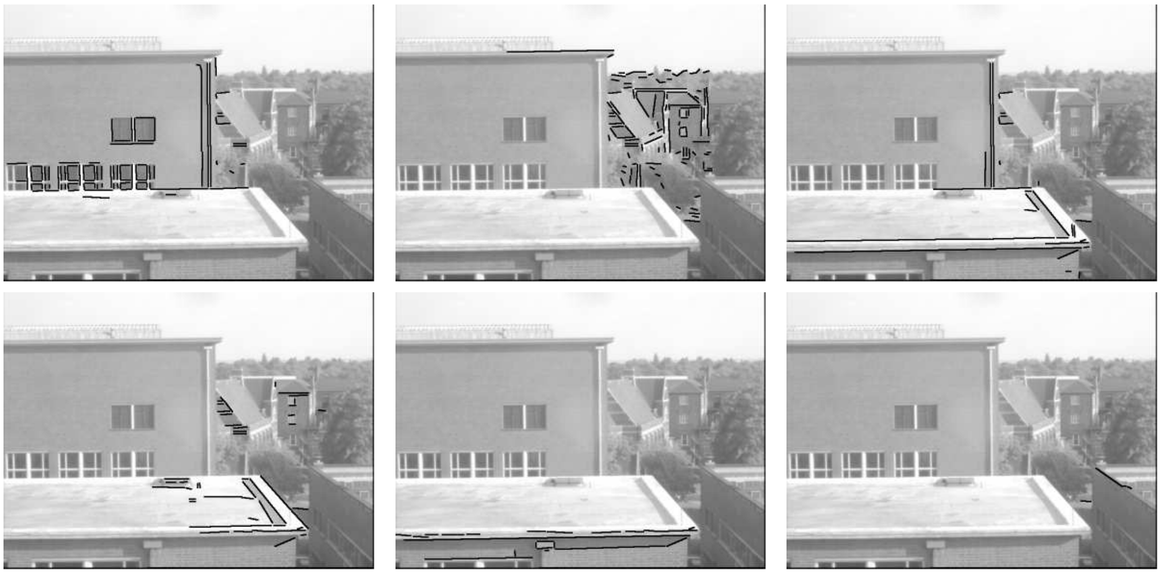


Figure 5.15: Planes recovered in the `roof` scene

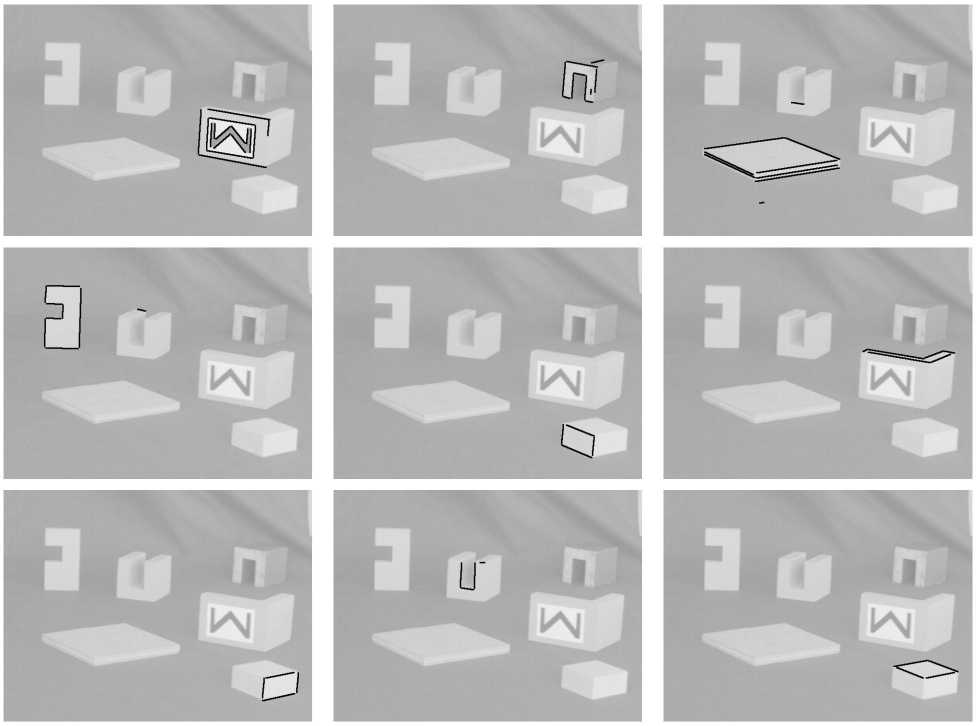


Figure 5.16: Planes recovered in the `blocks` scene

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.

