing a plane. The viewing transformation is a plane collineation between some world coordinate system (X, Y), and image plane coordinates (u, v), thus:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \sim \mathbf{T} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}, \tag{4.1}$$

where $\mathbf{T}$ is a 3×3 transformation matrix. The full perspective form of the transformation is used in this case because the workspace will generally be large and possibly foreshortened in one or both images.

The system is homogeneous, so we can fix $t_{33} = 1$ without loss of generality, leaving 8 degrees of freedom. To solve for $\mathbf{T}$ we must observe at least four points. By assigning arbitrary world coordinates to these points (e.g. $(0, 0)$, $(0, 1)$, $(1, 1)$, $(1, 0)$), a new coordinate system on the plane is defined, which we call *working plane coordinates.*

Now, given the image coordinates of a point anywhere on the plane, along with the image coordinates of the four reference points, it is possible to invert the relation and recover the point's working plane coordinates, which are invariant to the choice of camera location [88]. The same set of reference points in the world can be observed in a stereo pair of views, to compute two transformations $\mathbf{T}$ and $\mathbf{T}'$, one for each camera.

4.2.2 Recovering the indicated point in stereo

With natural human pointing behaviour, the hand is used to define a line in space, passing through the base and tip of the index finger. This line will not generally be in the ground plane but intersects the plane at some point. It is this point (the *'piercing point'* or *'indicated point'*) that we aim to recover. Let the pointing finger lie along the line l_w in space (see figure 4.1). Viewed by a camera, it appears on line l_i in the image, which is also the projection of a *plane*, $\mathcal{P}$, passing through the image line and the optical centre of the camera. This plane intersects the ground plane $\mathcal{G}$ along line l_{gp}. It can be seen that l_w lies in $\mathcal{P}$, and the indicated point in l_{gp}, but from one view we cannot see exactly where.

Note that the line l_i is an image of line l_{gp}; that is, $l_i = \mathbf{T}(l_{gp})$, where $\mathbf{T}$ is the projective transformation[1] from equation (4.1). If the four reference points are visible, this transformation can be inverted to find l_{gp} in terms of the working plane coordinates. The indicated point is constrained to lie upon this line on the plane.

Repeating the above procedure with the second camera C$'$ gives us another view l'_i of the finger, and another line of constraint l'_{gp}. The two constraint lines will intersect at a point on the ground plane, which is the indicated point. Its position can now be found in terms of the projective basis formed from the four reference points. This is similar to a construction used by Quan and Mohr [106], who present an analysis based on cross-ratios. Figure 4.2 shows the lines of pointing in a pair of images, and the intersecting constraint lines in a 'canonical' view of the working plane (in which the reference point quadrilateral is transformed to a square).

By transforming this point with matrices $\mathbf{T}$ and $\mathbf{T}'$, the indicated point can be projected back into image coordinates. Although the working plane coordinates of the indicated point depend on the configuration of the reference points, its back-projections into the images do not. Because all calculations are restricted to the image and ground planes, explicit 3-D reconstruction is avoided and no camera calibration is necessary. By tracking at least four points on the ground plane, the system can be made insensitive to camera motions.

4.2.3 Projective versus affine transformations

Assuming a weak perspective view of the plane, we could substitute an affine transformation between views for the projective one: this would require only 3 reference points. However, in this case there is little gain in robustness or simplicity using the affine model that would offset the loss of accuracy caused by perspective distortion.

[1] This is a slight abuse of notation, since for the standard representation of a line the appropriate transformation matrix is $\mathbf{T}^{-1}$. Here $\mathbf{T}()$ refers abstractly to a plane projective transformation which may be applied to points, lines or other image features.

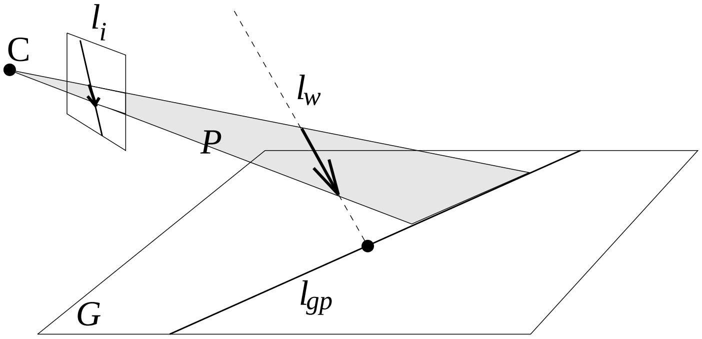
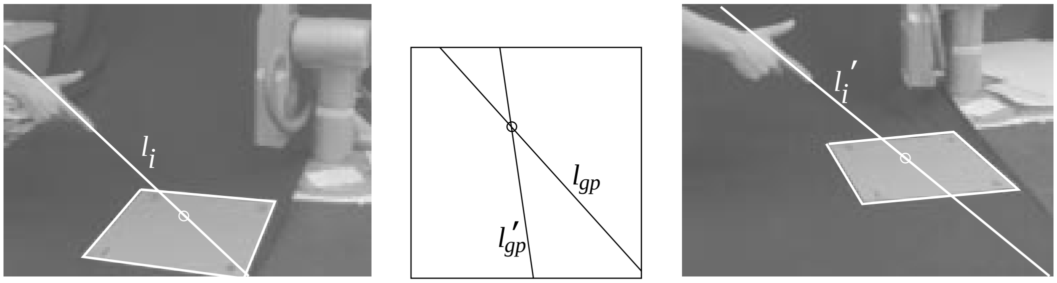


Figure 4.1: Relation between lines in the world, image and ground planes



(a) (b) (c)

Figure 4.2: Pointing at the plane. By taking the lines of pointing in left and right views (a, c), transforming them into the canonical frame defined by the four corners of the grey rectangle (b), and finding the intersection of the lines, the indicated point can be determined; this is then projected back into the images.

This is because we are considering points on a *plane*, using only 2-D projective transformations: these do not suffer the same sensitivity to noise as would a full 3-D reconstruction.[2] Errors in localising the 4 reference points result in only local inaccuracies in the projective transformation (see table 4.1, page 65).

We are interested, in the first instance, in the *open-loop* accuracy with which the indicated point may be recovered. With the camera setup used in these experiments, the ground plane is large and significantly foreshortened, and this would cause significant errors in a formulation based on affine transformations.

4.2.4 Pointing in a multi-faceted environment

The above geometrical framework relies on the target surface being planar in order to estimate the constraint lines l_{gp}, l'_{gp} and their intersection. This can be extended to environments consisting of more than one plane.

For each planar surface, we need 4 corresponding points, and a description of the surface's boundary, e.g. as a polygon, in either view (recall that the 4 points define a transformation between views, allowing the boundary to be 'transferred' into the other image). Given two views of a pointing hand, we can now ascertain which facet is being pointed to as follows:

- For each facet, test if the pointing line in each view intersects the facet's image boundary in that view.
- If so, solve for the *piercing point* and test that it too lies within the boundary of the facet.
- Where the pointing line intersects more than one facet, choose the one nearest to the fingertip. Distances to the fingertip of points along this line may be compared in either image.

Note that whilst this requires at least 4 correspondences per facet,[3] and *a priori* models of the surfaces and their boundaries in the images, the entire process is image-based and does not rely on a 3-D reconstruction of the hand or the environment.

[2] This is partly due to our choice of working plane coordinates and the use of four reference points in a rectangle, resulting in a well-conditioned **T** which is close to an affine transformation.

[3] For smaller facets which are not strongly foreshortened, 3 correspondences may suffice and an affine stereo model can be used.

4.3 Tracking a pointing hand

4.3.1 Background

There has been a lot of interest lately in the use of hand gestures for human–computer interfacing: they are intuitive for the operator, and provide a rich source of information to the machine. This type of interface is particularly appropriate in applications such as virtual reality, multimedia and teleoperation [123, 40, 9]. Most current commercial implementations rely on sensors that are physically attached to the hand, such as the 'DataGlove' [39]. More recently, systems have been proposed using *vision* to observe the hand. Some require special gloves with attachments or markings to facilitate the localisation and tracking of hand parts [135, 26], but others operate without intrusive hardware. This is attractive because it is convenient for the user and potentially cheaper to implement.

A large number of systems have been proposed for visual tracking and interpretation of hand and finger movements without gloves. These systems can broadly be divided into:

- those concerned with gesture identification (e.g. for sign language), which compare the image sequence with a set of standard gestures using correlation and warping of the templates [29], or classify them with neural networks [13];

- those which try to reconstruct the pose and shape of the hand (e.g. for teleoperation) by fitting a deformable, articulated model of the palm and finger surfaces to the incoming image sequence [69].

Common to many of these systems is the requirement to calibrate the templates or hand model to suit each individual user. They also tend to have high computational requirements, taking several seconds per frame on a conventional workstation, or expensive multiprocessor hardware for real time implementation.

4.3.2 Approach

Our approach differs from these general systems in an important respect: **we wish only to recover the line along which the hand is pointing**, to be able to specify points on a ground plane. This considerably reduces the number of degrees of freedom which we need to track. Furthermore, because the hand must be free to move about as it points to distant objects, it will occupy only a relatively small fraction of the pixel area in each image, reducing the number of features that can be distinguished.

In this case it is not unreasonable to insist that the user adopt a rigid gesture. For simplicity, the familiar ‘pistol’ pointing gesture was chosen. The pointing direction can now be recovered from the image of the index finger, although the thumb is also prominent and can be usefully tracked. The rest of the hand, which has a complicated and rather variable shape, is ignored. This does away with the need to calibrate the system to each user’s hand.

4.3.3 Tracking mechanism

A form of edge-seeking active contour model [64, 22, 56] was used to track the image of a hand in the familiar ‘pointing’ gesture, in real time. The tracker is an active contour, resembling a B-Spline snake [22], but constrained to deform only affinely in the images. It is based on a template, representing the shape of the occluding contours of an extended finger and thumb (see figure 4.3).

The tracker’s motion is restricted to 2-D affine transformations in the image plane, which ensures that it keeps its shape whilst tracking the fingers in a variety of poses. This approach is suitable for tracking planar objects under weak perspective [12]; however it also works well with fingers, which are approximately cylindrical.

A first-order temporal filter is incorporated into the tracker, to predict the future position of the contour, improving its real-time tracking performance. The filter is biased to favour rigid motions in the image, and limits the rate at which the tracker can change scale — these constraints represent prior knowledge of how the hand’s image is likely to change, and increase the reliability with which it can be tracked. The dynamics of the tracker are described in more detail in appendix A. It is similar to the trackers we use to track the robot’s gripper in stereo images, to provide visual feedback.

To extract the hand’s direction of pointing, we estimate the orientation of the index finger by fitting a pair of parallel lines to its image edges. The base of the thumb is also tracked to define the length of the index finger, and to resolve an *aperture problem* [131] induced by the finger’s long thin shape.

4.4 Pointing experiment

The above geometrical framework and tracking mechanism were implemented, to indicate points on a planar table top with a pointing hand. The two cameras were about 2m from the scene, angled about 20° apart.

4.4.1 Setup

In this experiment, the corners of a coloured rectangle on the table-top were used to define the working coordinate system. A pair of finger-trackers (one for each camera) were initialised, one after the other, by the operator holding his or her hand up to a template in the image and waiting a few seconds while it 'moulded' itself to the contours of the finger and thumb. Once both trackers were running, the hand could be used as an input device by pointing to places on the table-top. In this implementation, the position and orientation of the finger trackers, and the indicated point on the plane, were updated about 10 times per second.

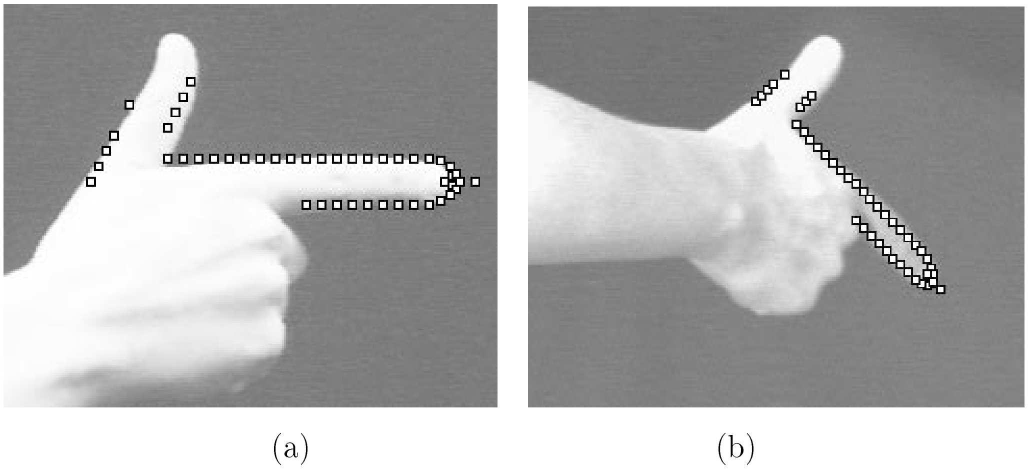
4.4.2 Performance

Figure 4.4 shows the system in operation. The corners of the white rectangle are the four reference points, and the overlaid square shows the position of the indicated point. Movements of the operator's hand caused corresponding movements of this point in real time.

Visual tracking can follow the hand successfully for several minutes at a time; however, abrupt or non-rigid hand movements could cause one or both of the trackers to fail. Because it samples the image only locally, a failed tracker will not correct itself unless the user makes a special effort to recapture it.

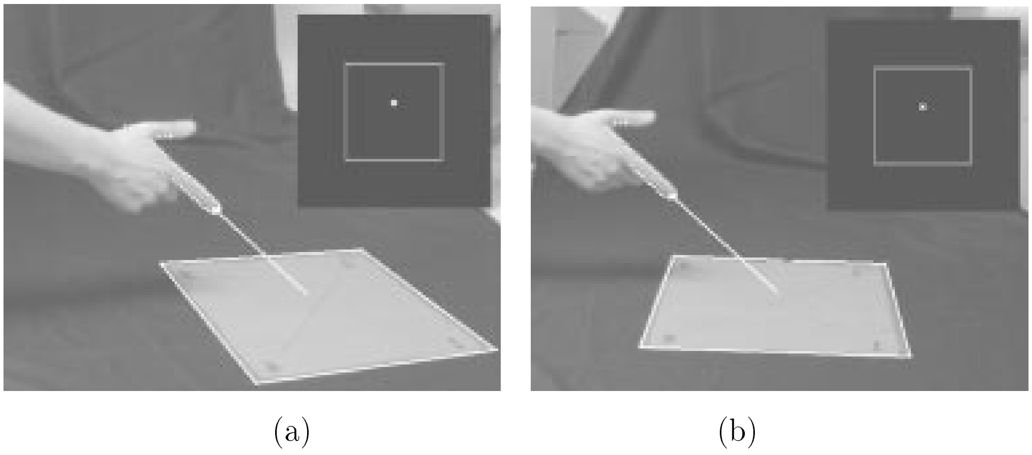
Users reported that the recovered point did not always correspond to their subjective pointing direction, which is related to the line of sight from *eye* to fingertip as well as the orientation of the finger itself. Initial subjective estimates of accuracy were in the order of 20–40mm. If the user received feedback by viewing the system's behaviour on a monitor screen, a resolution within 10mm could be achieved. It is a natural human skill to servo the motion of one's hand to control a cursor or other visual indication.

The system was also tested in a multi-planar environment (figure 4.5). The planes were represented by 9 given correspondences, which also defined bounding quadrilaterals. The user could then indicate points on 3 surfaces: transition between planes occurred automatically as the piercing point crossed their boundaries.



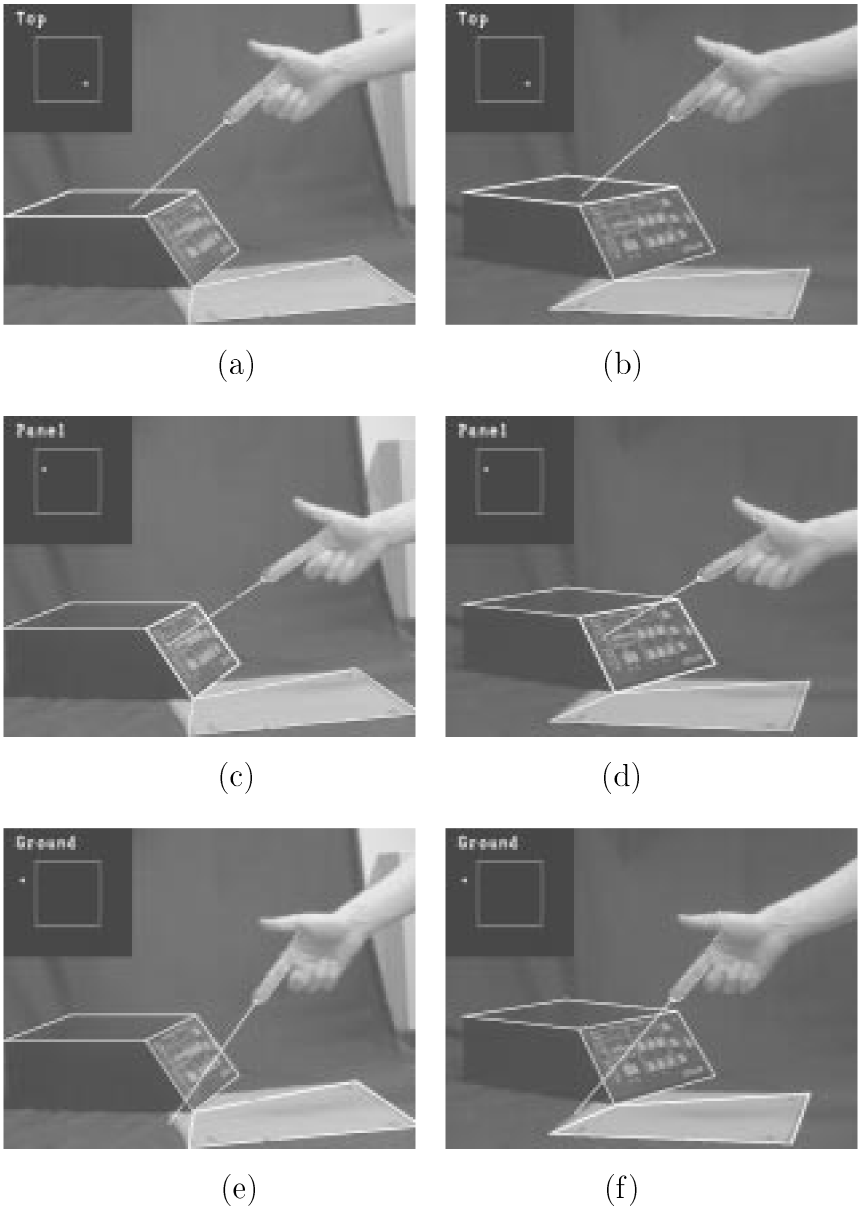
(a) (b)

Figure 4.3: The finger-tracking active contour (a) in its canonical frame (b) after an affine transformation in the image (to track a rigid motion of the hand in 3-D).



(a) (b)

Figure 4.4: Stereo views of a pointing hand. The two views are shown side by side. In each view an active contour is tracking the hand. The inlaid square is a representation of the indicated point in working plane coordinates.



(a) (b)

(c) (d)

(e) (f)

Figure 4.5: Pointing in a multi-planar environment: (a,b) pointing to the top surface of the object; (c,d) pointing to the sloping panel; (e,f) if the pointing line intersects neither of the above surfaces, it defaults to the ground plane.

4.4.3 Accuracy evaluation

To evaluate our system, we calculated the uncertainty of the image coordinates of the hand and reference points in our experimental setup. Using Monte Carlo methods, these were propagated into working plane coordinates, to assess the accuracy of the indicated point.

I. Finger tracker uncertainty

We can obtain a measure of uncertainty for the finger's position and orientation in the image by considering the *residual offsets* between modelled and observed image edges. These are the components of the normal offsets that remain after fitting a pair of parallel lines to model the index finger's occluding edges, with least-squares perpendicular error. They take into account the effects of image noise and occlusion, as well as pixel quantisation effects, and mismatches between the model and the actual shape of the index finger.

These offsets indicated that the image position of the finger's mid-line could be determined to sub-pixel accuracy (standard deviation typically $\sigma = 0.3$ pixels), and the orientation to an accuracy of 0.6°. From this uncertainty measure $\pm 2\sigma$ bounds were calculated for the lines l_i and l'_i; and, by projecting these onto the ground plane, the uncertainty in the indicated point could be estimated.

Figure 4.6 shows the results for three different configurations of the cameras, with a 95% confidence ellipse drawn around the indicated point. The constraint line uncertainties were much the same in each trial, but the uncertainty on the indicated point varied according to the separation between the stereo views: when the cameras were close together, the constraint lines were nearly parallel and tracker uncertainty became very significant (figure 4.6a); as the baseline was increased and the stereo views become more distinct, the constraint lines met at a greater angle and accuracy was improved (figure 4.6c).

II. Reference point uncertainty

In the above experiments, reference points were identified in the images by hand, and we assume an uncertainty of 1 pixel standard deviation (in an application, techniques exist to allow points or lines to be localised to higher accuracy, and errors may be reduced by observing more than 4 corresponding points – this is therefore a rather conservative estimate of accuracy).

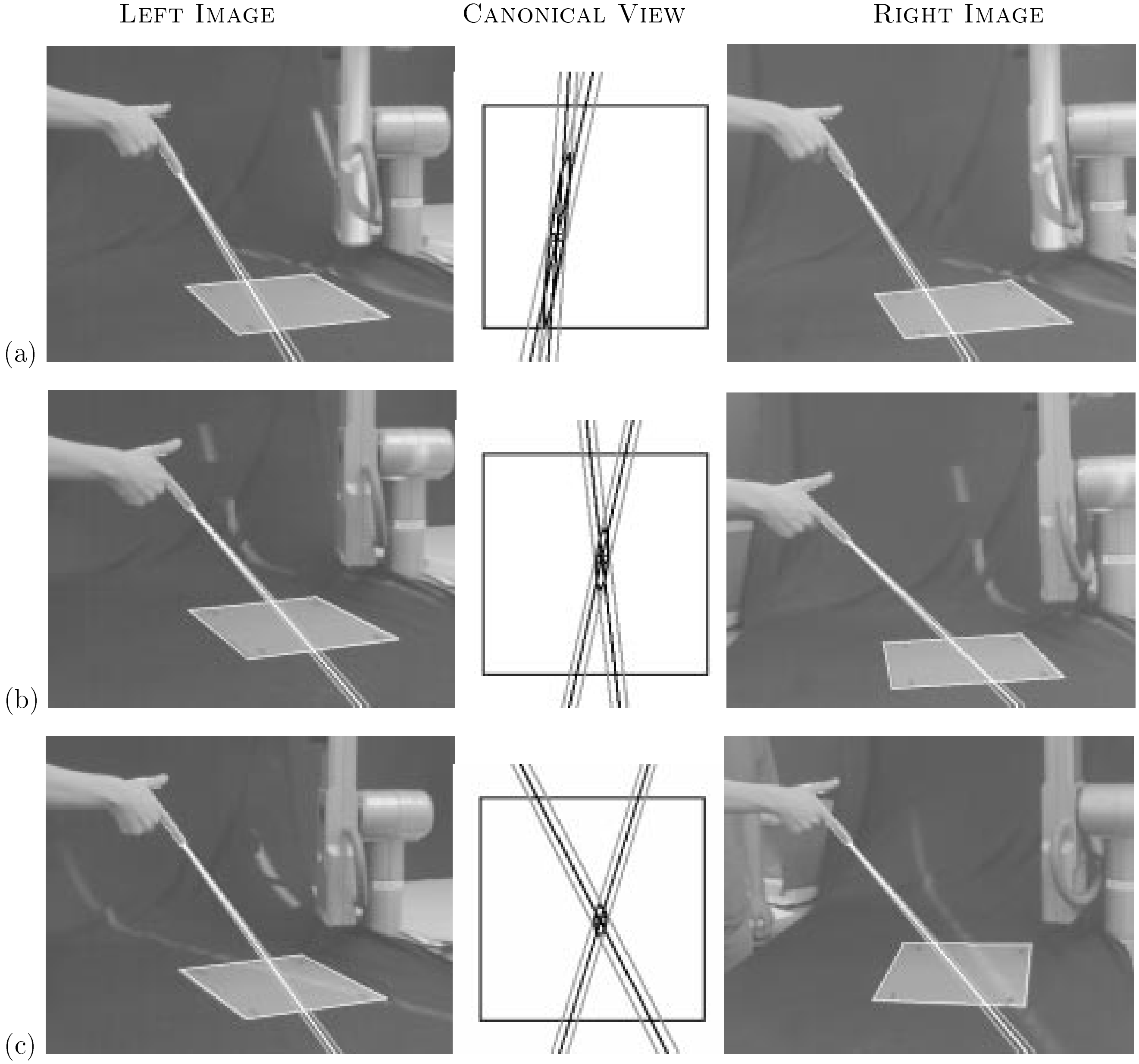


Figure 4.6: Indicated point uncertainty for 3 different camera configurations: 2σ bounds for the pointing lines, their projections into working plane coordinates, and error ellipses for the indicated point, when the angle between stereo views is (a) 7° (b) 16° (c) 34°. The uncertainty is greatest when the camera angle is small and the constraint lines nearly parallel.

We used Monte Carlo simulations (based around real-world configurations of cameras, hand and table) to assess the impact of this uncertainty on the coordinates of the indicated point. The results (table 4.1) show that this source of error is less significant than the tracker uncertainty, and confirm that the system is not especially sensitive to errors in the reference point image coordinates. Again, the errors were most significant when the camera separation angle was small.

Angle between the cameras	(i) Working plane coordinate error (with tracker noise)	(ii) Working plane coordinate error (with ref. point noise)	(iii) Working plane coordinate error (with both)
7°	.119	.040	.124
16°	.044	.019	.047
34°	.020	.008	.022

Table 4.1: Simulated RMS error in working plane coordinates, due to (i) tracker uncertainty derived from ‘residual offsets’ as detailed above; (ii) reference point image noise, $\sigma = 1$ pixel in each image; (iii) both. A value of 1.0 would correspond to a positioning uncertainty of about 40cm (the width of the reference point rectangle).

III. Experimental accuracy

Ground truth about the position and orientation of a human finger is, of course, very difficult to measure without intrusive equipment that could interfere with the stereo vision system. We therefore tested the accuracy of the pointing system using an artificial pointing device (figure 4.7). The test pointer was a white cylinder, about 15cm long, bounded by black end stops and wrapped around a rod which could be positioned by the robot arm to an accuracy of about 3mm. Whilst not identical to a human hand, it had approximately the same dimensions and was tracked in a similar manner.

A number of trials were carried out with the vision system tracking the rod as it was aligned with points on a grid on the target surface. The RMS error was 2.3% of the working plane coordinates, or 9mm in a 40cm workspace. The maximum reported error was 3.7% (15mm).

4.5 Robot control application

The proposed application for this stereo pointing system is to control a robot manipulator as it grasps and places small objects on a flat table-top. This time the four reference points were defined automatically by the robot itself in a plane a few centimetres above the table.

4.5.1 Setup

The reference points were defined by observing the robot gripper itself as it visited 4 known points in a plane. The robot began by opening and closing its gripper, and using the resulting image motion to initialize a pair of affine active contours (similar to those used to track the pointing hand, described in Appendix A). It was then tracked as it made deliberate motions across the plane. This not only defined the working coordinate system but related it to the robot's own world coordinate system. Finger-trackers were then initialised as before.

4.5.2 Performance

The robot was now instructed to move repeatedly to where the hand was pointing, in the horizontal working plane raised 50mm above the table-top. By watching the robot's motion, the operator was provided with a source of direct feedback of the system's output, allowing him or her to correct for systematic errors between subjective and observed pointing direction, and align the gripper over objects in the robot's workspace.

When the distance between hand and workspace is large, the system is sensitive to small changes in index finger orientation (as one would expect). To reduce this sensitivity, the operator maintains a steep angle to the horizontal, and points from a distance of less than 50cm from the plane, whilst still keeping his or her hand clear of the robot. One can then comfortably position the gripper with sufficient accuracy to pick up small objects (figure 4.8).

4.5.3 Using the interface to grasp objects

In experiments, it was found that two simple classes of object could be grasped reliably without any further planning:

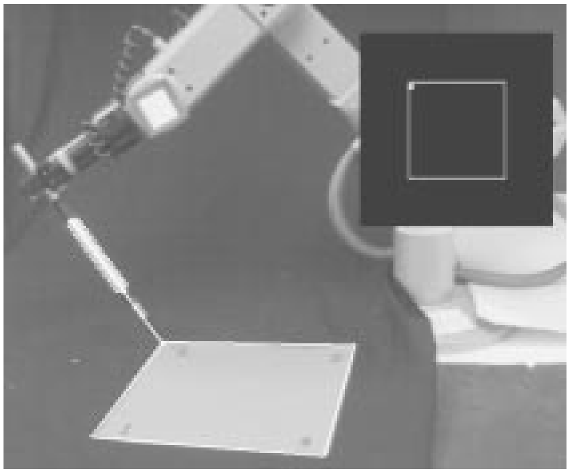
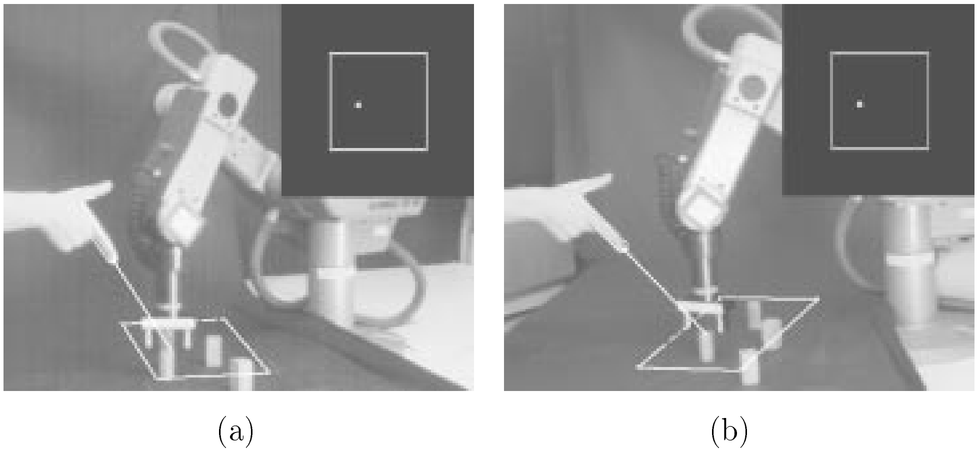


Figure 4.7: Mechanical pointing device used to test the accuracy of the system. We aligned the rod with known points on the workspace, and recorded its coordinates as recovered by the vision system.



(a)

(b)

Figure 4.8: Gestural control of robot position for grasping, seen in stereo.

Small cylinders. For small upright objects on the plane, the grasping operation is trivial and can take place without any further image processing (the grasp configuration being a function only of the target's position in two dimensions). Using visual feedback or under the direct control of the user's gestures (figure 4.8), the robot could be aligned with the target and the grasp executed.

Flat targets. The outer contours of the target's image were localised automatically using a stereo pair of 'expanding' B-spline snakes [21] initialised at the indicated point, enabling both the *position* and *orientation* of the graspable surface to be estimated using affine stereo. They could then be grasped using visual feedback as described in chapter 3.

For successful grasping of more complex objects, it is necessary to incorporate some sort of automatic *grasp planning* based on a stereo *reconstruction* of the target object, to analyse the shapes of its visible surfaces. This is dealt with in chapters 5 and 6.

4.6 Discussion

This algorithm for resolving the direction of pointing proves to be usable and stable in the presence of normal image noise. It does not require camera calibration because all calculations take place in the image and ground planes. By tracking 4 points on the plane it can be made invariant to camera motions.

The system presented here can be extended to situations in which more than one surface can be pointed at; however, this requires an image-based model of those surfaces and is harder to implement with moving cameras (because a large number of world features would have to be tracked to maintain invariance).

The main challenge to this system is the real time tracking of a pointing hand reliably in stereo. At present, this is only possible in an environment where there is a strong contrast between the hand and the background. Tracking is currently implemented on a standard workstation, and could be made more responsive using specialised hardware. Colour vision might also be useful for segmenting the hand in a cluttered scene.

Although subjective pointing direction depends on eye as well as hand position, it is not necessary to model this phenomenon. Instead, by providing the operator with feedback about the *objective* pointing direction (e.g. having a robot follow the pointing hand in real time), objects and locations may be specified for pick-and-place operations. However, in all but the simplest of robotic applications, this will need to be combined with visual reconstruction of objects so that they can be appropriately grasped.

Chapter 5

Uncalibrated Stereo Facet Reconstruction

In this chapter, existing stereo matching techniques are reviewed, and the interaction between matching, reconstruction, and the epipolar geometry is discussed. An algorithm is presented for matching line segments in weakly calibrated stereo and organising them into planar facets for grasp planning.

5.1 Introduction

Our goal in this chapter is to reconstruct, in an image-based frame, the shapes of objects in the robot's workspace. Many robot grippers consist of two parallel jaws, and such a mechanism is well suited to grasping objects with parallel planar surfaces. Thus a useful representation of the object for grasp planning would be a description of the visible planar facets. We shall assume that these facets are bounded by straight edges: thus the problem becomes one of matching edges in stereo views and recovering a description of the position and extent of each facet.

Our stereo vision system is weakly calibrated, meaning that the epipolar geometry and camera parameters are known only to a low level of accuracy, because only a simple self-calibration process has been performed, and the cameras may be subject to disturbances.

A large number of algorithms exist for stereo correspondence: they are summarised in section 5.2. None of these was entirely suitable for our purposes. Correlation and corner-based systems can be used to recover both structure and epipolar geometry, but are unable to match some indoor scenes in which corners are sparse, similar in appearance to one another, or confined to a few planes. Edgel-based systems give excellent results with rectified images, but are sensitive to errors in epipolar geometry because they match along epipolar lines. Line segment matching is more robust in weakly calibrated stereo, but the 3-D reconstruction of lines can be inaccurate unless extra constraints (such as junctions and coplanarity) are taken into account.

A novel system was therefore developed, extending existing algorithms for line segment matching and incorporating image-based coplanarity grouping, to reconstruct scenes composed mainly of straight edges and planar facets.

5.2 Review of stereo matching techniques

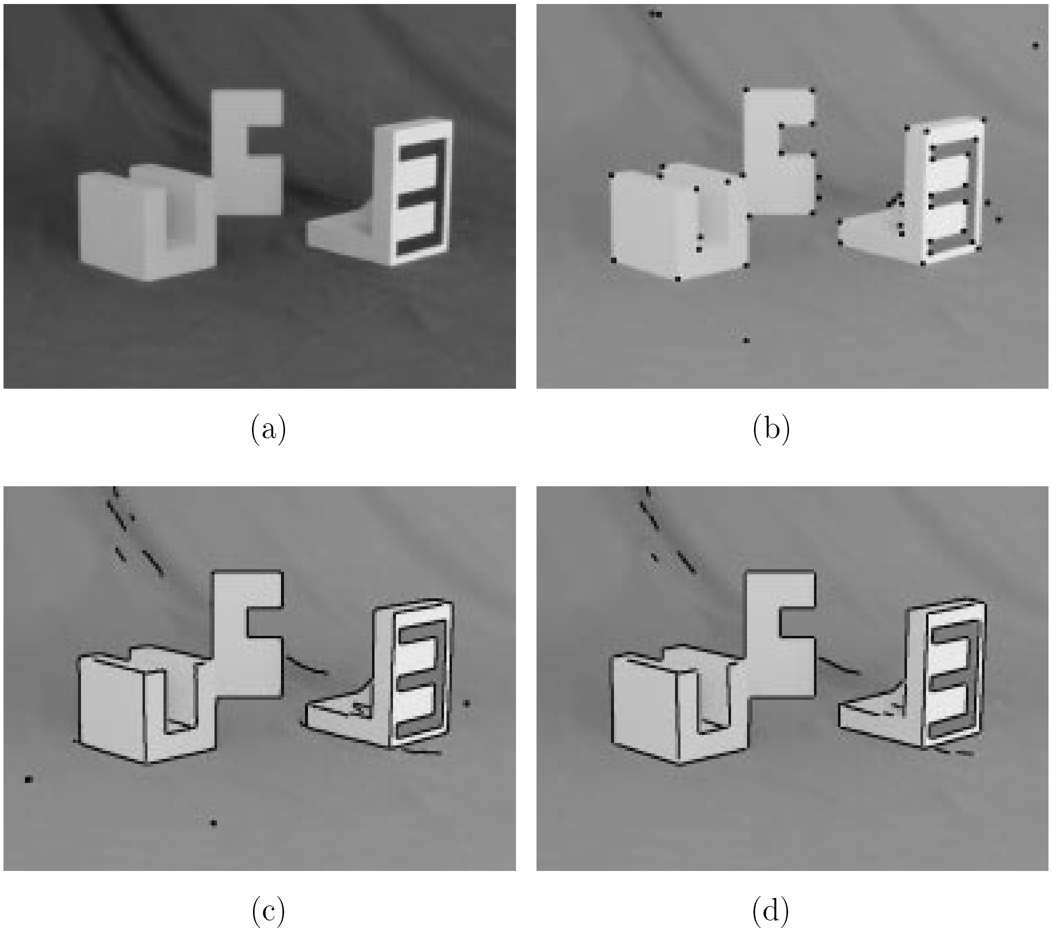
To reconstruct a scene from a stereo pair, it is necessary to find which points in the two views correspond to the same point in space. This is known as the *stereo correspondence problem.* For object reconstruction or recognition, the matched features must then be *grouped* to form surfaces and objects.

5.2.1 Feature extraction

Correspondence algorithms (reviewed in [98]) operate either on individual pixels or general patches of the images, or more commonly on a smaller set of *features* extracted independently in each image (see figure 5.1):

Intensity-based matching. In some cases, pixels on corresponding epipolar lines in the two images can be matched by their intensities [104], but this is easily defeated by noise, reflectance characteristics or differences in the photometric response of the cameras. To overcome noise and camera response differences, the outputs of local filters are considered, such as the smoothed derivative of intensity along the epipolar lines, or the ratio of intensity derivative to intensity [126].

Cross-correlation. Patches between images can be matched by looking for maxima of *normalised cross-correlation* or some other measure of similarity between the views [91]. This assumes that the apparent motion of each patch between views is a translation, i.e. depth variations are small.



(a)

(b)

(c)

(d)

Figure 5.1: Features extracted from an image: (a) original image; (b) corners detected using INRIA corner finder [138]; (c) edges detected using Canny's algorithm [14]; (d) straight line segments fitted to edges.

Edgels. These are points of maximal intensity gradient after smoothing with a filter designed to reject noise [14, 30], or 'zero crossings' after convolving with $\nabla^2\mathbf{G}$ at a given scale [81, 45]. This provides a reasonable distribution of matchable features across the images which are well localised and also geometrically very significant, often coinciding with depth or orientation discontinuities [100, 32].

Corners. These are points around which intensity variation occurs in more than one direction, making them good candidates for matching by cross-correlation as well as points of likely geometrical significance. They can be detected using ratios of first and second-order differential operators to find edge-like points having maximal curvature [94, 132], or points of maximal auto-correlation after Gaussian smoothing [57]. By matching only the corners, the complexity of the correspondence problem is greatly reduced; the rest of the scene can be reconstructed by interpolating between corners using triangulation [55].

Line and curve segments. Edgels generally exhibit continuity and are grouped into *chains* in each image. These chains can then be segmented into straight line segments [6] or parametric curves such as B-splines [47, 17], and entire segments matched between images, reducing the computational complexity. In the presence of noise and quantisation errors, lines and parametric curves can be localised to higher precision than individual pixels. However, the segmentation of chains is not always stable with respect to viewpoint changes, so the matches between images are not always 1-to-1 [84].

Higher-level features and groupings. In some environments in which the forms of objects are modelled it is possible to apply 'perceptual grouping' to features, organising them into higher-level structures by the detection of symmetry, parallelism, clustering or similarity within the image [80, 74]. By matching entire groupings, the search space for correspondence is reduced. Systems have been proposed which match a hierarchy of features, using both bottom-up monocular grouping and top-down stereo matching [87, 20].

5.2.2 Matching constraints

Most stereo systems exploit the *epipolar constraint* which restricts the search for matching features to a one-dimensional one. Often they require the images to be rectified. Limits may be imposed on the magnitude of the horizontal *disparity* between views, effectively bounding the depth range of reconstructed features.

A pair of features lying within the allowed window of disparity are considered a candidate match if they are sufficiently similar in appearance (e.g. for edgels, if their orientation difference is within a certain range). However, ambiguities often arise, which can be resolved only after considering the interactions between candidate matches. Constraints used to disambiguate matches include:

Uniqueness. A point in the world has only one 3-D position at a time. Therefore a feature in one image can match at most one feature in the other [82]. This constraint can be broken when matching group features such as curve segments which may be organised differently in each image [84, 6].

Ordering. The order of matching features along the epipolar lines will usually be the same in both images [7]. This constraint is occasionally broken at the *occluding edges* of slender objects, or where there is transparency [98, 60], but is obeyed by most images of opaque solid objects.

Surface shape. Constraints can be imposed on the shape of reconstructed or interpolated surfaces, to aid matching. The simplest of these are *smoothness* constraints, which assume local planar structure [60, 83]; and limits on the *disparity gradient*, to favour a continuous variation of depth [100]. These constraints cannot be applied at occlusion boundaries in the images.

Continuity. When matching edgels, it can be assumed that edgels that are continuous in the image also connect in space, and that the disparity of an edge will change smoothly along its length as it crosses the epipolar lines [98, 95]. Similarly line or curve segments which meet at a point in one image are likely to correspond to segments which meet in the other [101].

5.2.3 Matching algorithms

Many algorithms have been proposed for binary matching for computer vision and other applications. The problem in general is to find the subset of the candidate matches (of which there are up to n^2 where n is the number of features) which give an optimal correspondence between images, subject to given matching constraints. To a large extent, it is the form of those constraints which determine the algorithm used and its complexity.

With only the uniqueness constraint, a procedure such as the 'stable marriage' algorithm of Knuth et al. [66] can recover the optimal set of matches. This iterates through the features of one image, enumerating the candidate matches for that feature and choosing the one with greatest strength, whose feature in the other image is not already associated with a stronger match. A record is kept of the best match found so far for each feature. Complexity is $\mathcal{O}(n^2)$ in the number of features. Where additional mutual exclusivities must be imposed, a further depth of iteration is required to find all the matches, and complexity rises to $\mathcal{O}(n^3)$.

For edgel matching, correspondence can be formulated as a dynamic programming problem in which a 'path' must be found across each epipolar plane obeying the ordering and uniqueness constraints whilst seeking to minimise the disparity gradient, visiting each accepted match in left-to-right order [7]. Complexity is $\mathcal{O}(n^3)$ in the number of edgels on each epipolar line. Results can be enhanced by using connectivity information from the neighbouring epipolar planes [95], but this greatly increases computational complexity.

Correlation-based region matching can be made more efficient by imposing surface shape smoothness constraints and by the use of multi-scale algorithms that estimate disparities at successively finer resolutions [93]. Such an approach has also been applied to edge matching using a bank of $\nabla^2\mathbf{G}$ filters of different sizes [44].

In general, stereo matching of discrete features can be posed as a *cooperative* problem, with matching constraints taking the form of mutual positive or negative support, and/or mutual exclusivities between sets of two or more matches. Such a problem can be solved by a *relaxation* algorithm [82, 100, 84, 20]. Essentially, this iteratively updates the support given to each candidate match so as to minimise an objective function encoding the matching constraints, until all matches have been selected or rejected. An algorithm of this class will be used in section 5.4 in the design of our weakly-calibrated matching system.

The general form of a stereo vision system is summarised in figure 5.2.

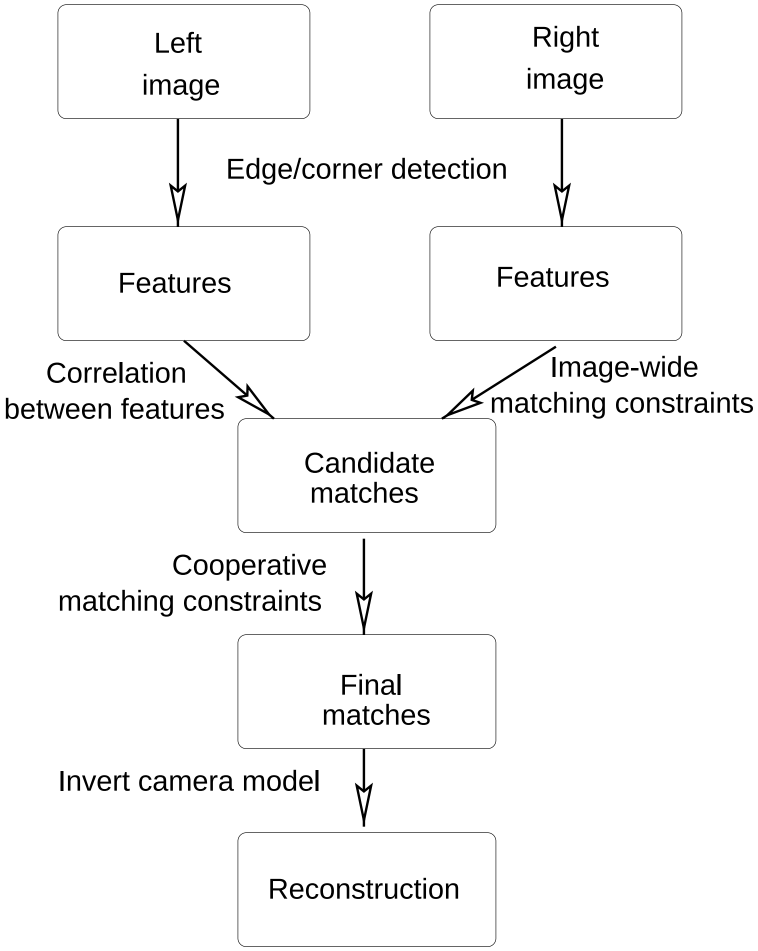


Figure 5.2: Overview of the steps and data representations of a typical feature-based stereo matching algorithm.

5.3 Uncalibrated stereo considerations

The behaviour of stereo matching systems with *uncalibrated* cameras is now discussed. Corner-based matching can be used to recover both scene structure and epipolar geometry, provided there is a sufficient density of correct matches. Edge-based matching is denser and more robust, but *cannot* be used to update the epipolar geometry directly. Hence reconstruction is sensitive to rectification errors, and uncalibrated reconstruction is not generally possible. Coplanarity constraints are one way to resolve this problem, and groups of coplanar features can be identified in uncalibrated stereo and reconstructed up to an affinity.

5.3.1 Point features

Much recent work on reconstruction without a prior epipolar constraint has relied on point features such as corners. These can be matched in stereo [138] and successfully tracked over long sequences of images in structure-from-motion [10], especially in natural scenes which tend to be rich in non-repeating texture. Robust statistical methods such as RANSAC [38] enable the fundamental matrix to be estimated even when there is a proportion of false matches [138, 127, 70]. The epipolar constraint enables most of the erroneous matches to be rejected, and the scene reconstructed up to a projectivity or affinity (as in chapter 2).

Figure 5.3 shows the results of the uncalibrated corner matching algorithm of Zhang et al. [138] on some indoor test scenes. The system uses cross-correlation between views to match corners, and a relaxation algorithm to enforce the uniqueness constraint; a fundamental matrix is then fitted to the correspondences using a robust estimator (Least Median of Squares [110]), and the matching process repeated using the recovered epipolar constraint.

This algorithm works well on highly textured or heterogeneous scenes (such as the `lab` images in figure 5.3c) where many corners can be localised and matched by cross-correlation, correctly recovering both the epipolar geometry and a dense set of correspondences. However, with simpler images (such as figure 5.3d), the density of correctly matched points is lower. The ‘corners’ detected in the images do not always coincide with polyhedral corners. Thus in the absence of texture, corners alone may not give a sufficiently detailed reconstruction for surface modelling, e.g. for robot grasp planning.

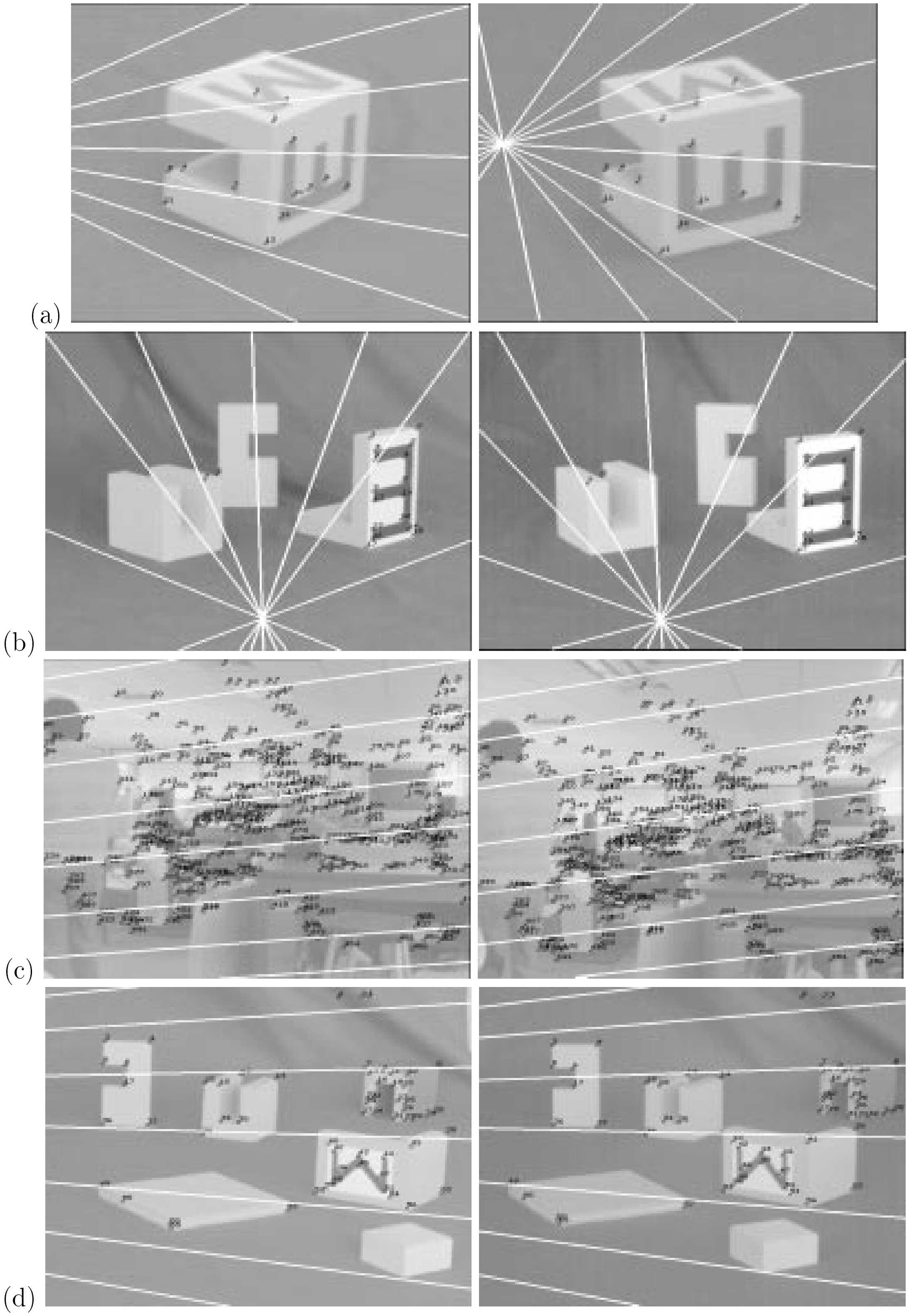


Figure 5.3: Results of corner matching [138] and estimated epipolar lines: (a) `cube` scene; (b) `test` scene; (c) `lab` scene; (d) `blocks` scene. Sparsity of corners and high incidence of coplanarity leads to incorrect solutions in (a) and (b).

On some images, such as the `test` and `cube` pairs, uncalibrated corner matching fails entirely: the corners look too similar to be distinguished by correlation, and the predominance of planar facets proves to be a hindrance rather than a help: large coplanar subsets of points can defeat the robust estimator and lead to a degenerate solution for the fundamental matrix (figure 5.3 a, b).[1]

5.3.2 Edge-based features

We shall now consider the feature type most prominent in many indoor scenes, *edges*. Because they are localised in only one dimension, matching of edge elements depends upon prior estimation of the epipolar constraint, and edgels tangent to the epipolar lines cannot be matched uniquely [100]. This is the *aperture problem* in stereo [131]. But by grouping the edgels into line or curve segments, complexity is reduced and the epipolar constraint can be relaxed to require that matching segments *overlap* when projected into the rectified vertical axis (that is, their ranges of y and y' values intersect), the degree of overlap indicating how well aligned the segments are in the two views [6, 137]. This allows some segments to be matched with only approximate epipolar calibration.

Ayache [6] presents an algorithm for matching line segments in rectified stereo pairs, in which matches give support to each other if they are nearby (according to a coordinate bucketing scheme) in both views. Hypothetical matches are formed between line segments if the y coordinate of the midpoint in one image falls within the range of y' values of the other (and *vice versa*), and if their lengths and orientations are similar within limits inferred from the camera geometry [4]. Matches are accepted if they give rise to maximal cliques of supporting matches under the uniqueness constraint. Reconstruction is performed under the assumption that the cameras are accurately calibrated.

Zhang [137] has used a numerical search method to solve for the camera motion (hence the epipolar constraint), given two intrinsically-calibrated images of matched line segments, by seeking to maximise the total epipolar overlap, thus permitting *uncalibrated* reconstruction. However, the system is not well constrained, and a very large number of segments is required for a satisfactory solution.

[1] One way to avoid this problem would be to search for sets of points consistent with a degenerate model (e.g. planes), then solve for the epipolar constraint using whatever features (if any) remain. This is essentially the approach taken by the PLUNDER motion segmentation algorithm [127].

5.3.3 The problem with vertical disparity

Consider the case when the epipolar constraint is inaccurately modelled. Corresponding features will exhibit misalignment or *vertical disparity*.[2] This vertical disparity will affect the 3-D reconstruction of any one-dimensional features such as edgel chains, lines and curves, which depend on the epipolar geometry to establish matching points and recover depth. What is more, the error varies with orientation: each unit of offset induces a horizontal disparity error of $\tan\theta$, where θ is the angle to the vertical (horizontal lines cannot be reconstructed at all). Figure 5.4 shows the results of this error, and its disruptive effect on plane reconstruction.

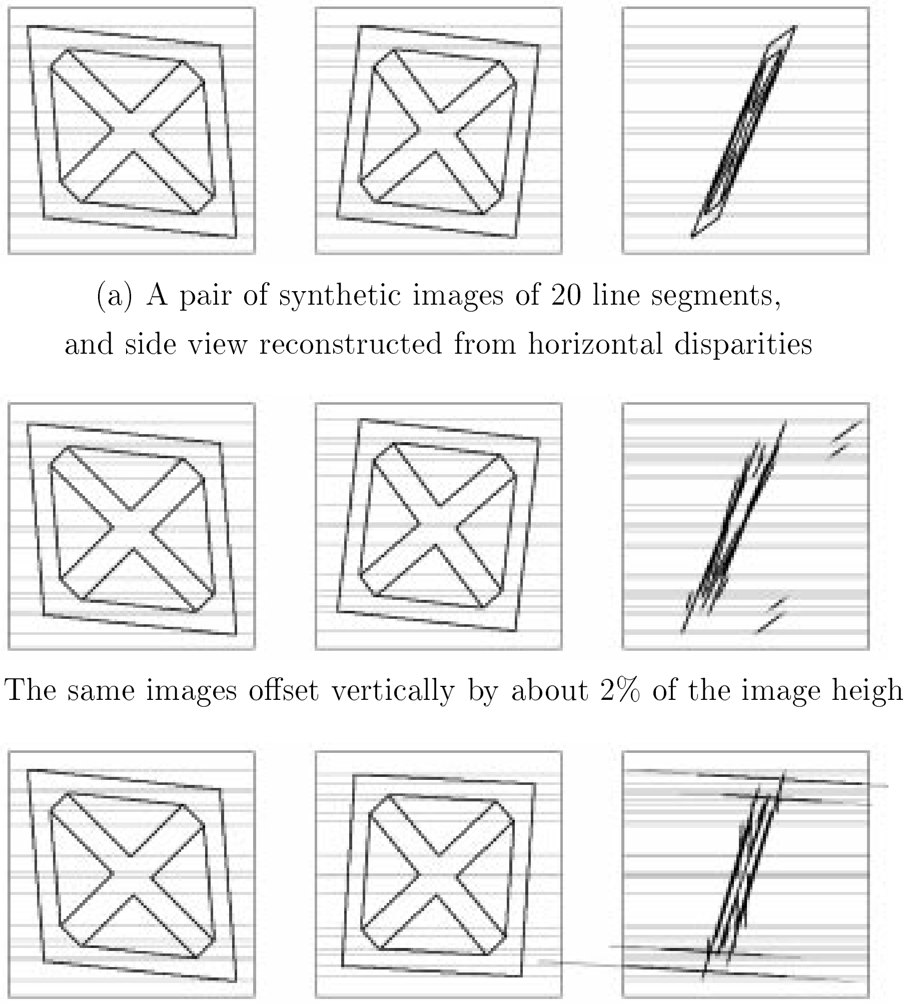
The vertical disparity problem can be overcome if line or curve segments are known to have matching endpoints: one can simply solve for the 3-D coordinates of the endpoints in some approximate (affine) world frame, or use them to re-estimate the fundamental matrix. This is not always the case in real images, due to the fragility of line fitting and segmentation (in the case of curve segments, it may be possible to match other distinguished points such as bitangencies for plane curves, or tangencies to the epipolar lines [102, 5]). However, where line segments meet at a junction in space — and more generally where they are coplanar but not parallel — their *intersections* in two images can be used as accurate point correspondences; this is exploited in section 5.5.

5.3.4 Coplanarity grouping of line segments

For the reconstruction of planar surfaces, it is necessary to group matching line segments into planar facets. Plane grouping could take place between matching and reconstruction; it could also be performed concurrently with the matching process, to favour candidate matches which belong to well-supported planar facets.

Coplanarity of line segments cannot be detected unambiguously in a single image, but requires stereo. Lines that are parallel in two weak perspective images are necessarily coplanar in space, though they might not be physically connected. Line segments which meet at a junction in both images are generally coplanar, unless the junction has been caused by occlusion (a 'broken T-junction').

[2] The terms *horizontal*, *vertical* and *disparity* are used here to mean the x, y and $(x' - x)$ quantities in a rectified coordinate system. This does not imply that both images must be rectified, merely that some estimate of the epipolar constraint is available by which rectified coordinates could be calculated.



(a) A pair of synthetic images of 20 line segments, and side view reconstructed from horizontal disparities

(b) The same images offset vertically by about 2% of the image height

(c) One of the images has been rotated by 3°, causing non-uniform vertical disparity

Figure 5.4: The effect of vertical disparity on edgel-chain, line or curve segment reconstruction in rectified stereo. Because the error induces a horizontal disparity that varies with orientation, the planar structure of the scene is destroyed.

With known epipolar constraint

Coplanarity can be tested using the following theorem: **Two non-parallel lines are coplanar if and only if their intersections in two images lie on corresponding epipolar lines.**[3] Whereas if they are not coplanar, there will be a vertical offset between their apparent intersections in the two images (in fact, this offset or *pseudo-disparity* is a cue to the depth difference of the segments [78]). This test is sensitive to errors in the epipolar geometry, though appropriate limits can be placed on the amount of vertical disparity allowed. Its resolution is therefore degraded as epipolar constraint uncertainty increases.

With unknown epipolar constraint

Even without camera calibration or an epipolar constraint, a pair of images can be segmented into planar regions by the following theorem: **Two views of a planar surface are related by a two-dimensional projective transformation; features are consistent with this transformation if and only if they lie on the plane**. In weak perspective, the transformation will be affine (see chapter 2).

Faugeras and Lustman [34] use this theorem to recover correspondence of line segments on a single plane, by hypothesising a set of matches that define a collineation between views, and testing for consensus with other segments. A Kalman filter is used to refine the estimate of the transformation. They show that two views of the plane allow structure and motion to be computed up to scale and a two-fold ambiguity (for intrinsically calibrated cameras). The ambiguity can be resolved using a second plane or a third view. The principle is not tested on more complex scenes.

Sinclair and Blake [118] use 2-D projective invariants to detect sets of 5 or more coplanar points (given corner correspondences) and construct an approximate piecewise-planar model of terrain with application to mobile robot navigation.

A drawback of this method is that it requires a minimum of four (or three) lines or points to define the projective (or affine) transformation, and at least one other feature to verify it. Hamid et al. [52] show that, for a typical stereo camera geometry, feature localisation must be to sub-pixel accuracy to segment nearby planes reliably using such an algorithm alone.

[3] Outline proof: For the lines to intersect in both views at corresponding epipolar lines, each line must exhibit the same disparity where it crosses this epipolar plane, therefore they must be at the same depth. This is degenerate if one or both lines are horizontal.

5.4 An algorithm for uncalibrated matching

This section describes a stereo matching algorithm which operates on line segments. It is based upon existing techniques, but has been designed to deal explicitly with the bounded uncertainty in epipolar line correspondence found in weakly calibrated stereo. Figural relations between segments are analysed to add robustness.

5.4.1 Feature extraction

It was found that corners and correlation-based algorithms do not always give the required density and resolution of reconstruction for the recovery of facets. Edgel-based matching along epipolar lines was also rejected because of the problem of epipolar misalignment, which could disrupt planar surfaces. Edge segment matching was therefore chosen. For simplicity only straight edges are considered here; though much of our approach could be extended to curve segments.

Line segments were extracted from Canny edgel data [14] using a recursive algorithm that searches for straight segments of edgel chains (figure 5.1(d), p73). Each segment is represented by its endpoint coordinates; and associated with it are uncertainty measures for its orientation and normal offset, obtained from the residual errors after fitting it to the edgels by orthogonal least squares [122].

5.4.2 Monocular relations

Consider figural relations between just two line segments in a single image (figure 5.5). It is assumed that the edges are extracted in such a way that segments do not cross. Notable binary relations include:

Parallelism. Parallelism between segments (within some given margin of error) can be determined very quickly using an orientation bucketing scheme. It is obviously not meaningful to look for the intersection of parallel lines.

Collinearity. This is a special case of parallelism. Collinearity is used to extend the uniqueness constraint to line segments: because of the fragility of line segmentation, one or more collinear segments in the first image may match one or more collinear segments in the second.

Junction. This occurs when an endpoint of one segment in an image lies within some maximum distance of an endpoint of another segment, suggesting that the edges are coincident in space.

Collinear junction. This occurs when segments are collinear and meet at a junction. Such junctions are not generally stable between views; they could also be a component of a 'broken T-junction' between occluding and occluded segments.

T-junction. This occurs when the endpoint of one segment lies close to another segment. It suggests an occlusion boundary.

The system identifies these relations between segment pairs in each of the two images. The threshold angle for parallelism was set to 3° greater than the given orientation uncertainty, to detect parallel lines in the presence of optical distortion or mild perspective effects. For junction detection, endpoints were required to be within 6 pel (or up to 12 pel for longer or more uncertain segments) of a point extended 3 pel out from the other segment's endpoint. These limits were chosen to defeat the observed 'corner-rounding' behaviour of the edge detection and line fitting code, caused by the isotropic smoothing stage of Canny's algorithm.

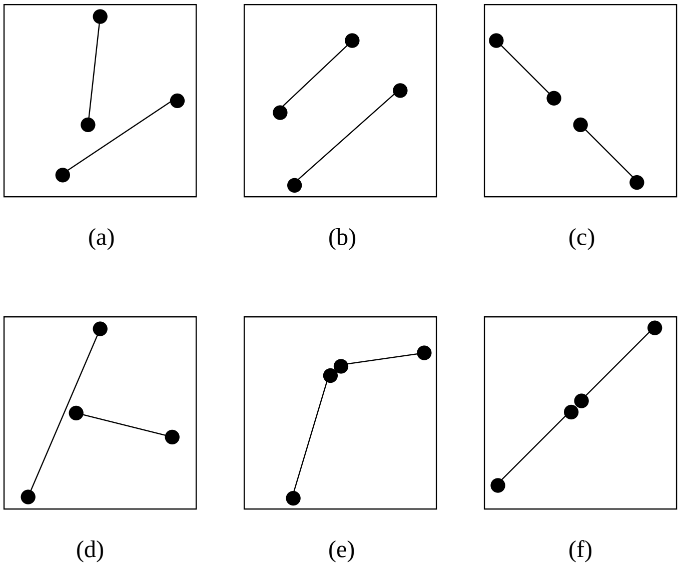


Figure 5.5: Binary figural relations between line segments in a single image: (a) general case (b) parallel (c) collinear (d) T-junction (e) junction (f) collinear junction.

The search for related segments could be accelerated considerably by sorting them into 'buckets' [65] of segments whose orientations or endpoint coordinates lie within particular intervals: for a typical image of 200 segments with 2000 related pairs, execution time is approximately 0.25 seconds.

5.4.3 Candidate matches

The next stage in the algorithm is the enumeration of 'candidate matches,' or segment pairings between images that could possibly be images of the same edge in the world. To generate these, an approximate epipolar constraint is required — this can be a linear estimate, provided by just four reference points[4]. The matching criteria are based on those of Medioni and Nevatia [84] and Ayache [6], with some adaptations for weakly calibrated stereo:

- **Epipolar overlap** as a fraction of total vertical extent (min. 25%),
- **Length ratio** in a 'vertically stretched' rectified frame (max. 3),
- **Orientation difference** in the rectified frame (max. 60°),
- **Disparity limits** extrapolated from the disparities of the reference points.

The constant values specified in the criteria were chosen by hand to optimise performance, though the system is not critically sensitive to any of them.

Notes:

1. The epipolar overlap criterion is broadened somewhat by allowing up to 16 pel of vertical offset, to overcome rectification errors. This allows many nearly-aligned segments to be matched (including horizontal ones) even when they show no overlap at all.

2. Length and orientation comparisons are performed in a 'vertically stretched' rectified coordinate system. This gives more weight to the direction normal to the estimated epipolar lines which should be invariant to viewpoint changes; but also takes into account the component along the epipolar lines, to allow near-horizontal features to be compared and matched. For typical stereo camera configurations, length and orientation differences will be within the above bounds for all except the most foreshortened of matching segments.

[4] If no reference points are available, it is assumed that the epipolar lines are approximately horizontal and the range of permitted disparities is ±100 pel.

Angle threshold and orientation-based support:

$$a_{ij} = \begin{cases} 2(\cos\theta_{ij} - 0.5) & \text{if } \cos\theta_{ij} > 0.5, \\ 0 & \text{otherwise.} \end{cases}$$

Length-ratio threshold and support:

$$b_{ij} = \begin{cases} 1.5(L_i/L'_j - 0.33) & \text{if } L_i < L'_j < 3L_i, \\ 1.5(L'_j/L_i - 0.33) & \text{if } L'_j < L_i < 3L'_j, \\ 0 & \text{otherwise.} \end{cases}$$

Epipolar overlap constraint and support:

$$\lambda_{ij} = \frac{\textsc{overlap}(y_{i[0]}..y_{i[1]}; y'_{j[0]}..y'_{j[1]}) + 16.0}{\textsc{max}(y_{i[0]}, y_{i[1]}, y'_{j[0]}, y'_{j[1]}) - \textsc{min}(y_{i[0]}, y_{i[1]}, y'_{j[0]}, y'_{j[1]})}$$

$$c_{ij} = \begin{cases} \frac{4}{3}(\lambda_{ij} - 0.25) & \text{if } 0.25 < \lambda_{ij} < 1, \\ 1 & \text{if } \lambda_{ij} > 1, \\ 0 & \text{otherwise.} \end{cases}$$

Midpoint disparity limits:

$$\delta_{ij} = \tfrac{1}{2}(x'_{j[0]} + x'_{j[1]} - x_{i[0]} - x_{i[1]})$$

$$d_{ij} = \begin{cases} 1 & \text{if } \delta_{MIN} < \delta_{ij} < \delta_{MAX} \\ 0 & \text{otherwise.} \end{cases}$$

Overall intrinsic support of the candidate match:

$$\sigma_{ij} = a_{ij}\, b_{ij}\, c_{ij}\, d_{ij}$$

Table 5.1: Algebraic summary of the constraints for a candidate match between segments. i and j are segment numbers in the first and second image respectively. Angles and lengths are in a 'stretched' rectified frame in which the vertical component has double weight. The pair is considered a candidate match if $\sigma_{ij} > 0$.

3. Matches are only permitted on segments having the same sense in each image, i.e. originating from edges having the same sign of gradient. Practically all correct matches will meet this criterion (except occasionally at an occlusion boundary where there is background intensity variation), whilst a further 50% of false matches are rejected [6].

Again, bucketing is used to speed up the search, indexing the segments which intersect a number of epipolar bands. Each test assigns a number to the candidate match, indicating by what margin the criterion is met; these are multiplied to yield the *intrinsic support* of the match (see table 5.1).

In tests with 8 stereo pairs of blocks and laboratory scenes, between 5%–20% of the line segments in each image could not be matched according to these criteria, and about 50% had more than one candidate match. The average number of candidate matches per line segment varied between 1.5 and 3.0.

5.4.4 Inter-match constraints

In our scheme, matching constraints are expressed by two types of relations between candidate matches: mutual exclusivities between pairs of matches (hereinafter dubbed RIVALS), and mutual positive support between matches (FRIENDS). The ordering constraint is extended for weakly calibrated stereo, and a novel connectivity constraint is introduced, based on junction relations common to both images.

Uniqueness constraint. Matches that share a segment in either image are RIVALS, unless they are collinear and connected in the other image (as this could be due to fragmentation of the edgel data).

Ordering constraint. For approximately rectified stereo views, we extend the ordering constraint to matches that 'cross over' in a vertical as well as a horizontal sense (figure 5.6). 'Epipolar ordering' is tested for segments that intersect one another's projection onto the (rectified) vertical axis in both views; 'General ordering' is violated by segments that cross one another during a linear warp between views (assuming matching endpoints), which was found by human inspection to be an important cue for spotting inconsistent matches. Out-of-order matches are RIVALS.

Collinearity. Matches that are collinear in *both* images are FRIENDS (since this does not generally happen by accident, it suggests they are correct matches and derive from collinear features in the world).

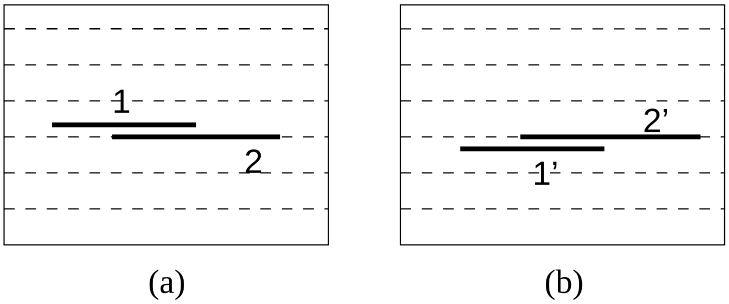


Figure 5.6: Ordering constraint violation by segments with vertical disparity: matches 1 and 2 are out of order — they 'cross over' between views (a) and (b), even though they are ordered in the direction of the estimated epipolar lines.

Connectivity. Matches are FRIENDS if they have *common* junction or T-junction relations in both images; but RIVALS if they have *incompatible* junction relations (i.e. they meet at different endpoints, implying a large relative motion between views).

5.4.5 Constraint propagation

Our goal is to try to find a set of matches with maximal total support, consistent with all the matching constraints. The solution should favour pairs matches which are FRIENDS of other chosen matches, but not include any pairs of RIVALS.

This is achieved by an iterative algorithm that propagates FRIEND and RIVAL constraints between matches. To start with, a proportion of each candidate match's intrinsic support is added to the support of each of its FRIENDS.

At each iteration, we enumerate the 'winning' matches having greater support than any of their RIVALS[5]. A proportion of the *least ambiguous* winners is then selected — these are the winners with the greatest support difference over their nearest rival. These matches are considered correct: their FRIENDS receive extra support, and their RIVALS are eliminated. As matches are eliminated, they withdraw the support that they earlier gave to their FRIENDS. The process repeats until all matches have been either promoted or eliminated.

[5] The search is made efficient by use of the uniqueness constraint which partitions the matches into small sets of RIVALS (where there are multiple collinear matches, for simplicity only one is promoted at each iteration).

This is a *'some-winners-take-all'* algorithm (similar to that described by Zhang et al. [138] in the context of corner matching; cf. [100] for a related strategy for edgel matching). In our implementation, only $\frac{1}{4}$ of the winning matches are promoted on the first iteration, as this helps to prevent early convergence to a local minimum; thereafter this rises to $\frac{2}{3}$. As with other relaxation algorithms, an optimal solution is not guaranteed, though the final set of matches must be consistent and locally optimal. Convergence is assured so long as at least one winner is promoted each time, and generally occurs after 4–10 iterations. With the use of suitable data structures, already-matched segments can be skipped, making each iteration faster than the previous one.

5.4.6 Epipolar constraint re-estimation

If the initial estimate of the epipolar geometry was inaccurate (or missing, and assumed horizontal), it is useful to recompute the epipolar constraint and repeat the matching process. We use the *junction* relation between segments to find intersection points that correspond between views. There are very few outliers amongst such points, but some of the junctions may be false intersections generated by occlusion ('broken T-junctions'); therefore, an iterative re-weighting scheme is used to reject those furthest from their supposed epipolar lines.[6] The parameters of the epipolar constraint (in the form of equation 2.9) are estimated using linear least squares.

Often the number and accuracy of correspondences is improved by repeating the matching process using the newly recovered epipolar constraint and disparity limits. It should be noted that the above matching algorithm can function even with some epipolar mismatch, and there is never any benefit to repeating the process more than twice.

5.4.7 Results

Figure 5.7 shows how the algorithm matches line segments on a pair of simple '`cube`' images by propagating qualified uniqueness, epipolar ordering, general ordering and connectivity constraints. The cameras were arranged by hand to be roughly rectified, but no calibration was performed. 39 and 40 line segments were detected in each image. Correspondence was solved for 35 segments in four iterations.

[6] The weight for each point match is proportional to $16 - |y' - y|$ and falls to zero for junctions with vertical offsets above 16. Just 4 junctions within this range are required for the method to be successful.

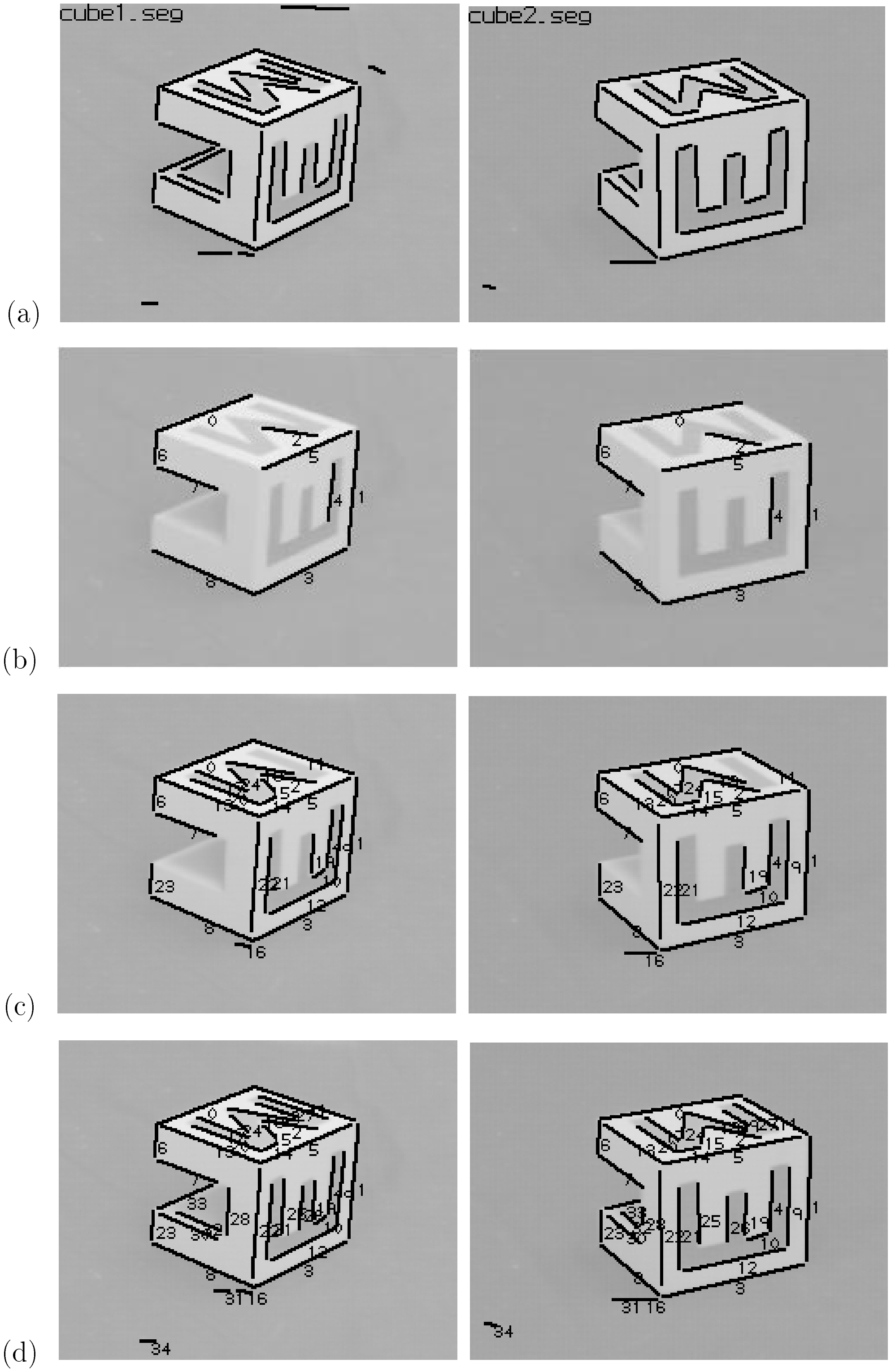


Figure 5.7: Matching by constraint propagation: (a) line segments from the `cube` images; (b) 'least ambiguous' segments matched on the first iteration; (c) segments matched after 2 iterations; (d) final result reached after 4 iterations. Uncalibrated views — epipolar lines roughly horizontal.

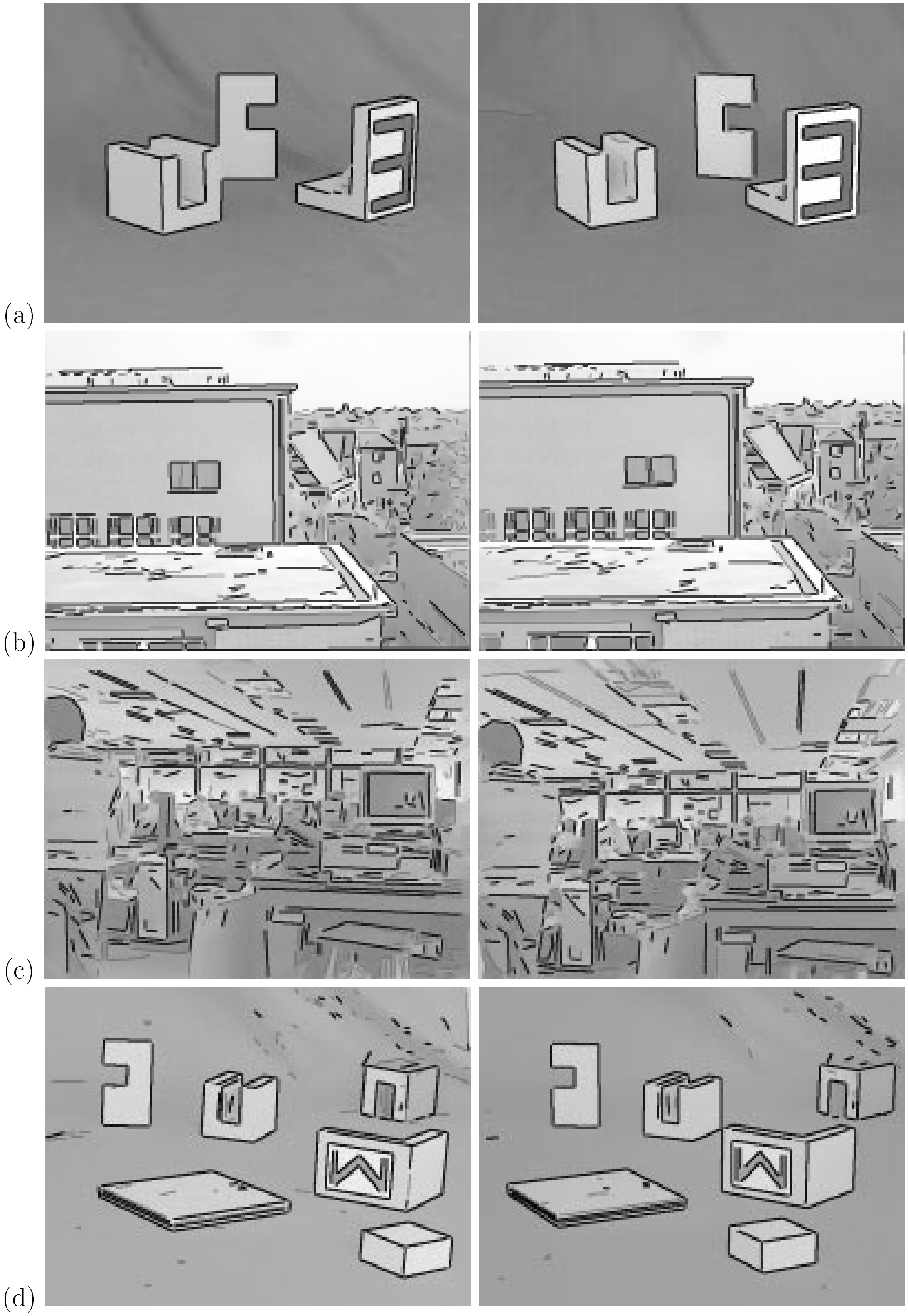


Figure 5.8: Matched (black) and unmatched (grey) segments from 4 stereo pairs: (a) `test` scene (no calibration); (b) `roof` scene (given 4 point correspondences); (c) `lab` scene (rectified using corner matching); (d) `blocks` scene (no calibration).

Figure 5.8 shows matching output for 4 other stereo pairs. The matcher was run twice on the `test` and `blocks` images in order to recover the appropriate epipolar geometry and disparity range (results of second pass shown). For the `roof` images, these were obtained by hand-matching four points, and for the `lab` scene, using the INRIA corner matching system; ground truth calibration data were not available. In all cases a linear epipolar constraint is used. Results are good, considering that no surface shape constraint has yet been imposed; however there are a number of false correspondences — typically unconnected short segments without a true match which become associated with one another more or less randomly.

Reconstruction is not attempted at this stage.

5.4.8 Complexity analysis

Feature extraction. Line segment extraction is based upon edge detection, which is quite slow on general-purpose hardware due to the convolution stage. This takes a constant time for a given image area, and is independent of the number of features. For simple 'blocks world' scenes, edge detection is the slowest part of the entire stereo algorithm (taking about 10 seconds per image).

Monocular relations. Although bucketing is used to reduce the complexity in simple scenes, in clutter the search for related pairs reverts to $\mathcal{O}(n^2)$. Let the number of relations per line segment be r (usually much less than n).

Enumeration of candidate matches. As above, this has complexity $\mathcal{O}(n^2)$. Let the number of candidate matches be nm.

Enumeration of FRIENDS**.** This involves enumerating the pairs of candidate matches for each pair of segments related in one image. Complexity is therefore $\mathcal{O}(nrm^2)$ which in clutter tends to $\mathcal{O}(n^3)$.

Constraint propagation. The uniqueness constraint makes this efficient,[7] since each iteration need only traverse the list of candidate matches once ($\mathcal{O}(nm)$) to find the subset of at most n with more support than any uniqueness-RIVAL. Each of these must be compared with $\mathcal{O}(n)$ other promoted matches to test for RIVALRY, and the number of iterations is itself bounded above by n. In the

[7] Even where multiple collinear matches abound, their number is implicitly limited by the criteria for a candidate match, which each must meet.

absence of FRIENDS, overall complexity is between $\mathcal{O}(n^2)$ and $\mathcal{O}(n^3)$. With f FRIENDS per match, time is required to adjust the support of the FRIENDS of matches which are destroyed, with complexity $\mathcal{O}(nmf)$.

It is noted that the complexity depends not only on the number of line segments in the images but also on the number of *junctions* and other relations between segments (which also affects the number of FRIENDS). Where this is small, overall complexity is $\mathcal{O}(n^3)$ but in pathological images (such as when all line segments radiate from a point), r tends to n, f tends to nm, and complexity approaches $\mathcal{O}(n^4)$.

5.5 Coplanarity grouping

We now consider the integration of plane grouping into the stereo system, using the paradigm of [34, 118] and others. Coplanar sets of segments are identified by consensus with an affine transformation between views. In uncalibrated stereo it is important that coplanarities be detected before full 3-D reconstruction is attempted, so as to reduce the disparity errors caused by epipolar misalignment. Rather than occurring after correspondence, plane hypothesis and grouping are incorporated into the cooperative matching stage and introduce a shape constraint (see table 5.2, p98).

5.5.1 Plane hypothesis formation

A plane hypothesis is an affine transformation between views, defined by a set of three (or four, if there is parallelism) matching segments called a *seed*. It would be computationally expensive to enumerate all the triplets of candidate matches, so seeds are generated only from promoted matches, and heuristics employed to choose the seeds most likely to lie on planar facets.

Three forms of seed were considered:

- **Triangle form** (figure 5.9a). This consists of three matching segments, whose intersections in two images lie close to corresponding epipolar lines (i.e. they are pairwise coplanar, allowing for epipolar misalignment), and whose direction vectors in $(u, v, \mathit{disparity})$ space form a triple product close to zero (i.e. their directions in 3-D are approximately coplanar). To reduce the search space, we require that two of the intersections be junctions in both images. A triangle gives a minimal definition of an affine transformation between views.

- **Parallel form** (figure 5.9b). Because many facets are rectangular, a second form of seed was allowed, in which two segments are parallel, and may even be horizontal, and are joined by a third segment so that the intersections in two images lie close to corresponding epipolar lines. The lines are likely to be coplanar but do not define a unique affine transform. Therefore an endpoint or junction with a *fourth* segment (which need not itself be coplanar) is required to fully define the transformation.

- **Parallelogram form** (figure 5.9c). Parallelograms in the images are often produced by rectangular facets in the world, but due to edge fragmentation these do not always yield seeds of the above form. Therefore a variation of the parallel seed was introduced, formed from two pairs of parallel segments which meet at two opposite junctions and define more than half of the perimeter of a parallelogram. The junctions and one of the other intersections are used to define the affine transformation.

Plane hypotheses are systematically enumerated out of the promoted matches as these emerge from the correspondence process; each is then tested for consensus with previously promoted matches as well as with candidate matches.

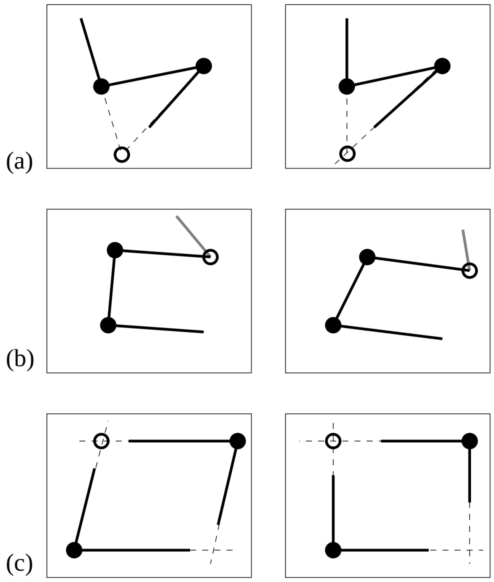


Figure 5.9: Forms for plane seeds, consisting of 3 or 4 matching segments: (a) triangle form; (b) parallel form; (c) parallelogram. In each case, two intersections (blobs) and an additional endpoint or intersection (circle) provide a minimal basis for the affine transformation.

5.5.2 Hypothesis testing and coplanar segment support

To test for coplanarity, promoted or candidate matches are tested for consensus with the affine transformation defined by the seed (figure 5.10). The test consists of 'transferring' the line defined by the segment from one view into the image coordinate frame of the other, and comparing it with the matching segment. The test is performed symmetrically in both directions between the views. No assumption is made about endpoint correspondences,[8] but the segments are required to 'overlap' on the plane by at least 33% of each segment's length.

Although Canny's algorithm can detect step edges to sub-pixel accuracy [14], it can produce correlated errors which do not manifest themselves in the residual errors after line fitting. Rather than propagating the residual errors, a simple threshold of 2.0 pel normal offset at the endpoints was chosen (cf. [52]). Candidate matches consistent with one or more hypotheses received extra support in the matching process (table 5.2).

Complexity

The number of plane hypotheses is proportional to the number of triples of connected line segments, which is approximately nr^2. Let there be h plane hypotheses. The testing of promoted and candidate matches for consensus with each plane hypothesis increases the computational complexity of the matching/grouping algorithm by $\mathcal{O}(nmh)$. In most scenes, h/n is small and complexity varies $\mathcal{O}(n^3)$, so the speed of the algorithm is not significantly reduced.

Results

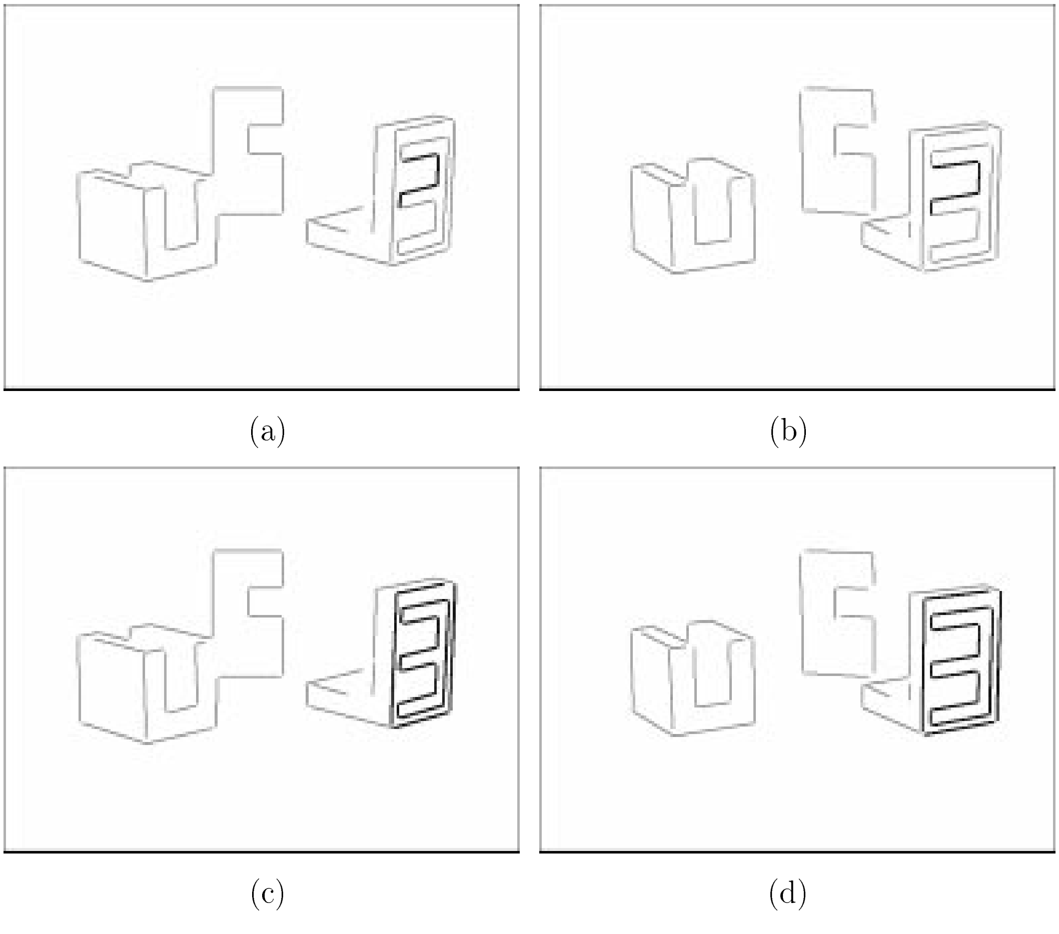
The above forms of plane seed were successful at identifying most of the planes in the test images without generating many false hypotheses, although planes without three connected edges were missed. Many of the facets produced several hypotheses, but these did not always return the same consensus set; and occasionally a segment was wrongly grouped as belonging to two or more conflicting hypotheses. Such problems are inherent in any threshold-based test, in which noisy inliers cannot be perfectly separated from nearby outliers. Figure 5.11 shows successful plane seeds

[8] Segments at a small angle ($< 10^{\circ}$) to the epipolar lines must also have at least *one* endpoint consistent with the transformation. This inelegant extra constraint is necessitated by the degenerate behaviour of horizontal lines.

and their consensus sets on the `cube` pair.

Despite the varying accuracy of coplanarity grouping with different seeds, it was found that candidate matches belonging to a coplanar group were almost invariably correct matches — we conclude that grouping by common affine transformation is valuable as a *matching constraint* as well as a cue to reconstruction. By giving these matches extra support in the disambiguation process, a significant improvement was seen in the accuracy of correspondence.

Figure 5.12 shows the matched segments on the `test`, `roof`, `lab` and `blocks` scenes which were consistent with one or more plane hypotheses — matches are sparser than figure 5.8, but more reliable.


(a)

(b)

(c)

(d)

Figure 5.10: Plane grouping by consensus: (a,b) a 'seed' used to generate a plane hypothesis; (c,d) segments consistent with the affine transformation.

```
• for each pair of line-segments intersecting corresponding epipolar bands
    • evaluate the pair as a candidate match
    • if criteria met, calculate intrinsic support
• for each candidate match
    • enumerate FRIENDS (using monocular figural relations)
    • each match gives extra +ve support to its FRIENDS
• while candidate matches remain
    • enumerate winning matches with more support than any RIVAL
    • sort winners by support difference over nearest RIVAL
    • for a proportion of the least ambiguous winners
        – promote winning match to a confirmed match
        – for each of its RIVALS
            ∗ withdraw support from its FRIENDS
            ∗ destroy the rival match
        – give extra support to FRIENDS
        – enumerate plane seeds that can be formed with promoted matches
    • for each new plane seed
        – for each other candidate or promoted match
            ∗ test for consensus with affine transformation
            ∗ give +ve support to coplanar candidate matches
        – if there are ≥ 4 supporting matches, record plane hypothesis
```

Table 5.2: The combined matching/grouping algorithm, which forms line segment matches, propagates FRIEND and RIVAL constraints, and generates plane hypotheses which help to guide the matching process.

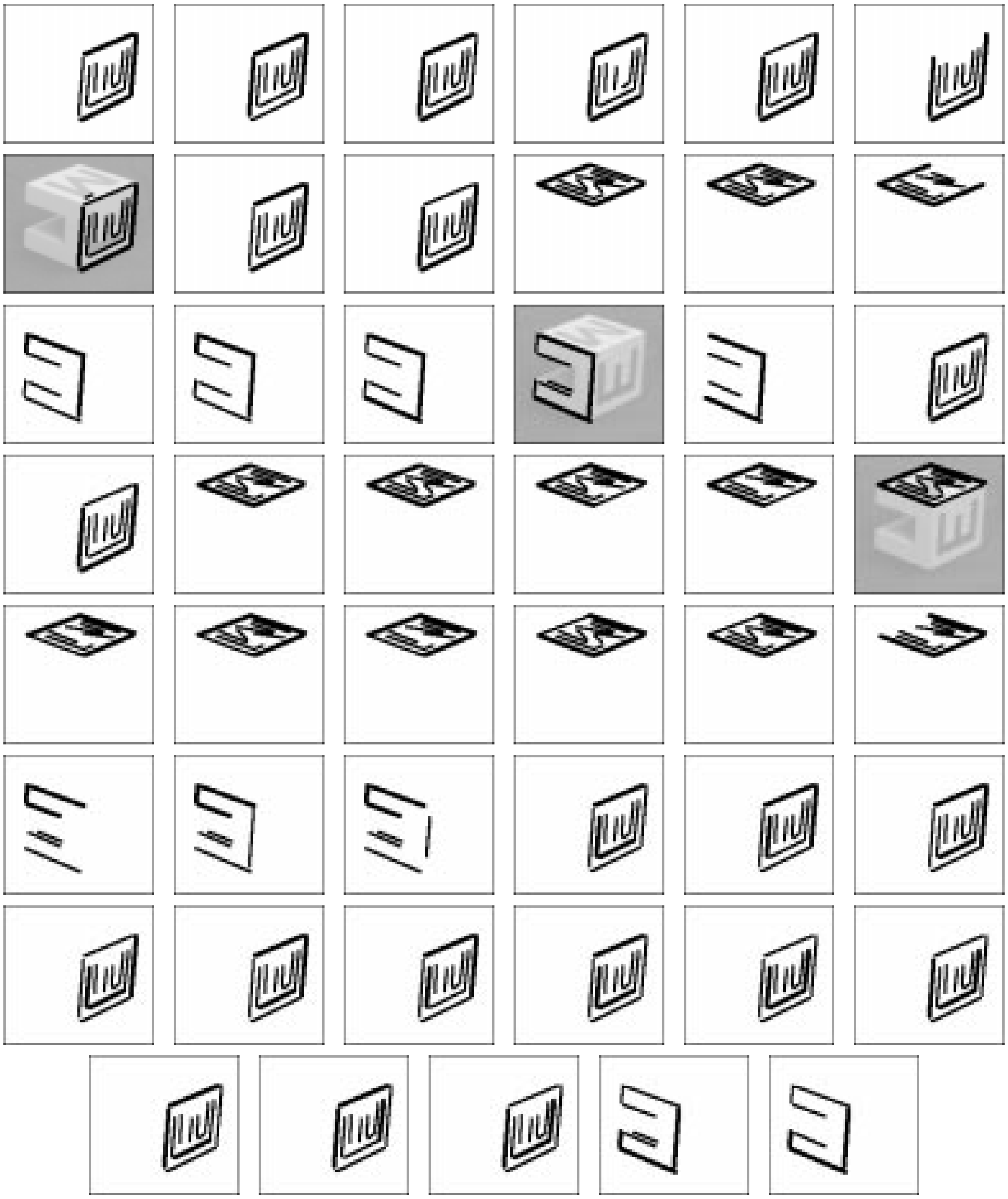


Figure 5.11: Plane seeds (bold) and final matches consistent with the affine transformation for the `cube` pair. The seed on each plane with the largest consensus set is shown against the original image (left view). Apparent duplicates are where different junctions have been used to disambiguate a parallel seed.

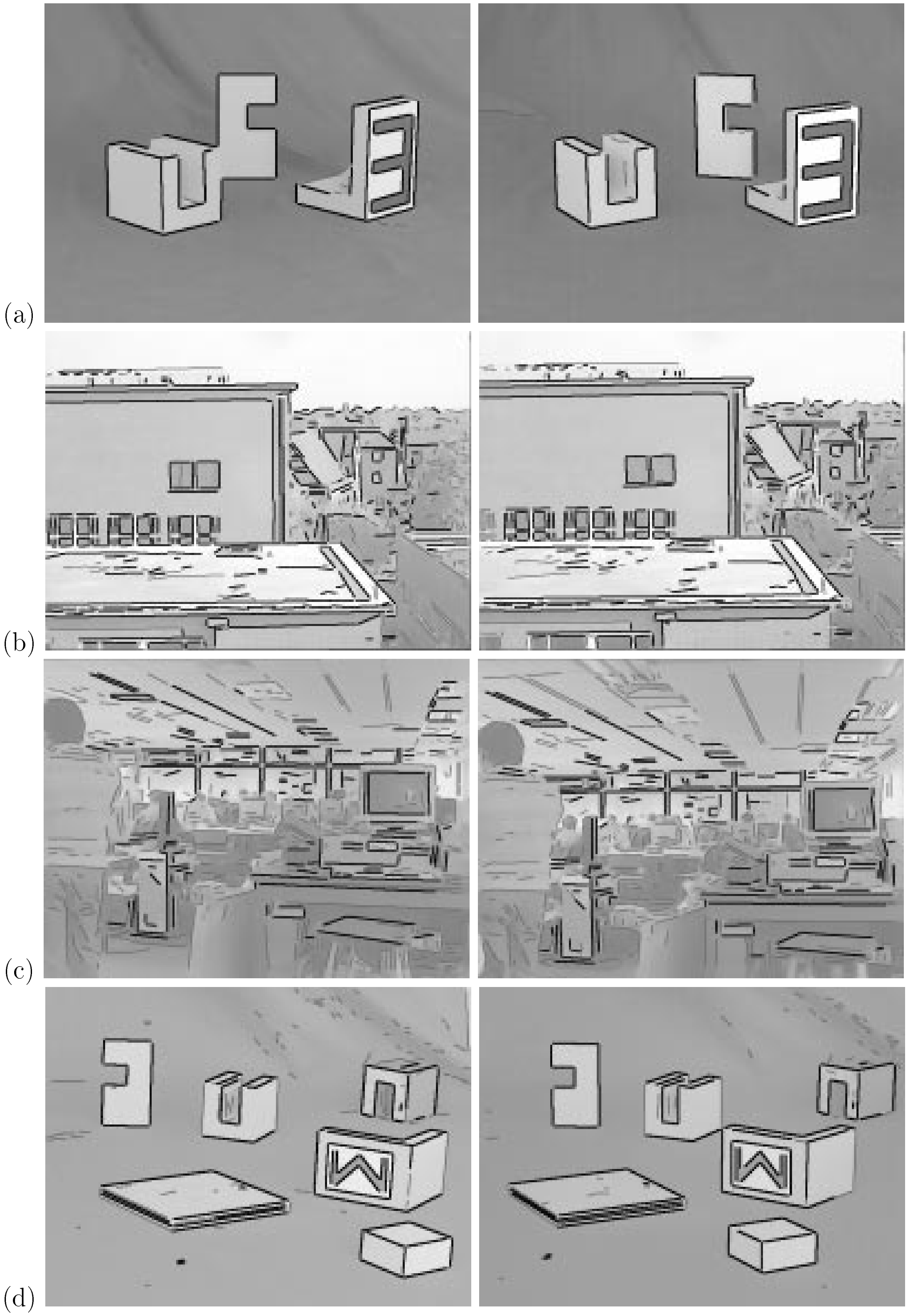


Figure 5.12: Matching results for the `test`, `roof`, `lab` and `blocks` scenes, showing only matches consistent with one or more plane hypotheses (in black).

5.5.3 Hypothesis selection

We have now obtained reliable correspondences for line segments, and a set of plane hypotheses consisting of an affine transformation between views. This may be quite large, and contain multiple hypotheses for each plane. We must therefore select an appropriate set of hypotheses to form a global segmentation of the scene into planar groupings.

If we can assume that the scene contains a discrete set of planes, rather than gently curving surfaces, we would expect the consensus sets of plane hypotheses to exhibit *convexity* in the neighbourhood of each plane — that is, the most representative model for each surface will be the one with the largest number of supporting segments, and hypotheses further from the actual plane model will have smaller consensus sets.

The following rule is therefore applied to select locally optimal plane models, and reduce the number of hypotheses: **If a plane hypothesis has at least half of its consensus set of edges in common with another hypothesis, which has a larger or equal consensus set, the first hypothesis is discarded.** To prevent unnecessary deletions, the hypotheses are first sorted into descending order of support.[9] The rule is then applied in turn to each pair until all duplicates have been deleted.

Results

The `cube` images are correctly segmented into the three planes marked in figure 5.11. Note that one short edge segment is incorrectly assigned to both the top and right faces of the cube. Figures 5.13 to 5.16 show the plane models recovered from the `test`, `lab`, `roof` and `blocks` scenes.

The `test` and `blocks` images yield a small set of distinct planes, which include all of the prominent planes in the scenes. Several of the edge segments are (correctly) grouped as belonging to two adjacent facets; there are also a few ‘accidental’ groupings of unconnected segments which are approximately coplanar with another facet. Grouping accuracy is noticeably coarser for edge segments close to the horizontal (the approximate direction of the epipolar lines).

[9] Where hypotheses have the same size of consensus set, they are ordered according to the total *length* of supporting edge segments.

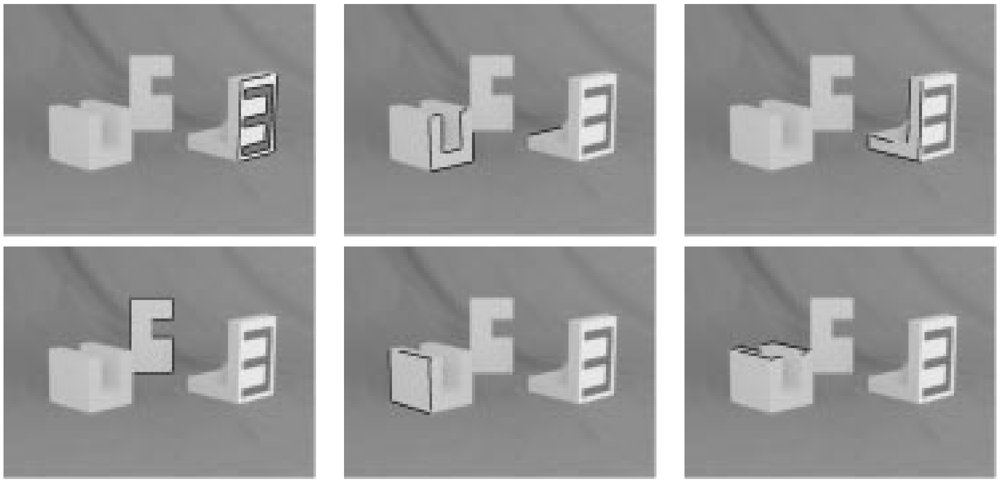


Figure 5.13: Planes recovered in the `test` scene

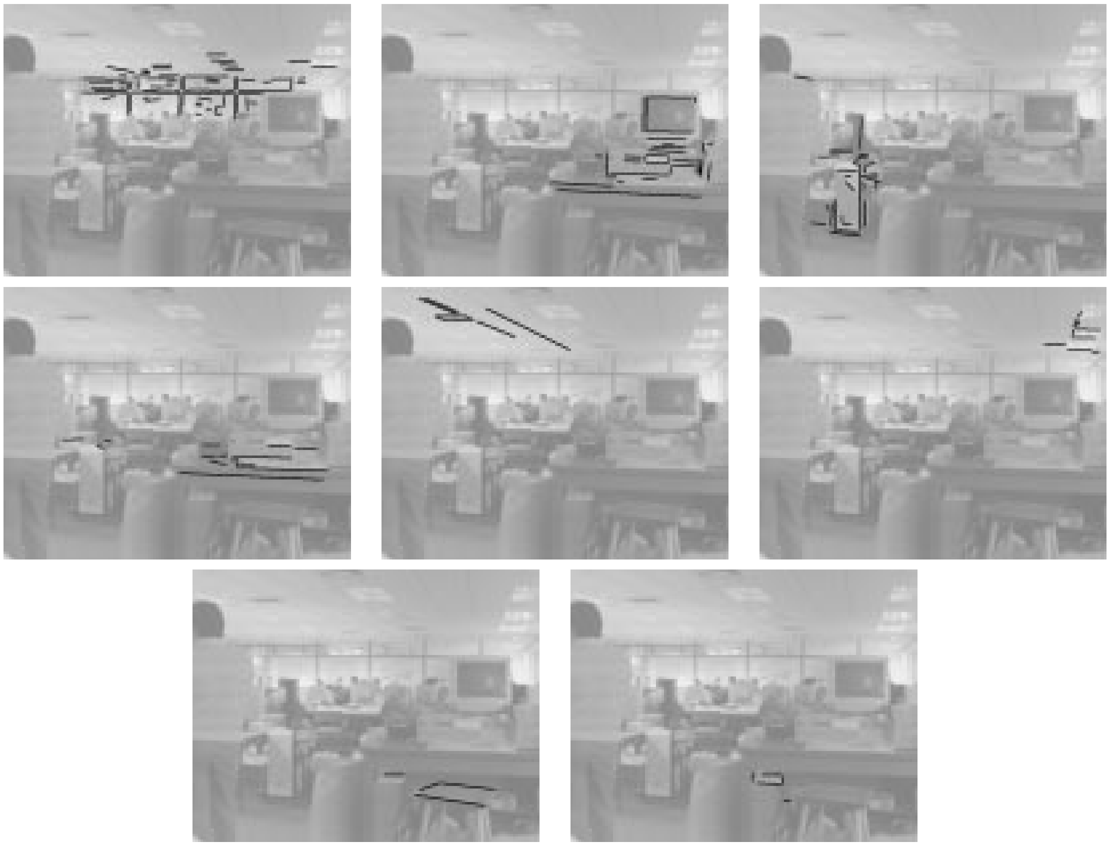


Figure 5.14: Planes recovered in the `lab` scene

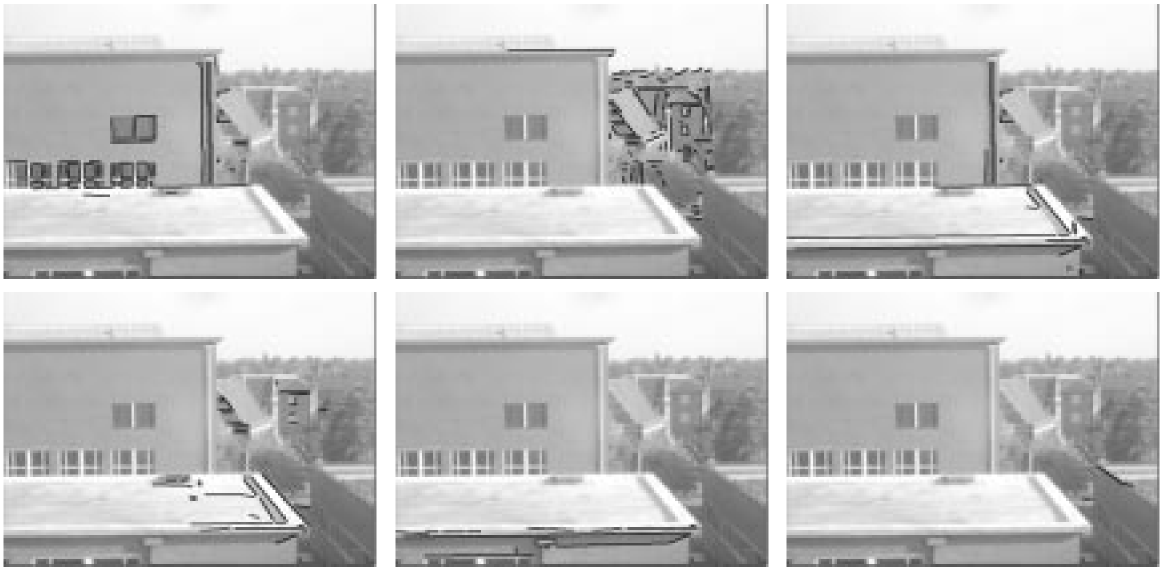


Figure 5.15: Planes recovered in the `roof` scene

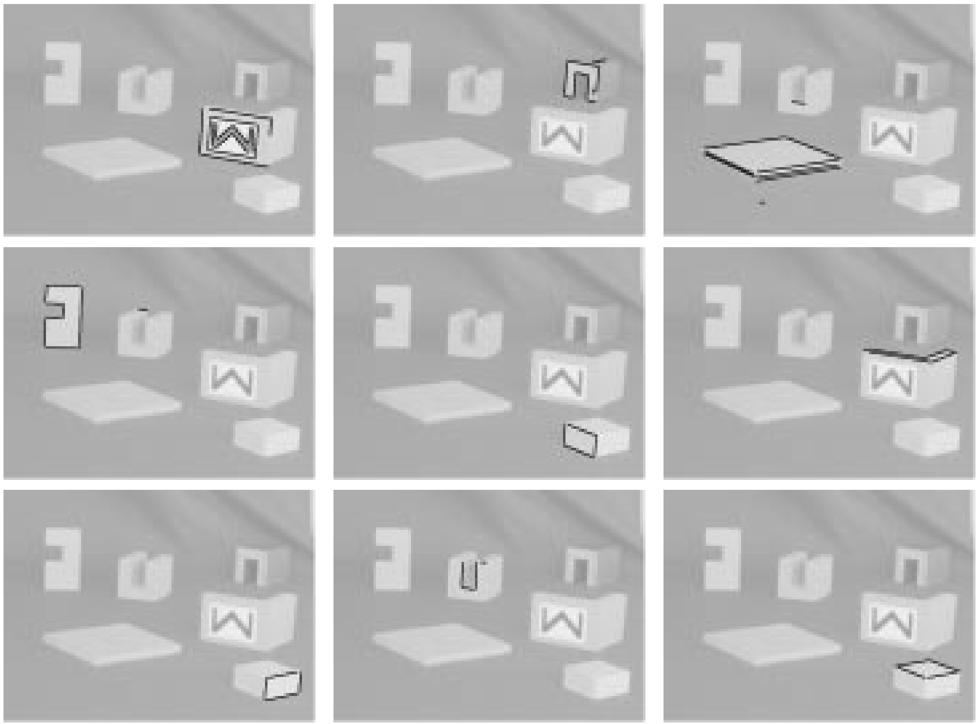


Figure 5.16: Planes recovered in the `blocks` scene

For successful plane segmentation, a reasonably large camera baseline is needed (inter-camera angle above about 20°), and the conditions for weak perspective must be met by all visible surfaces. The two general scenes do not obey these conditions, and are less accurately segmented: in the `lab` scene, the ceiling is strongly foreshortened and becomes split into two distinct groups; in the `roof` scene, distant planes cannot be resolved from one another, and the gabled buildings and skyline are all grouped together as a single plane.

5.5.4 From planes to facets

By matching edges, we have obtained an essentially wire-frame description of the scene, and whilst it is possible to detect coplanar sets of line segments, these do not uniquely define the actual surfaces: this is known as the *figure–ground problem*. Some *a priori* assumptions must be made in order to resolve it and obtain a surface description.

For the purposes of this investigation, we define as *facet* to be the union of connected sets of coplanar edges whose convex hulls[10] intersect one another. It is assumed that such groupings define physical surfaces in the scene. We allow for the possibility that two or more distinct facets conform to the same plane model. There may also be some isolated edges which conform to the plane model but are not attached to any facet — these 'accidental' coplanarities are not used for surface reconstruction (they also account for most of the cases in which a near-horizontal edge has been incorrectly grouped with one or more planes).

Formation of facets

To enumerate the facets, each plane model is partitioned into connected components (by extending the *junction* relation between edge segments) and the convex hull of each component formed in a cyclopean image (coordinates averaged between views).

Components whose convex hulls intersect are considered to belong to the same facet; thus facets can be built up by a simple merging process. This scheme successfully segments most of the facets of the indoor scenes — results are shown in the following section.

[10] A convex hull in 2-D is the smallest convex polygon enclosing a set of features. In the case of line segments, it is the same as the convex hull of all their endpoints.

5.6 Edge and facet reconstruction

Both coplanarity and connectivity constraints can be brought to bear on the reconstruction of line segments in three dimensions, to overcome the inherent inaccuracy of edge-based stereo when the epipolar constraint is not precisely known.

In the absence of full camera calibration, features can be reconstructed in a space whose basis consists of three linearly independent combinations of the (u, v, u', v') coordinates. Where weak perspective applies, this will result in an affine reconstruction. Here we choose to reconstruct features in $(\frac{1}{2}(u+u'), \frac{1}{2}(v+v'), (u'-u))$ space: that is, cyclopean image coordinates and disparity in the u direction. This is a convenient image-based representation of the scene.[11] Because the endpoints of matching edge segments may not coincide between views, we reconstruct their *union* in space, using the outermost pair of the four visible endpoints [6].

5.6.1 Facet reconstruction

Before reconstruction begins, the affine transformation defining the plane of each facet is re-estimated from its final set of consistent edges by linear least squares (recall that previously, it was defined only by the *seed* edges for the entire plane-group). To reconstruct edges on a facet, this affine transformation is used to obtain point correspondences between views. An affine transformation can also be derived which will transfer the facet to a plane within the 3-D cyclopean–disparity space.

A number of edge segments may lie on more than one facet, because the facets intersect on that line, or are very nearly coincident. In such cases, good results were obtained by averaging the endpoint coordinates of the segment over the relevant facets. The use of facets to constrain reconstruction allows us to overcome vertical disparity, which cause errors in the reconstruction of edges close to the horizontal, as well as removing the component of any noise perpendicular to the plane.

5.6.2 Facet boundary description

The simplest description of a facet boundary is its convex hull; but this supposes that all facets are convex. Many of the test objects studied had non-convex facets (sometimes L or C shapes), whose concavities correspond to sites where grasping

[11] For camera configurations close to 'parallel,' the disparity coordinate will be nearly orthogonal to the cyclopean coordinates and may be considered a measure of depth.

might reasonably be attempted. We therefore form a bounding polygon in the cyclopean view which is a slightly modified convex hull: for each segment of the convex hull which does not correspond to a physical edge but whose endpoints lie on a connected component, we search for a chain of connected edges that bridges the gap whilst still enclosing all the edges of the facet. If such a bridge exists, it will be unique and may be interpreted as a concavity in the boundary of the facet.

This procedure allows concavities to be detected whenever there is a connected boundary, and reverts to a convex hull where the matched edges are patchy or fragmented.

5.6.3 Reconstruction of other edges

Where edges do not lie on any facet, junction relations are applied where available, to increase the accuracy of reconstruction; otherwise, we must revert to using the estimated epipolar constraint to reconstruct line segment matches:

Edges with a junction at both ends. If a segment has a junction at both ends, the intersection points at the two junctions are taken to be the corresponding endpoints of the segment, and reconstructed in the 3-D affine frame using linear least squares (if there are multiple junctions to choose from, the ones closest to the estimated epipolar lines are selected).

Non-horizontal edges with a junction at one end. The intersection point is used to reconstruct one of the ends, and the other is reconstructed using epipolar constraint information, on the assumption that the offset between predicted and actual epipolar lines is the same across the length of the segment, which is generally true if the segment is short.

Non-horizontal edges. For the remaining edges making an angle of more than 10° to the epipolar lines in both views, the estimated epipolar constraint is used to reconstruct their union in 3-D space.

This leaves a small number of unconnected ungrouped edges parallel to the epipolar lines. These are degenerate, and *cannot* be accurately reconstructed in stereo, without some additional constraint. They are therefore discarded.

5.6.4 Results

I. Epipolar constraint based reconstruction

This is the traditional approach to stereo reconstruction of edges [7, 100, 6, 32], but it gives poor results in weakly-calibrated stereo, since contours at small angles to the epipolar lines are very sensitive to small errors in epipolar geometry.

Figures 5.17(a) and 5.18(a) show the output of the matching algorithm reconstructed using the epipolar lines to form point correspondences, with synthetic top, cyclopean and side views of the `blocks` and `test` scenes. The depths of reconstructed features are disrupted, due to noise and to inaccuracies in epipolar constraint fitting.

II. Using endpoint coordinates

If the endpoints of corresponding line segments represented the same points in space, they could be reconstructed without reference to the epipolar lines.

Figures 5.17(b) and 5.18(b) show reconstructions based on this assumption. It is approximately correct for many of the edge segments, but fails for those which have been fragmented by noise or truncated by occlusion. Errors in endpoint correspondence can cause large errors in disparity.

III. Using coplanarity and junction constraints

Segments grouped in a facet are reconstructed using the affine transformation associated with the facet to constrain their reconstruction in space. Other segments are reconstructed using *junctions*, where present, to obtain point correspondences and defeat errors in the epipolar geometry. The use of junctions is more robust than endpoints. The remaining ungrouped segments are reconstructed using the epipolar constraint.

Figures 5.17(c) and 5.18(c) show the `blocks` and `test` scenes reconstructed using the algorithms proposed in section 5.6. Facet boundaries are also shown, in grey. Using facets and intersections, the accuracy of reconstruction is greatly increased, provided the groupings are correct; but in figure 5.17(c) some of the segments on the flat object are incorrectly grouped as coplanar.

The modified convex hulls correctly describe the boundaries of most of the facets. Subjectively, this appears to be a good solution to the figure–ground problem, in the absence of detailed prior models of the objects in the scene.

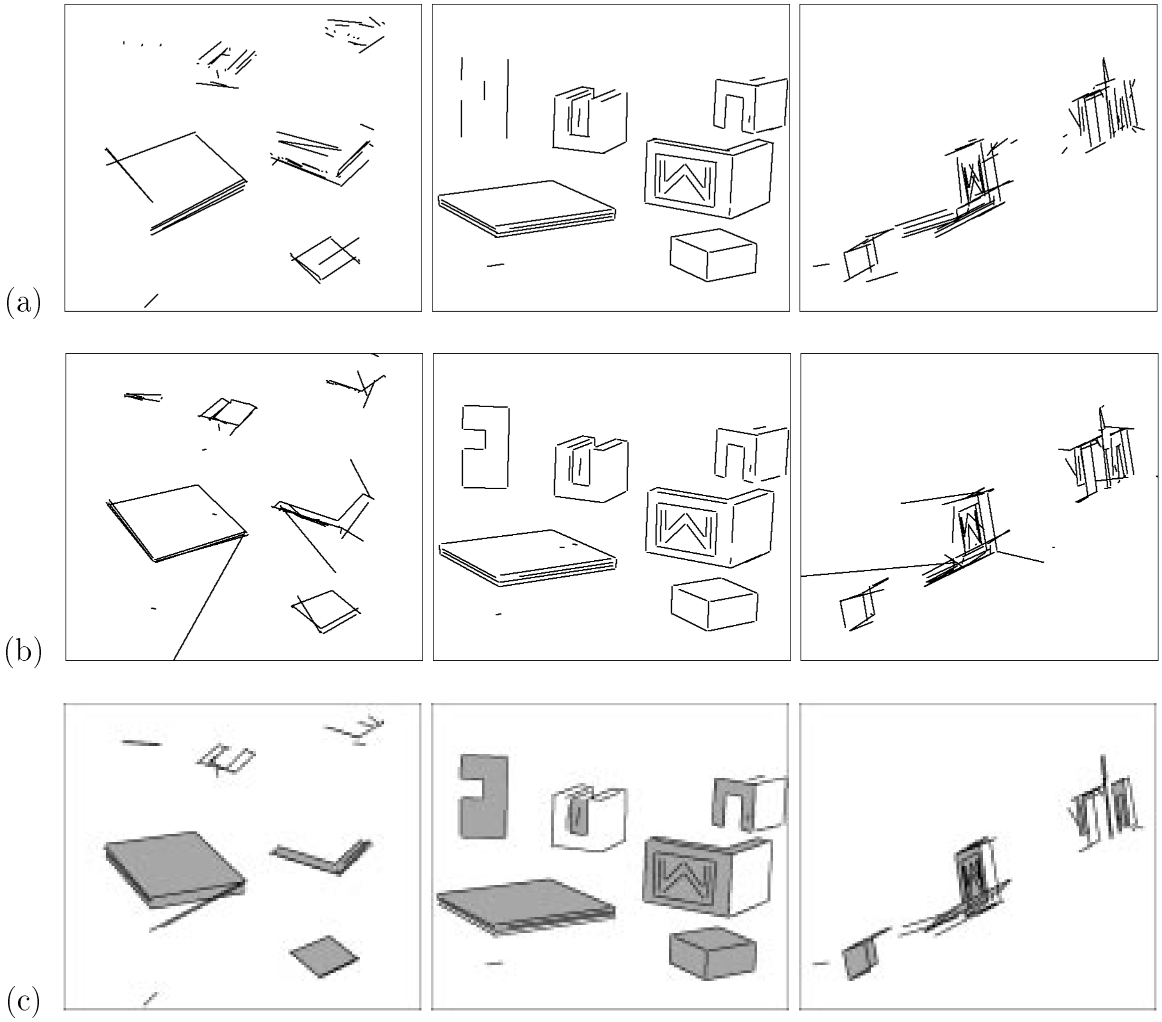


Figure 5.17: Reconstruction of the `blocks` scene illustrated by synthetic top, cyclopean and side views, using: (a) epipolar matching only, (b) corresponding endpoints, (c) facet grouping and junctions.

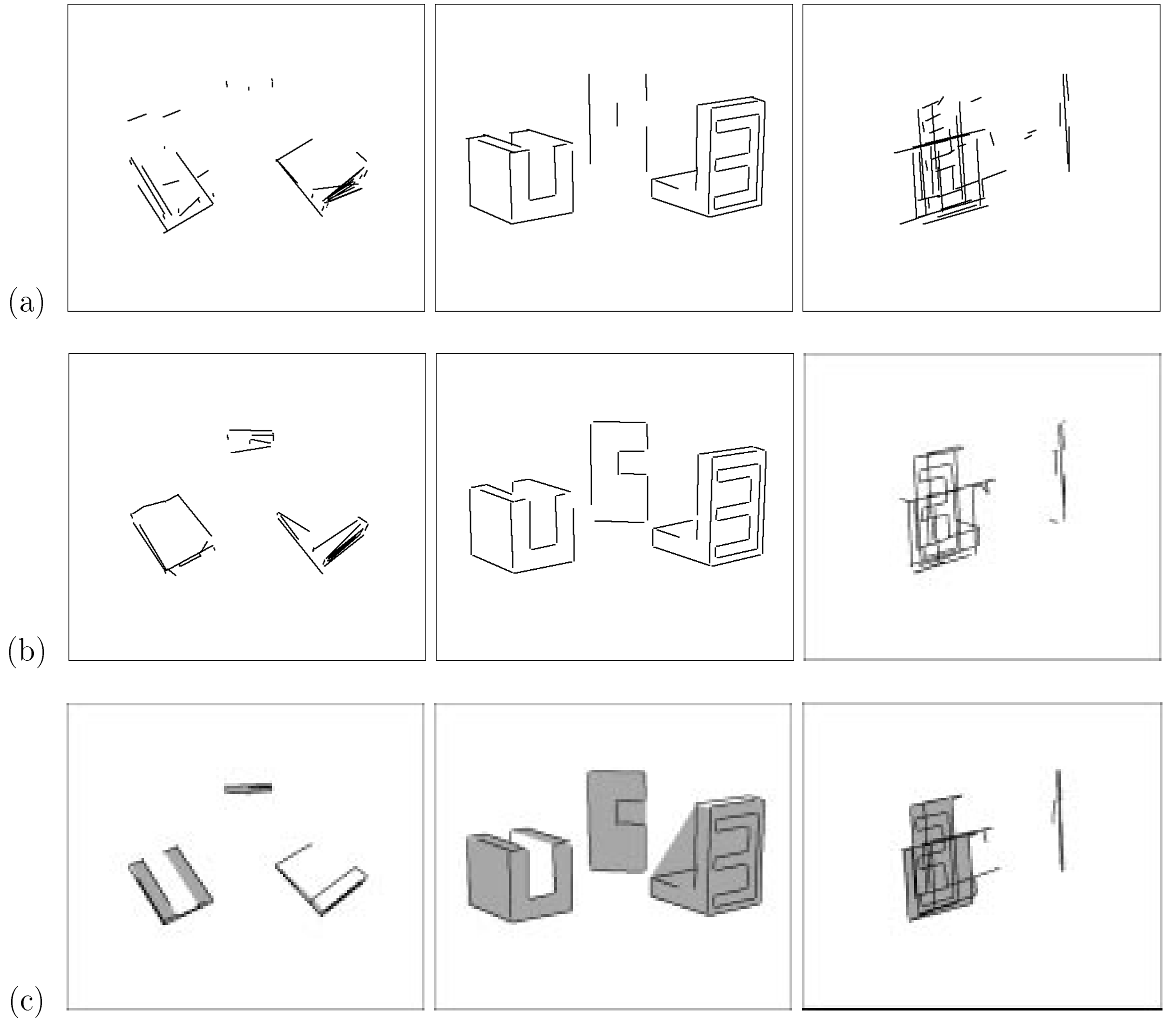


Figure 5.18: Reconstruction of the `test` scene illustrated by synthetic top, cyclopean and side views, using: (a) epipolar matching only, (b) corresponding endpoints, (c) facet grouping and junctions.

5.7 Discussion

5.7.1 Summary

In this chapter, existing ideas have been combined to form a novel algorithm for stereo matching, grouping and reconstruction of line segments. The novelty of the system is two-fold:

1. It has been specifically designed to operate in the 'weakly calibrated' case in which epipolar geometry is known up to a small error. The linear model of epipolar geometry is used throughout. This gives the system robustness, as it removes the need for accurate calibration, whilst retaining the disambiguating power of an approximate epipolar constraint.

2. Image-based coplanarity grouping is integrated into an already complete stereo matching algorithm: the combined system does not rely on the existence of junctions or planes in order to match features between views, but if either are detected, they are used to guide both matching and reconstruction.

The chapter also investigated the ways in which errors can degrade stereo reconstruction, and explained why some previous approaches are unsuited to weakly calibrated stereo.

The new algorithm is best suited to weak perspective images of scenes dominated by planar facets and straight edges, and gives excellent results in this case; it also copes gracefully with more general indoor and outdoor scenes. The system does not require precise calibration: 4 points provide an adequate estimate of the epipolar geometry. It can even 'bootstrap' its epipolar constraint from a suitable initial guess.

Much use has been made here of simple binary relations between segments, such as the *junction* relation which is quick to determine, and has proven useful as a grouping heuristic as well as a cue to 3-D structure. Image-based plane grouping is another valuable tool for the matching, reconstruction and segmentation of indoor scenes. Without it, reconstruction is sensitive to errors and planes can become fragmented. The subsequent use of 3-D grouping has been advocated [46, 6] to reduce these errors, but this is only possible when the system is well enough calibrated that planes may be reliably distinguished in space.

5.7.2 Accuracy of plane and facet extraction

On images of polyhedral blocks, most of the surfaces were correctly detected. On the views of more general scenes, planar facet recovery was less precise, because the stereo system was operating close to the limits of camera resolution [52]. The system also grouped very distant features into a single plane, and fragmented planes which were distorted by strong perspective.

The extraction of facet boundary descriptions necessarily made use of *ad hoc* assumptions which, whilst adequate for simple ‘blocks’ type shapes, might require modification for more general use. For instance, it did not allow for the possibility of facets with holes in them. The scheme presented here also neglects the analysis of *occlusion* between facets, which can itself be a powerful cue to scene structure and segmentation [41, 78].

5.7.3 Use of edge segment based stereo

The use of line segments to represent image features was successful in dealing with images of polyhedral objects, but is less useful for matching more general scenes, particularly natural ones. It is probable that much of this approach could be extended to the matching of parametric curve segments fitted to edges in the image.

Any feature-based stereo system depends on the extraction of primitives from the images which are:

- accurately identified and localised,
- matchable between images and stable with respect to viewpoint changes,
- sufficiently dense to recover the structure of the scene.

We have rejected a corner-based approach because it returns only sparse correspondences, which are not always sufficient to describe the shapes and boundaries of facets in the scene. A recent corner-matching algorithm [138] was tried on several test images and found to be unsuited to polyhedral scenes, because the corners could not be reliably distinguished by correlation. With sparse and often coplanar correspondences, fundamental matrix fitting may fail even when a robust estimator is used [127]. Edge segment matching is better suited to the reconstruction of depth and orientation discontinuities which describe the visible surfaces, and figural relations between segments (junctions) provide cues to the presence of coplanar subsets, and permit epipolar constraint fitting and uncalibrated reconstruction.

5.7.4 Computation time

Table 5.3 gives timings for all the images shown in this chapter.[12] When operating on uncluttered scenes, the matching and plane grouping algorithms are quite fast; by far the slowest component is edge detection. When presented with more detailed images, computational complexity approaches $\mathcal{O}(n^3)$ in the number of line segments.

Execution time for matching, grouping and reconstruction of the `cube` pair was under 1 second, whereas for the `roof` pair (the most cluttered of the test scenes) it was 1.5 minutes.

Name	`cube`	`test`	`blocks`	`lab`	`roof`
Image size	232×185	640×480	640×480	704×512	768×576
Number of line segs	39, 40	61, 61	174,131	584, 572	600, 544
Edge detection	2.3	14.1	14.2	20.4	22.8
Line segment fitting	0.2	0.4	0.6	3.6	4.5
Monocular relations	0.09	0.19	0.61	6.93	7.08
Enumerate matches	0.01	0.02	0.04	0.22	0.24
Enumerate FRIENDS	0.18	0.24	1.25	24.5	42.0
Matching/grouping	0.46	1.22	2.23	17.8	42.8
Hypothesis selection	0.17	0.27	0.27	0.41	1.81
Facets/reconstruction	0.04	0.08	0.13	1.21	2.76

Table 5.3: Timings in seconds for matching and reconstruction of the test images

[12] For a prototype implementation in 'C', running on a SPARCstation 20.

Chapter 6

Visually Guided Grasping: Implementation

This chapter describes a grasp planning scheme based on stereo reconstructions of polyhedral objects, using simple rules to select an appropriate facet and edge pair for grasping. The algorithms developed in this dissertation are combined to implement visually guided grasping.

6.1 Introduction

Algorithms have been developed for robot grasping by visual feedback, the interpretation of pointing gestures to specify objects to be grasped, and the reconstruction of unknown polyhedral objects in uncalibrated stereo. Our ultimate goal is to combine these algorithms into a visual grasping system whose operation comprises three phases:

- **Programming phase.** The user indicates what operation is to be performed using pointing gestures.
- **Planning phase.** The system uses robust stereo cues to determine the shape of the object to be grasped.
- **Execution phase.** The system uses real-time visual tracking and feedback to accurately align the robot gripper with its target.

The three parts of the system communicate in *image-based* terms, which are independent of any estimates of the camera positions or parameters.

To complete the system, we require *grasp planning* to select the appropriate grasping operation for a given part. This is described in the following section. No new theory is advanced, but it is shown that the stereo reconstructions of chapter 5 are sufficient to support grasp planning on block-like objects, using a simple scheme which searches for parallel facet-and-edge pairs and tests for collision with a model of the robot.

6.2 Grasp Synthesis

6.2.1 Introduction

Before grasping can be performed, it must be planned, to specify the robot configuration required to grasp the object successfully. For most robotic systems, grasps are synthesised offline, either by hand or by analysis of CAD models [75]. The grasping operation relies on *recognition* of the object in order to select and retrieve the appropriate grasping method [103], although this may be parameterised to allow for variations in the pose or dimensions of the object to be grasped.

Here machine vision is used to facilitate the grasping of an *unmodelled* object, by means of stereo reconstruction of its visible surfaces. A grasp is planned for a conventional parallel-jawed gripper. The grasp is defined in terms of one of the reconstructed facets of the scene, and an edge segment behind it. Although the evaluation of grasp sites requires metric information about the reconstructed scene, they may be *specified* in terms of image coordinates, and executed accurately using image-based servoing.

6.2.2 Notes on grasp synthesis for a parallel gripper

Force closure

The robot used in this project has a parallel-jawed gripper with two padded fingers. It is well known that such a gripper is suitable for grasping objects across two parallel (or near-parallel) surfaces [92, 79, 11] by placing the contacts so that both of the surface normals are at a small angle to the axis of the gripper (figure 6.1). The maximum permissible angle is $tan^{-1}\mu$, where μ is a lower bound on the coefficient of friction. Provided the robot can withstand the necessary compressive, frictional and torsional forces, such a grasp will achieve *force closure*, i.e. it will completely constrain the pose of the object and can resist external forces or torques in any direction [92]. Grasps with a parallel gripper are always dynamically stable, because the gripper mechanism constrains the relative finger pose to a single degree of freedom, so the angles between finger and object surfaces cannot change while the object is being grasped.

Other constraints

For a grasp to be permissible, it must not only have force closure but must be *feasible*, i.e. within the physical reach of the robot, and *collision-free*, i.e. the robot can adopt the grasping configuration without colliding with this or any other object other than at the points of contact. In a complete motion planning system, it must also be checked that the grasp configuration can be reached from a given starting configuration without collision, that the object can be carried away, and that the grasp chosen is also suitable for putting the object down in the required place [75]. These higher-level path planning issues are not addressed here.

The general paradigm for parallel grasp planning [62] is therefore to *search* for a pair of facets or patches which support a force closure grasp, *hypothesise* a suitable robot configuration to perform the grasp, and then to *test* the grasp for accessibility and collisions; if the test fails, alternative grasp sites must be enumerated and tested.

Grasping smooth surfaces

On any smooth object, there will always be at least one parallel grasp at its maximum diameter, though this is not generally the only feasible grasp. Taylor and Blake [11, 125] describe theory and algorithms for finding *extremal grasps*, which are finger placements requiring a minimal coefficient of friction to attain force closure, from a B-Spline representation of an object's silhouette. The spline allows them efficiently to calculate the *symmetry* and *asymmetry* sets of grasping configurations (where the angles between the normals at the grasp points and the line joining them are equal on the same or opposite sides, respectively). These sets intersect only at parallel grasps. A third set which they call the *critical set* is computed and used to find grasps with local extrema of μ. They extend their analysis from two-dimensional to three-dimensional smooth convex objects, for which a band of the surface has been reconstructed by using a B-Spline snake to extract its silhouette in multiple views [22].

General surfaces

When grasping general three-dimensional surfaces, the problem of grasp planning is dominated by the search for force closure grasps, as the configuration space (the placement of two contacts on a 2-D surface) can be quite large. Rutishauser and Stricker [111] describe a search for suitable grasps in scenes which have been reconstructed by a laser rangefinder. Range images from three different viewpoints are fused — this is so that some opposing parallel patches will be detected. The scene is segmented by continuity into regions likely to correspond to individual objects, and the system searches the space of *pairs* of contact points on each object to find optimal grasps (using an objective function based on the local geometry of the contact points to assess their suitability for a parallel gripper). A strategy known as *tabu search* is employed [43], which combines steepest-descent with rules which cause it to avoid previously-visited minima. This scheme allows multiple candidate grasps to be enumerated, for selection by some higher-level planning process.

Summary

With a parallel gripper, grasps can be synthesised by searching for two nearly parallel patches on which a pair of contacts may be placed within one another's friction cones. For curved surfaces, the search space may be large. With a polyhedral model of the object, the search is a simple task of complexity $\mathcal{O}(n^2)$ in the number of facets, followed by 2-D enumeration of point-pairs on those facets to test for feasibility and collisions.

6.2.3 Hypothesising parallel surfaces

We have seen in chapters 2 and 5 how the planar facets of the target object may be identified in uncalibrated stereo, and their shapes and positions reconstructed in an approximate metric frame using just a small number of calibration points.

From a single stereo pair of views, at most half of the object's surfaces will be reconstructed – thus only one of each pair of graspable surfaces will be visible. The presence of the opposing surface is therefore inferred by an appeal to symmetry: we consider each edge segment which lies on the boundary of a visible facet, and search for a *parallel* segment reconstructed behind it on the same 3-D connected component (parallelism can be determined in the images without metric reconstruction). In the absence of other visual cues, it is a reasonable hypothesis that this edge segment marks the boundary of a parallel facet which may not be visible (see figure 6.2).

6.2.4 Feasible grasps

In many robot arm applications, the gripper approaches the workspace from above. A 6-DOF arm is able to grasp surfaces of any orientation, but simpler 4- and 5-DOF arms have restrictions on the orientation of the gripper. Here we restrict ourselves to grasps in which the line joining the fingers is horizontal, i.e. the graspable surfaces are vertical. We therefore consider grasps only on surfaces which are within 20° of the vertical in an approximate metric reconstruction.

Grasps are proposed on points on the edges of the graspable surfaces, since these are the most accessible: we identify points on a front edge segment (the one on the boundary of the visible facet) for which there is a corresponding point on the rear (in the plane normal to the front edge). These are called *grasp sites*.

Due to the dimensions of the gripper, there is a maximum limit on the *width*, or horizontal distance between the graspable surfaces.[1] In this case it is 50mm. The conditions for a feasible grasp are summarised in figure 6.3.

6.2.5 Testing for collisions

It is important to choose a grasping configuration which is free from collisions: that is, in which the gripper does not intersect any part of the scene. To do this, we need to model the shape and size of the gripper. A conservative model of the gripper is shown in cross-section in figure 6.4. The thickness of the model is 40mm. Note that the model extends upwards, so as to guard against collisions with the robot as it approaches its target from above. The model is used to test for intersections between the gripper and any edge or facet of the scene.

Implementation

In our scheme, only the most central feasible grasp site of each facet-and-edge pair is considered and tested for collisions. The collision testing process works as follows:

1. Edge segments and facets are enumerated which protrude above the 'low water mark'. This is the horizontal plane 25mm below the grasp site, which is the

[1] It is permitted for the front and rear edges to be the same, so that the width is zero.

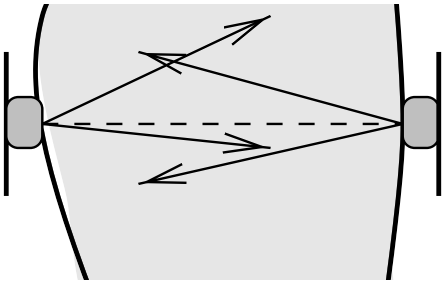


Figure 6.1: Grasping with a parallel gripper: the grasp is successful if the line joining the contacts lies within the *friction cone* of each (i.e. the contact surfaces are nearly orthogonal to the gripper's axis).

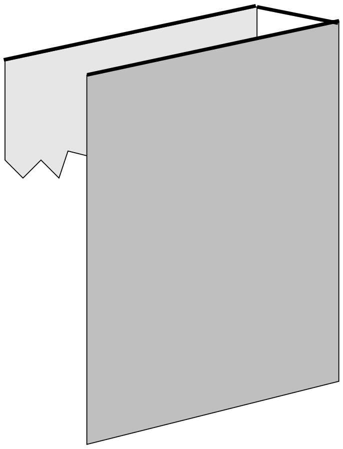


Figure 6.2: Inference of parallel surfaces from parallel edges: if two edge segments on a 3-D connected component are parallel, one lies on a visible surface boundary (dark grey) and the other is behind it, we hypothesise that there is a parallel surface (light grey).

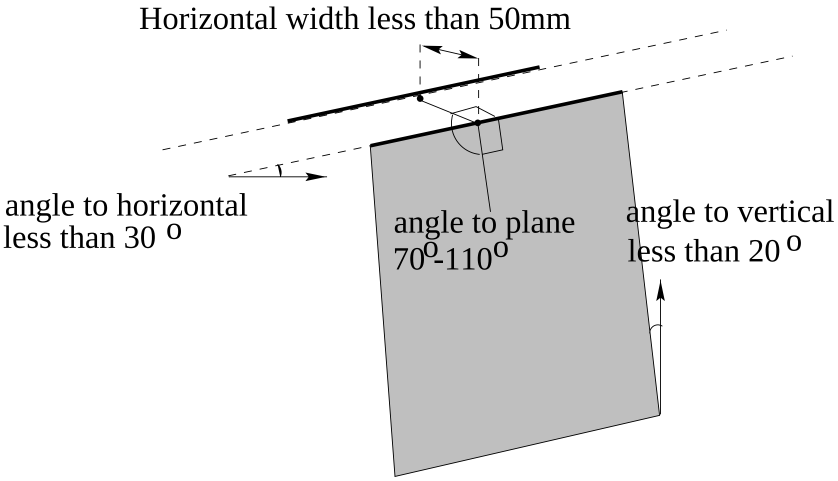


Figure 6.3: Summary of the conditions to be met by a pair of points on a facet and edge pair, to be considered a feasible grasp for our robot.

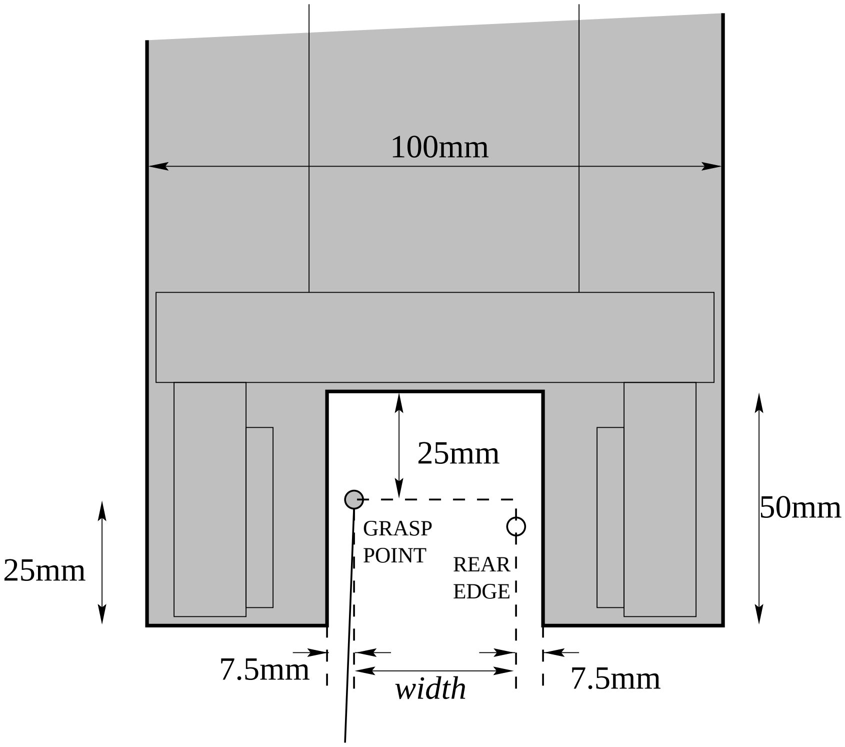


Figure 6.4: Cross section of the robot gripper model used for collision checking. The gripper approaches vertically from above. No visible features may intersect the shaded region.

plane of the fingertips in the proposed grasping configuration. Features are clipped to this plane and projected into a plan view, to test for intersection with either of two rectangles representing the volume swept out by the gripper fingers.

2. If there were no collisions with the fingers, features are then clipped to the 'high water mark' 25mm above the grasp site. Features are tested for intersection with a 40mm×100mm rectangle representing the upper part of the gripper.

In the unlikely event that there is more than one collision-free grasp on a facet, the highest one is selected, as this is deemed to be the most accessible.

6.2.6 Results

Figure 6.5(a) shows the facets of the test scene which were within 20° of the vertical, and their top edges within 30° of horizontal.[2] Part (b) shows grasp sites corresponding to parallel edges (including grasps of zero width) which meet the criteria for a feasible grasp; part (f) shows the two grasps which survive collision testing.

Figure 6.5 parts (c–e) show plan views of the features which protrude above the 'low water mark' used in collision testing, for three feasible grasps. The bold rectangles represent the gripper fingers. The last example indicates a collision between a facet and one finger of the gripper model. The other 4 feasible grasps (not shown) also lead to collisions.

[2] No calibration data were available for this scene, so metric tests were performed on the assumption that the cyclopean u, v and disparity axes were orthogonal, v vertical, with scales of $(\frac{2}{3}, \frac{2}{3}, 2)$ millimetres per pel respectively.

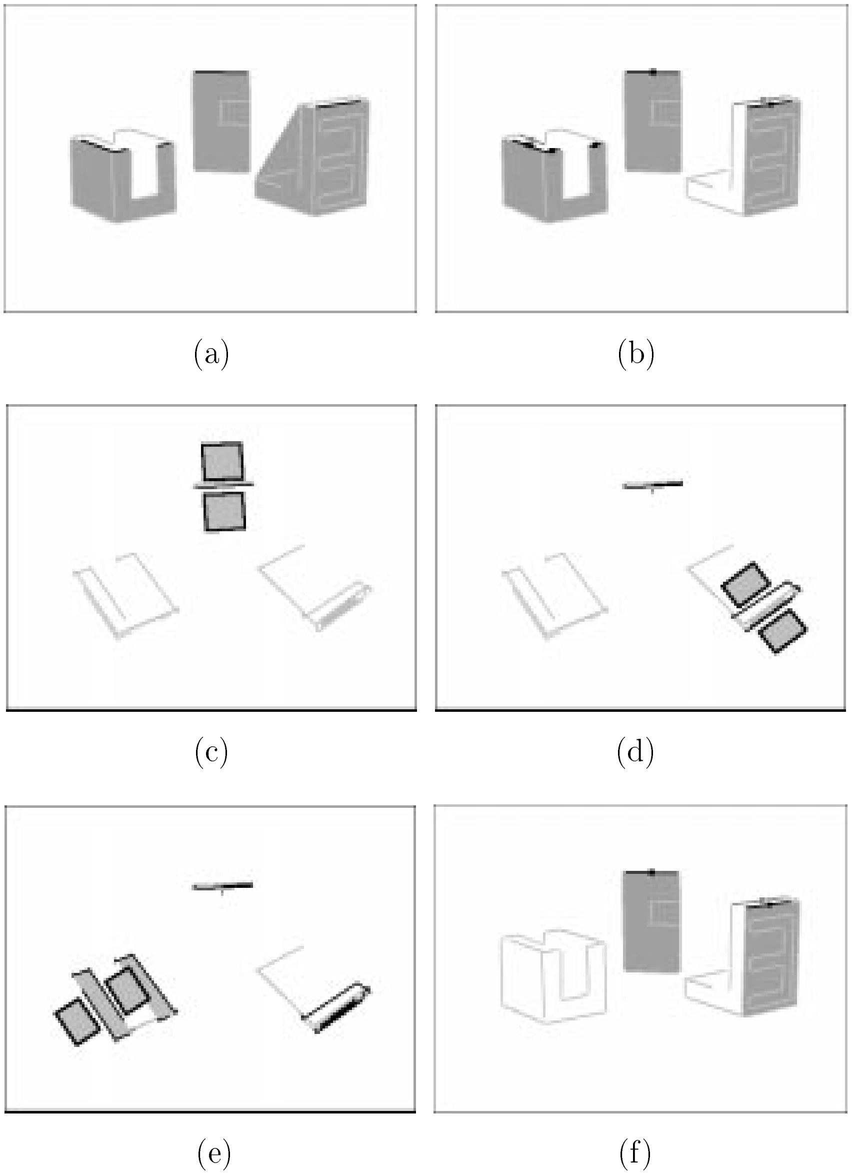


Figure 6.5: Parallel grasp planning on stereo reconstruction of the `test` scene: (a) near-vertical facets and near-horizontal upper edges; (b) feasible grasp sites; (c,d) plan views of successful grasp sites showing finger placements; (e) grasp site where finger would collide with object; (f) collision-free grasps.

6.3 Implementation

6.3.1 Setup

A stereo pair of CCD cameras were set up as usual, fixating on the robot's workspace from a distance of about 2 metres with an inter-camera angle of about 20°. Camera setup was somewhat constrained by the need to view the entire workspace, with some room for the operator's hand, whilst maintaining a high enough resolution for plane grouping to be reliable.

The image positions of the gripper are acquired as in chapter 3, by watching as the robot opens and closes its fingers. The gripper is then tracked as it visits a cage of eight points: the lower four points are used for pointing and define a plane a few centimetres above the ground plane; these and the upper points are also used for approximate stereo calibration and to determine the *disparity limits* of the robot's workspace.

6.3.2 Pointing to the target object

Rather than try to construct a multi-faceted model of the scene, only single plane pointing was used. This is because the test objects were quite small, and visible facets were sometimes at a very shallow angle to the line of pointing: it would therefore be unreliable to expect an operator to indicate a single facet by pointing alone. Instead, the operator indicated a point on the *working plane*, which is a few centimetres above the table, nearest to the desired object. The robot followed the indicated point some distance above, to provide feedback to the operator (figure 6.6(a,b)).

As implemented here, the pointing and grasp planning parts of the system do not communicate with one another, and can be executed in either order.

6.3.3 Stereo reconstruction and grasp planning

For the next phase of operation, the operator removed his or her hand from the scene, and a stereo pair of images was taken. Edge detection, line segment fitting, stereo correspondence and facet description were performed (figure 6.6(c,d)). The grasp planning scheme described above was then executed.

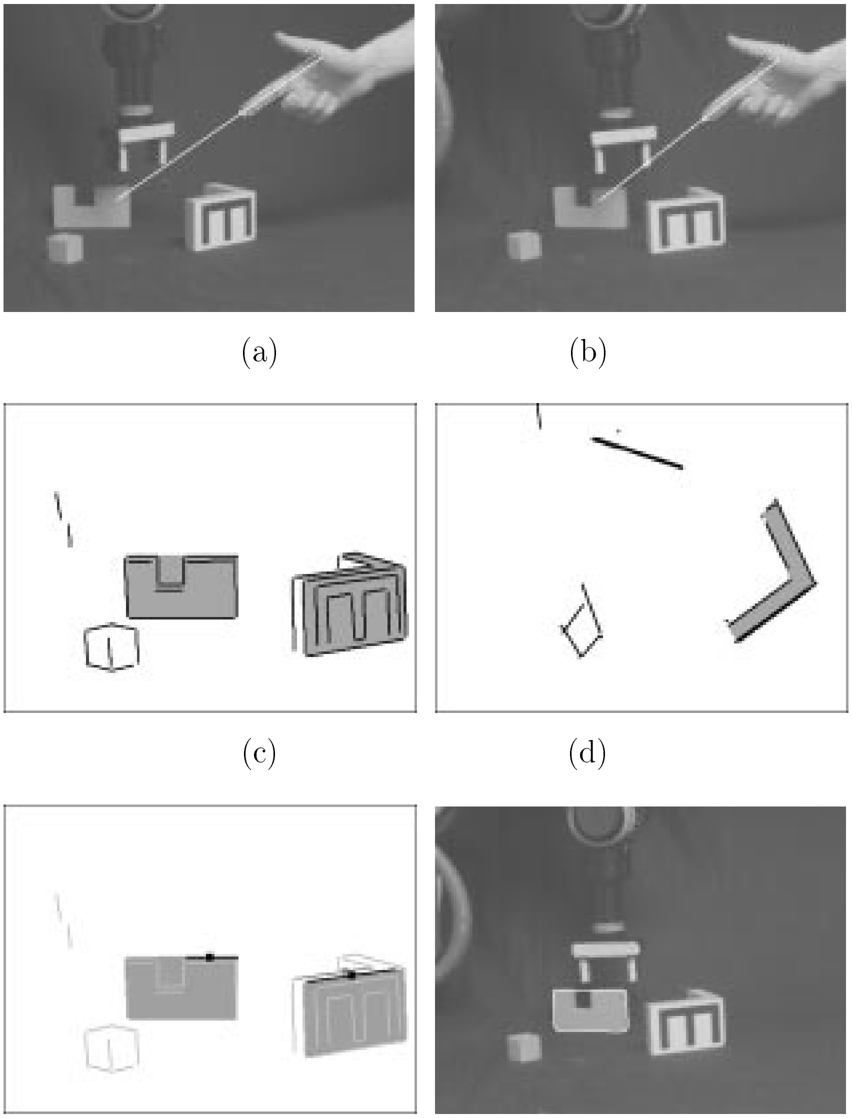
In the example scene, the system detected only two permissible grasp sites, across the top of the wedge-shaped object, and on the rear object. Not enough of the small cube was matched to generate a plane hypothesis, so it did not yield any grasp sites.

6.3.4 Grasp execution

The grasp site nearest to the indicated point was chosen for execution.

The image locations of the facet associated with the chosen grasp site were used to initialise a pair of active contours (this was to detect any small movements of the target object since the original stereo pair was taken). Tracking of the gripper was reestablished, again by watching it open and close its fingers.

Visual feedback was used to align the robot with the grasp site, first by making it coplanar with the target facet, and then performing an open-loop grasping manouever. The position of the grasp site was expressed in a coordinate system based on the target facet.



(a) (b)

(c) (d)

(e) (f)

Figure 6.6: Visually guided grasping of an object: (a,b) stereo views of a pointing hand indicating the object to be grasped; (c,d) reconstruction of the scene; (e) permissible grasp sites; (f) alignment using visual feedback.

6.4 Discussion

A practical grasp planning scheme for collections of straight edges and facets has been presented, which addresses the case in which only one side of the scene has been reconstructed, by hypothesising unseen parallel surfaces at parallel edges. Simple rules enumerate the grasp sites which can be reached from above by a parallel gripper, and collision testing is used to check that no visible features will collide with the gripper during grasp execution. Grasp planning has been used in conjunction with uncalibrated stereo feedback, to execute a visually-specified grasp.

For stereo reconstruction and grasp planning to be successful, camera setup and lighting had to be carefully controlled, so that objects were observed at a high enough resolution, and enough of their edges were correctly detected and matched; otherwise suitable facet descriptions would not be generated and grasp planning would fail. For instance, in the last example, no grasp sites were identified on the small cube because some of its edges were not reconstructed. Further work to improve the robustness of edge segment detection and matching is desirable.

During the period in which edge detection and stereo matching are busy, the robot is 'blind' and cannot track any movements of the scene, contradicting our requirement that the system be insensitive to camera motions. However, if the image motion of the graspable facet is small, it will be located again when its active contours are initialised, and can be tracked until the grasp is executed.

Chapter 7

Conclusions

The dissertation concludes with a summary of the findings and contributions of this project, and notes some areas in which further investigation is warranted.

7.1 Summary

This dissertation has explored the use of *uncalibrated* stereo vision; that is, stereo vision in which accurate camera parameters are not initially known. This includes systems in which a few reference correspondences are available, permitting an approximate epipolar constraint and affine stereo model to be fitted, which we have called *weakly calibrated*.

Affine stereo. In chapter 2, it was argued that a linear model of stereo vision (*affine stereo*), developed from the weak perspective camera model, is well suited to practical uncalibrated stereo systems, particularly when absolute Euclidean reconstruction is not required. Although it is less accurate that the projective model in noiseless calibrated systems, it is more robust to errors.

Numerical simulations have shown that the affine stereo model can be estimated more accurately than the unconstrained projective model, from a small number of noisy reference points, and that it is less sensitive to camera disturbances. Similarly, the fitting of the linear form of the epipolar constraint by least squares is less sensitive to image coordinate errors than the more general fundamental matrix form.

We conclude that linear models are indeed a useful approximation to both the epipolar constraint and the world–image relation, for practical stereo configurations in which external cameras fixate on a compact scene.

Pointing interface. Chapter 4 showed how to interpret images of a pointing hand in uncalibrated stereo, given an image-based description of the surfaces to which the operator may be pointing. The method does not involve 3-D reconstruction, so avoids the difficulty of camera model estimation. Instead, it models the transformation of planes between the images, which was found to be well-conditioned.

A novel user interface was developed based on an affine template active contour to track a pointing hand in real time. Its accuracy was evaluated, and it was demonstrated as a means to control a robot, by interactively specifying points for pick and place operations.

Uncalibrated matching and reconstruction. The problems of matching and reconstruction in weakly calibrated stereo were investigated, noting the weaknesses of current methods for feature-based stereo reconstruction. A novel stereo matching algorithm was developed incorporating image-based coplanarity grouping and segmentation. This allowed an accurate ‘qualitative’ model (local shape up to an affine transformation) to be reconstructed without calibration, and was robust to inaccuracies in the estimated epipolar constraint.

Affine stereo visual feedback. The affine stereo model was exploited to develop algorithms for visual feedback control of a robot manipulator, allowing it to grasp an object by first aligning a surface on the gripper with one of the visible facets of the object. Affine active contours were used to track both the robot and the target object. Since the (coplanar) alignment of robot and target can be tested directly from image measurements, the system is insensitive to disturbances or movements of the cameras.

The parts of the project are united by their use of *image-based* measurements, which do not depend on estimates of the camera positions, or on Euclidean reconstruction.

7.2 Future work

Implementation and integration of these systems revealed a number of weaknesses which could benefit from future work.

Active cameras. For the user interface and for controlling the robot, a broad view of the workspace is required; but for accurate reconstruction of parts, high resolution is needed. These conflicting demands constrained the camera geometry and severely limited the size of the workspace relative to that of the parts. This problem could be alleviated using cameras with zoom lenses, mounted on pan–tilt heads, for active control of the field of view [89]. This would be problematic for a stereo system dependent on *calibration,* as the camera parameters would be continually changing [71], but should cause few problems in affine stereo.

Feature detection. Throughout this dissertation we have relied on the detection of *edges* in the images, for tracking, segmentation and the determination of shape. These are subjectively the most prominent feature type in many polyhedral scenes; however, where surface markings were not present, not all of the orientation and depth discontinuities could be detected as edges: controlled lighting was required to reliably extract enough matchable features to reconstruct all of the planar facets in some scenes.

The isotropic smoothing stage of Canny's edge detector [14], whilst providing some robustness to noise and loss of distracting detail, degrades the localisation of edges, causing nearby edges to interfere with one another (thus disturbing the affine invariance of coplanar groups) and corners to be rounded and lost (making junctions harder to detect). Better results might be obtained by using an edge detection mechanism incorporating *anisotropic* smoothing of the images [96].

Curved contours. Our stereo matching system uses straight line segments as its primitive features, to reconstruct straight edges and planar facets on polyhedral objects. However, many industrial parts are not polyhedral and exhibit curved as well as straight contours.

In chapter 5 it was suggested that the much of our approach could be extended to curve segments: these can be matched using similar criteria to line segments, using junctions to aid disambiguation; in the case of plane curves, matches can be verified using *affine invariant signatures* [114] which establish point correspondences and recover the affine transformation between views. Coplanar curves may be grouped into planes by common affine transformation. Space curves not on a plane will not exhibit affine invariance between views: these must be reconstructed (like ungrouped line segments) using the epipolar constraint.

Contours may also be present that do not correspond to any markings or discontinuities but are the silhouettes formed from rays tangent to a curved surface. These *apparent contours* cannot be matched in stereo because they do not correspond to the same space curve in two views (three nearby views are required to reconstruct local surface structure [22]). A system for the reconstruction of curved objects will need to identify such contours. One way that this can be done is by noting that they generally meet surface contours at a tangent.

Multiple views. The use of 'parallel camera' stereo vision has meant that only parts of a scene (those surfaces visible to both cameras) could be reconstructed, and this has limited the generality of grasp planning from stereo. A multiple camera system would enable more of the surfaces to be reconstructed. By tracking the robot in more than 2 views, redundancy is introduced and the robustness of vision-guided operations could be increased.

In his book, Ayache [6] makes a strong case for trinocular stereo vision. With three or more cameras, the epipolar constraint is generalised (to the so-called *trifocal tensor* [127]) so that 'horizontal' segments no longer lead to degeneracy, and both structure and motion can be recovered from line matches. Furthermore, the correspondence problem is simplified, since a match hypothesised from two views can be verified by testing for a feature in the appropriate position in the third (with uncalibrated stereo, however, this test is more complicated as its position cannot be predicted precisely). However, the extension of our algorithms to uncalibrated trinocular stereo was beyond the scope of this investigation.

Appendix A

Affine template active contours

To track the contours of surfaces on the robot's gripper and and target objects (chapter 3), and of the user's pointing hand (chapter 4), a real-time edge tracking mechanism was employed, based on affine active contours.

A.1 Background

An *active contour* (or *'snake'*) [64] is a curve defined in the image plane that moves and deforms according to various 'forces'. These comprise *external forces*, which are local properties of the image, and *internal forces* which are functions of the snake's own shape. Typically, a snake will be attracted to maxima of image intensity gradient, and used to track the edges of a moving object.

More recent active contours have been proposed based on parametric curves such as B-splines [22]. These are represented compactly by a small number of *control points* from which the curve can easily be interpolated; the parametric form automatically enforces smoothness, so that internal forces are not required. The behaviour of the snake may be further restricted by imposing constraints on the configuration and motion of the control points; using principal component analysis, snakes can even be trained to track particular classes of object [12].

Our model-based trackers are a novel form of active contour. They resemble B-spline snakes [22] but consist of discrete sampling points, rather than a smooth curve. In this respect they resemble 3-D model-based trackers [56]. Pairs of trackers operate independently in the two stereo views. The trackers can deform only affinely, to track planes viewed under weak perspective: this constraint leads to a more efficient and reliable tracker than a B-spline snake, that is less easily confused by background contours or partial occlusion. The very simple design of the tracker permits fast implementation and supports real-time tracking of multiple objects on a standard workstation.

A.2 Anatomy

Each tracker has a predetermined affine shape, which may be modelled *a priori* or derived from an existing B-Spline snake. It consists of a set of (of the order of 100) sampling points evenly spaced around the predicted contour shape. At each sampling point there is a *local edge-finder* which measures the offset between modelled and actual edge positions in the image, by searching for the maximum of gradient along a short line segment [28].[1] Due to the so-called *aperture problem* [131], only the normal component of this offset can be recovered at any point (figure A.1).

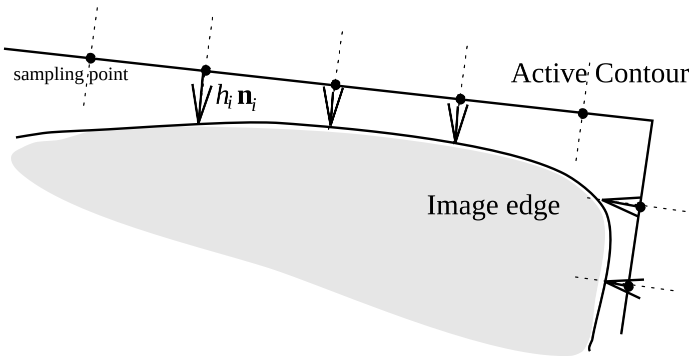


Figure A.1: An active contour: The image is sampled in segments normal to the predicted contour (dotted lines) to search for the maximal gradient. The offsets between predicted and actual edges (arrows) are combined globally to guide the active contour.

The positions of the sampling points are expressed in affine coordinates, and their image positions depend upon the tracker's *local origin* and two *basis vectors*. These are described by six parameters, which change over time as the object is tracked. The contour tangent directions at each point are also described in terms

[1] Two flavours of snake were implemented: 'two-sided' snakes which detect maximal edges without regard to the sign of the gradient, and 'one-sided' snakes which only detect edges with a particular sense. The latter are more robust when tracking bright or dark objects amid clutter.

of the basis vectors, and these are used to calculate the contour normal directions along which the local edge detectors sample the image.

A.3 Dynamics

At each time-step the tracker's parameters are changed, enabling it to move and deform to minimise the sum of squares offset between model and image edges. In our implementation this is done in two stages. First the optimal translation is found, then the deformation, rotation, divergence components are calculated. Splitting the task into these two stages was found to increase stability, as fewer parameters were being estimated at once. To find the optimal translation $\mathbf{t}$ to account for normal offset h_i at each sampling point whose image normal direction is $\mathbf{n}_i$, we solve the following equation:

$$h_i = \mathbf{n}_i \cdot \mathbf{t} + \epsilon_i. \tag{A.1}$$

ϵ_i is the error term, and we solve the whole system of equations using a least-squares method to minimise $\sum \epsilon_i^2$.

Once the translation has been calculated, the other components are estimated. It is assumed that the distortion is centred about the tracker's local origin (normally its centroid, to optimally decouple it from translation). The effects of translation ($\mathbf{n}_i \cdot \mathbf{t}$) are subtracted from each normal offset, leaving a residual offset. We can then find the matrix $\mathbf{A}$ that maps image coordinates to displacement.

$$(h_i - \mathbf{n}_i \cdot \mathbf{t}) = \mathbf{n}_i \cdot (\mathbf{A}\mathbf{p}_i) + \epsilon_i', \tag{A.2}$$

where $\mathbf{p}_i$ is the sampling point's position relative to the local origin and ϵ_i' is again the error term to be minimised.

In practice this formulation can lead to problems when the tracked surface moves whilst partially obscured (often, a tracker will catch on an occluding edge and become 'squashed' as it passes in front of the surface). It can also be unstable and sensitive to noise when the tracker is long and thin. We therefore use a simplified approximation to this equation that ignores the aperture problem (equating the normal component with the whole displacement):

$$(h_i - \mathbf{n}_i \cdot \mathbf{t})\mathbf{n}_i = \mathbf{A}\mathbf{p}_i + \mathbf{e}_i. \tag{A.3}$$

$\mathbf{e}_i$ is a vector, and our implementation solves the equations to minimise $\sum |\mathbf{e}_i|^2$. This produces a more stable tracker that, although sluggish to deform, is well suited to those practical tracking tasks where motion is dominated by the translation component. The tracker positions are updated from **t** and **A** using a real time first-order predictive filter: that is, the velocity of the snake is estimated and used when calculating the next placement of the sampling points, depending on the time delay between iterations. This enhances performance when tracking fast-moving objects.

This *ad hoc* design was found to give good results tracking the robot gripper and hand in real time.

Bibliography

[1] P.K. Allen, A. Timcenko, B. Yoshimi, and P. Michelman. Automated tracking and grasping of a moving object with a robotic hand–eye system. *IEEE Trans. Robotics and Automation*, 9(2):152–165, April 1993.

[2] R.L. Anderson. Dynamic sensing in a ping-pong playing robot. *IEEE Trans. Robotics and Automation*, 5(6):723–739, 1989.

[3] M. Armstrong, A. Zisserman, and R. Hartley. Self-calibration from image triplets. In *Proc. 4th European Conf. Computer Vision*, volume 1, pages 3–16, Cambridge, 1996. Springer LNCS 1064.

[4] R.D. Arnold and T.O. Binford. Geometric constraints in stereo vision. *Proc. SPIE*, 238:281–292, July 1980.

[5] K. Åström, R. Cipolla, and P. J. Giblin. Generalised epipolar constraints. In *Proc. 4th European Conf. Computer Vision*, volume 2, pages 97–118, Cambridge, 1996. Springer LNCS 1065.

[6] N.J. Ayache. *Artificial Vision for Mobile Robots: Stereo Vision and Multisensory Perception.* English translation by P.T. Sander. MIT Press, 1991.

[7] H.H. Baker and T.O. Binford. Depth from edge and intensity based stereo. In *Int. Joint Conf. Artificial Intelligence*, pages 631–636, August 1981.

[8] D.H. Ballard and C.M. Brown. *Computer Vision.* Prentice–Hall, New Jersey, 1982.

[9] T. Baudel and M. Beaudouin-Lafan. Charade: remote control of objects with freehand gestures. *Communications of the ACM*, 36(7):28–35, 1993.

[10] P. Beardsley, P. Torr, and A. Zisserman. 3D model acquisition from extended image sequences. In *Proc. 4th European Conf. Computer Vision*, volume 2, pages 683–695, Cambridge, 1996. Springer LNCS 1065.

[11] A. Blake. Computational modelling of hand-eye coordination. In Y. Aloimonos, editor, *Active Perception.* Academic Press, 1993.

[12] A. Blake, R. Curwen, and A. Zisserman. Affine-invariant contour tracking with automatic control of spatiotemporal scale. In *Proc. 4th Int. Conf. on Computer Vision*, pages 66–75, Berlin, 1993.

[13] R.M. Bolle, A. Califano, R. Kjeldsen, and R.W. Taylor. Visual recognition using concurrent and layered parameter networks. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 625–631, San Diego, Ca., 1989.

[14] J.F. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Analysis and Machine Intell.*, 8:679–698, 1986.

[15] B. Carlisle, S. Roth, J. Gleason, and D. McGhie. The PUMA VS-100 robot vision system. In *Proc. First Int. Conf. Robot Vision and Sensory Controls*, pages 149–160, Kingston upon Hull, 1981.

[16] W.T. Cathey. Imaging and ranging for robot vision. *J. Optical Soc. America 'A'*, 1(12):1247, 1984.

[17] T.-J. Cham and R. Cipolla. Automated B-spline curve representation with MDL-based active contours. In *Proc. British Machine Vision Conf.*, pages 363–372, Edinburgh, 1996.

[18] W.Z. Chen, U.A. Korde, and S.B. Skaar. Position control experiments using vision. *Int. J. Robotics Research*, 13(3):199–208, 1994.

[19] P. Chongstitvatana and A. Conkie. Behaviour based robotic assembly using vision. In C. Archibald and E. Petriu, editors, *Advances in machine vision: strategies and applications.* World Scientific, Singapore, 1992.

[20] R. Chung and R. Nevatia. Use of monocular groupings and occlusion analysis in a hierarchical stereo system. *Computer Vision and Image Understanding*, 62(3):245–268, 1995.

[21] R. Cipolla and A. Blake. Surface orientation and time to contact from image divergence and deformation. In *Proc. 2nd European Conf. Computer Vision*, pages 187–202, Santa Margherita Ligure, Italy, 1992. Springer LNCS 588.

[22] R. Cipolla and A. Blake. Surface shape from the deformation of apparent contours. *Int. Journal of Computer Vision*, 9(2):83–112, 1992.

[23] R. Cipolla and N.J. Hollinghurst. Visual robot guidance from uncalibrated stereo. In C.M. Brown and D. Terzopoulos, editors, *Realtime Computer Vision*, pages 170–184. CUP, 1994.

[24] R. Cipolla and N.J. Hollinghurst. Human–robot interface by pointing with uncalibrated stereo vision. *Image and Vision Computing*, 14(3):171–178, 1996.

[25] R. Cipolla, N.J. Hollinghurst, A.H. Gee, and R.R. Dowland. Computer vision in interactive robotics. *Assembly Automation*, 16(1):18, 1996.

[26] R. Cipolla, Y. Okamoto, and Y. Kuno. Robust structure from motion using motion parallax. In *Proc. 4th Int. Conf. on Computer Vision*, pages 374–382, Berlin, 1993.

[27] C. Colombo and J. L. Crowley. Uncalibrated visual tasks via linear interaction. In *Proc. 4th European Conf. Computer Vision*, volume 2, pages 583–592, Cambridge, 1996. Springer LNCS 1065.

[28] R. Curwen and A. Blake. Dynamic contours: real-time active splines. In A. Blake and A. Yuille, editors, *Active Vision*, pages 39–57. MIT Press, 1992.

[29] T. Darrel and A. Pentland. Space–time gestures. In *IEEE Computer Vision and Pattern Recognition*, pages 335–340, New York, 1993.

[30] R. Deriche. Using Canny's criteria to derive a recursively implemented optimal edge detector. *Int. Journal of Computer Vision*, 1:167–187, 1987.

[31] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE transactions on Robotics and Automation*, 8(3):313–326, 1992.

[32] O. Faugeras. *Three-dimensional computer vision.* MIT Press, 1993.

[33] O. Faugeras. Stratification of 3D vision: projective, affine and metric representations. *J. Opt. Soc. America 'A'*, 12(3):465–484, March 1995.

[34] O. Faugeras and F. Lustman. Motion and structure from motion in a piecewise planar environment. *Int. Journal of Pattern Recognition and Artificial Intelligence*, 2(3):458–508, 1988.

[35] O.D. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig. In *Proc. 2nd European Conf. Computer Vision*, pages 563–578, Santa Margherita Ligure, Italy, 1992. Springer LNCS 588.

[36] O.D. Faugeras and S.J. Maybank. Motion from point matches: multiplicity of solutions. In *IEEE Workshop on Motion*, pages 248–255, Irvine, California., 1989.

[37] O.D. Faugeras and G. Toscani. The calibration problem for stereo. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 15–20, 1986.

[38] M. Fischler and R. Bolles. Random sample consensus. *Graphics and Image Processing*, 24(6):381–395, 1981.

[39] J. Foley, A. van Dam, S. Feiner, and J. Hughes. *Computer Graphics: Principles and Practice.* Addison-Wesley, 1990.

[40] M. Fukumoto, K. Mase, and Y. Suenaga. Realtime detection of pointing actions for a glove-free interface. In *Proc. IAPR Workshop on Machine Vision Applications MVA'92*, pages 473–476, Tokyo, December 1992.

[41] D. Geiger, B. Ladendorf, and A. Yuille. Occlusions and binocular stereo. In *Proc. 2nd European Conf. Computer Vision*, pages 425–433, Santa Margherita Ligure, Italy, 1992. Springer LNCS 588.

[42] A. Gelb. *Applied Optimal Estimation.* MIT Press, 1974.

[43] F. Glover and M. Laguna. Tabu search. In C.R. Reeves, editor, *Modern Heuristic Techniques for Combinatorial Problems*, pages 70–150. Blackwell, Oxford, 1993.

[44] W. E. L. Grimson. Computational experiments with a feature-based stereo algorithm. *IEEE Trans. Pattern Analysis and Machine Intell.*, 7(1):17–34, 1985.

[45] W.E.L. Grimson. *From Images to Surfaces.* MIT Press, Cambridge, USA, 1981.

[46] P. Grossmann. Compact: a surface representation scheme. *Image and Vision Computing*, 7:115–121, 1989.

[47] A. Gueziec and N.J. Ayache. Smoothing and matching of 3-D space curves. *Int. Journal Computer Vision*, 12(1):79–104, February 1994.

[48] G. Hager, S. Hutchinson, and P. Corke. A tutorial introduction to visual servo control. *IEEE Trans. Robotics and Automation*, 12(5):651–670, 1996.

[49] G.D. Hager. A modular system for robust positioning using feedback from stereo vision. Technical report, Department of Computer Science, Yale University, New Haven, CT., 1996. (available at `ftp://ftp.cs.yale.edu/pub/hager/modular.ps.gz`). To appear in *IEEE Trans. Robotics and Automation.*

[50] G.D. Hager, W.-C. Chang, and A.S. Morse. Robot hand–eye coordination based on stereo vision. *IEEE Control Systems Magazine*, 15(1):30–39, February 1995.

[51] E.L. Hall and J.H. Nurre. Robot vision overview. *Proc. SPIE*, 528:216–239, 1985.

[52] G. Hamid, N. Hollinghurst, and R. Cipolla. Identifying planar regions in a scene using uncalibrated stereo vision. In *Proc. British Machine Vision Conf.*, pages 243–252, Edinburgh, 1996.

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

[54] R.C. Harrell, P.D. Adsit, R.D. Munilla, and D.C. Slaughter. Robotic picking of citrus fruit. *Robotica*, 8:269–278, October 1990.

[55] C. Harris. Geometry from visual motion. In A. Blake and A. Yuille, editors, *Active Vision*, pages 263–284. MIT Press, 1992.

[56] C. Harris. Tracking with rigid models. In A. Blake and A. Yuille, editors, *Active Vision*, pages 59–74. MIT Press, 1992.

[57] C. Harris and M.J. Stephens. A combined corner and edge detector. In *Proc. 4th Alvey Vision Conference*, pages 147–152, Manchester, 1988.

[58] J.-Y. Hervé, R. Sharma, and P. Cucka. The geometry of visual coordination. In *Proceedings of the 9th National Conf. Aritficial Intelligence*, Anaheim, California, 1991.

[59] J.-Y. Hervé, R. Sharma, and P. Cucka. Toward robust vision-based control: hand/eye coordination without calibration. In *Proc. IEEE Int. Symposium on Intelligent Control*, Arlington, Virginia, 1991.

[60] W. Hoff and N. Ahuja. Extracting surfaces from stereo images: an integrated approach. In *Int. Conf Computer Vision ICCV'87*, pages 284–294, London, June 1987.

[61] N. Hollinghurst and R. Cipolla. Uncalibrated stereo hand–eye coordination. *Image and Vision Computing*, 12(3):187–192, 1994.

[62] K. Ikeuchi, H.K. Nishihara, B.K.P. Horn, P. Sobalvarro, and S. Nagata. Determining grasp configurations using photometric stereo and the PRISM binocular stereo system. *Int. Journal of Robotics Research*, 5(1):47–65, 1986.

[63] M. Jägersrand, O. Fuentes, and R. Nelson. Acquiring visual-motor models for precision manipulation with robot hands. In *Proc. 4th European Conf. Computer Vision*, volume 2, pages 603–612, Cambridge, 1996. Springer LNCS 1065.

[64] M. Kass, A.P. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Int. J. Computer Vision*, 1(4):321–331, January 1988.

[65] D. Knuth. Sorting and searching. In *The Art of Computer Programming.* Addison-Wesley, 1975.

[66] Donald E. Knuth, Rajeev Motwani, and Boris Pittel. Stable husbands. In *Proceedings of the First Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 397–404, San Francisco, California, January 1990.

[67] J.J. Koenderink. Optic flow. *Vision Research*, 26(1):161–179, 1986.

[68] J.J. Koenderink and A.J. van Doorn. Affine structure from motion. *J. Opt. Soc. America*, A8(2):377–385, 1991.

[69] J. Kuch and T. Huang. Virtual gun: a vision based human computer interface using the human hand. In *Proc. IAPR Workshop on Machine Vision Applications MVA'94*, Tokyo, December 1994.

[70] J.M. Lawn and R. Cipolla. Reliable extraction of the camera motion using constraints on the epipole. In *Proc. 4th European Conf. Computer Vision*, volume 2, pages 161–173, Cambridge, 1996. Springer LNCS 1065.

[71] F. Li, J.M. Brady, and C. Wiles. Fast computation of the fundamental matrix for an active stereo vision system. In *Proc. 4th European Conf. Computer Vision*, volume 1, pages 157–166, Cambridge, 1996. Springer LNCS 1064.

[72] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981.

[73] H.C. Longuet-Higgins. The visual ambiguity of a moving plane. *Proc. R. Soc. Lond.*, B223:165–175, 1984.

[74] D. Lowe. *Perceptual Organisation and Visual Recognition.* Kluwer Academic, Boston, 1985.

[75] T. Lozano-Perez, J.L. Jones, E. Mazer, and P.A. O'Donnell. HANDEY*: A Robot Task Planner.* MIT Press, 1992.

[76] Q.-T. Luong and O.D. Faugeras. The fundamental matrix: Theory, algorithms, and stability analysis. *Int. J. Computer Vision*, 17(1):43–75, January 1996.

[77] C.A. Malcolm and T. Smithers. Programming assembly robots in terms of task achieving behavioural modules. In *Proc. Advanced Robotics Program 2nd Workshop on Manipulators, Sensors and Steps towards Mobility*, pages 15.1–15.16, Manchester, UK, October 1988.

[78] J. Malik. On binocularly viewed occlusion junctions. In *Proc. 4th European Conf. Computer Vision*, volume 1, pages 167–174, Cambridge, 1996. Springer LNCS 1064.

[79] X. Markenscoff, L. Ni, and C.H Papadimitrou. The geometry of grasping. *Int. Journal of Robotics Research*, 9(1):61–74, 1990.

[80] D. Marr. *Vision.* Freeman, San Francisco, 1982.

[81] D. Marr and E. Hildreth. Theory of edge detection. *Proc. Roy. Soc. London. B.*, 207:187–217, 1980.

[82] D. Marr and T. Poggio. Cooperative computation of stereo disparity. *Science*, vol.194:283–287, 1976.

[83] P.F. McLauchlan, J.E.W. Mayhew, and J.P. Frisby. Stereoscopic recovery and description of smooth textured surfaces. *Image and Vision Computing*, 9(1):20–26, 1991.

[84] G. Medioni and R. Nevatia. Segment-based stereo matching. *Computer Vision, Graphics and Image Processing*, 31(1):2–18, 1985.

[85] B. W. Mel. MURPHY: a robot that learns by doing. In *AIP Proc. 1987 Neural Information Processing Systems Conf.*, Denver, Colorado, 1987.

[86] B.W. Mel. *Connectionist Robot Motion Planning.* Academic Press, San Diego, 1990.

[87] R. Mohan and R. Nevatia. Perceptual organisation for scene segmentation and description. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(6):616–634, 1992.

[88] J.L. Mundy and A. Zisserman, editors. *Geometric Invariance in Computer Vision.* MIT Press, 1992.

[89] D.W. Murray, F. Du, P.F. McLauchlan, I.D. Reid, P. Sharkey, and M. Brady. Design of stereo heads. In A. Blake and A. Yuille, editors, *Active Vision*, pages 155–172. MIT Press, 1992.

[90] B. Nelson and P.K. Khosla. Integrating sensor placement and visual tracking strategies. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 1351–1356, 1994.

[91] R. Nevatia. *Machine Perception.* Prentice-Hall, 1982.

[92] V.-D. Nguyen. Constructing force-closure grasps. *Int. Journal of Robotics Research*, 7(3):3–16, 1988.

[93] H. K. Nishihara. PRISM: a practical real-time imaging stereo matcher. *Optical Engineering*, 23(5):536–545, September 1984.

[94] J. A. Noble. Finding corners. *Image and Vision Computing*, 6(2):121–128, May 1988.

[95] Y. Ohta and T. Kanade. Stereo by intra- and inter-scan line search using dynamic programming. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 7(2):139–154, 1985.

[96] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990.

[97] C.U. Petersson. An integrated robot vision system for industrial use. *Proc. SPIE*, 449(1):305–311, 1983.

[98] G.F. Poggio and T. Poggio. The analysis of stereopsis. *Annual Review of Neuroscience*, 7:379–412, 1984.

[99] S. B. Pollard, J. Porrill, J. E. W. Mayhew, and J. P. Frisby. Matching geometrical descriptions in 3-space. *Image and Vision Computing*, 5(2):73–78, 1987.

[100] S.B. Pollard, J.E.W. Mayhew, and J.P. Frisby. PMF: A Stereo Correspondence Algorithm Using A Disparity Gradient. *Perception*, 14:449–470, 1985.

[101] S.B. Pollard, J. Porrill, and J.E.W. Mayhew. Recovering partial 3D wire frame descriptions from stereo data. *Image and Vision Computing*, 9(1):58–65, 1991.

[102] J. Porrill and S. Pollard. Curve matching and stereo calibration. *Image and Vision Computing*, 9:45–50, 1991.

[103] J. Porrill, S. B. Pollard, T. P. Pridmore, J. B. Bowen, J. E. W. Mayhew, and J. P. Frisby. TINA: a 3D vision system for pick and place. *Image and Vision Computing*, 6(2):91–99, 1988.

[104] L.H. Quam. Hierarchical warp stereo. In *DARPA Image Understanding Workshop*, pages 149–155, New Orleans, October 1984.

[105] L. Quan. Affine stereo calibration for relative affine shape reconstruction. In *Proc. British Machine Vision Conf.*, volume 2, pages 659–668, Guildford, 1995.

[106] L. Quan and R. Mohr. Towards structure from motion for linear features through reference points. In *Proc. IEEE workshop on visual motion*, pages 249–254, New Jersey, 1991.

[107] A.A. Rizzi and D.E. Koditschek. An active visual estimator for dextrous manipulation. *IEEE Trans. Robotics and Automation*, 12(5):697–713, 1996.

[108] L.G. Roberts. Machine perception of three - dimensional solids. In J.T. Tippet, editor, *Optical and Electro-optical Information Processing.* MIT Press, 1965.

[109] C.A Rothwell. *Object Recognition through Invariant Indexing.* Oxford University Press, 1995.

[110] P.J. Rousseeuw. *Robust Regression and Outlier Detection.* Wiley, New York, 1987.

[111] M. Rutishauser and M. Stricker. Searching for grasping opportunities on unmodelled 3D objects. In *Proc. British Machine Vision Conf.*, pages 277–286, Birmingham, 1995.

[112] M. Rygol, S. Pollard, and C. Brown. A multiprocessor 3D vision system for pick-and-place. In *Proc. British Machine Vision Conference*, pages 169–174, 1990.

[113] J. Sato and R. Cipolla. Image registration using multi-scale texture moments. *Image and Vision Computing*, 13(5):341–353, 1995.

[114] J. Sato and R. Cipolla. Affine integral invariants and matching of curves. In *Proc. International Conference on Pattern Recognition*, volume 1, pages 915–919, Vienna, August 1996.

[115] L.S. Shapiro, A. Zisserman, and M. Brady. 3D motion recovery via affine epipolar geometry. *Int. J. Computer Vision*, 16(2):147–182, October 1995.

[116] Y. Shirai. Robot vision. *Future Generation Computer Systems*, 1(5):325–352, September 1985.

[117] Y. Shirai and H. Inoue. Guiding a robot by visual feedback in assembling tasks. *Pattern Recognition*, 5(2):99–108, 1973.

[118] D. Sinclair and A. Blake. Quantitative planar region detection. *Int. Journal of Computer Vision*, 18(1):77–91, 1996.

[119] S.B. Skaar, W.H. Brockman, and W.S. Jang. Three-dimensional camera space manipulation. *Int. Journal of Robotics Research*, 9(4):22–39, August 1990.

[120] M.W. Spong and M. Vidyasagar. *Robot Dynamics and Control.* Wiley, 1989.

[121] M. Spratling and R. Cipolla. Uncalibrated visual servoing. In *Proc. British Machine Vision Conf.*, volume 2, pages 545–554, Edinburgh, 1996.

[122] G. Strang. *Linear algebra and its applications.* Harcourt Brace Jovanivich, 1988.

[123] D. Sturman, D. Zelter, and S. Pieper. Hands-on interaction with virtual environments. In *UIST: Proc. ACM SIGGRAPH Symposium on User Interfaces*, pages 19–24, Williamsburg, Virginia, November 1989.

[124] K. Tarabanis, R.Y. Tsai, and D.S. Goodman. Calibration of a computer controlled robotic vision system with a zoom lens. *CVGIP Image Understanding*, 59(2):226–241, 1994.

[125] M.J. Taylor and A. Blake. Grasping the apparent contour. In *Proc. 3rd European Conf. Computer Vision*, volume 1, pages 25–34, Stockholm, 1994. Springer LNCS 800.

[126] C. Tomasi and R. Manduchi. Stereo without search. In *Proc. 4th European Conf. Computer Vision*, volume 2, pages 452–465, Cambridge, 1996. Springer LNCS 1065.

[127] P. Torr. *Motion segmentation and outlier detection.* PhD thesis, Department of Engineering Science, University of Oxford, 1995.

[128] R.Y. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal of Robotics and Automation*, RA-3(4):323–344, 1987.

[129] R.Y. Tsai and R.K. Lenz. A new technique for fully autonomous and efficient 3D robotics hand-eye calibration. In *4th International Symposium on Robotics Research*, volume 4, pages 287–297, 1987.

[130] R.Y. Tsai and R.K. Lenz. Techniques for calibration of the scale factor and image center for high accuracy 3D machine vision metrology. *IEEE Trans. Pattern Analysis and Machine Intell.*, 10(5):713–720, 1988.

[131] S. Ullman. *The interpretation of visual motion.* MIT Press, Cambridge, USA, 1979.

[132] H. Wang and M. Brady. Real-time corner detection algorithm for motion estimation. *Image and Vision Computing*, 13(9):695–703, November 1995.

[133] L.E. Weiss, A.C. Sanderson, and C.P. Neuman. Dynamic sensor-based control of robots with visual feedback. *IEEE Trans. Robotics and Automation*, 3(5):404–417, 1987.

[134] S. W. Wijesoma, D. F. H. Wolfe, and R. J. Richards. Eye-to-hand coordination for vision-guided robot control applications. *Int. J. Robotics Research*, 12(1):65–78, 1993.

[135] B. Wirtz and C. Maggioni. Imageglove: a novel way to control virtual environments. In *Proc. Virtual Reality Systems*, New York City, April 1993.

[136] B. Yoshimi and P.K. Allen. Visual control of grasping and manipulation tasks. In *Proc. IEEE Int. Conference on Multi-Sensor Fusion*, Las Vegas, October 1995.

[137] Z. Zhang. Estimating motion and structure from correspondences of line segments between two perspective images. *IEEE Trans. Pattern Analysis and Machine Intell.*, 17(12):1129–1139, 1995.

[138] Z. Zhang, R. Deriche, O. Faugeras, and L. Quan. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. Technical Report 2273, INRIA, Sophia-Antipolis, France, 1994. (available from `http://www.inria.fr/robotvis/`).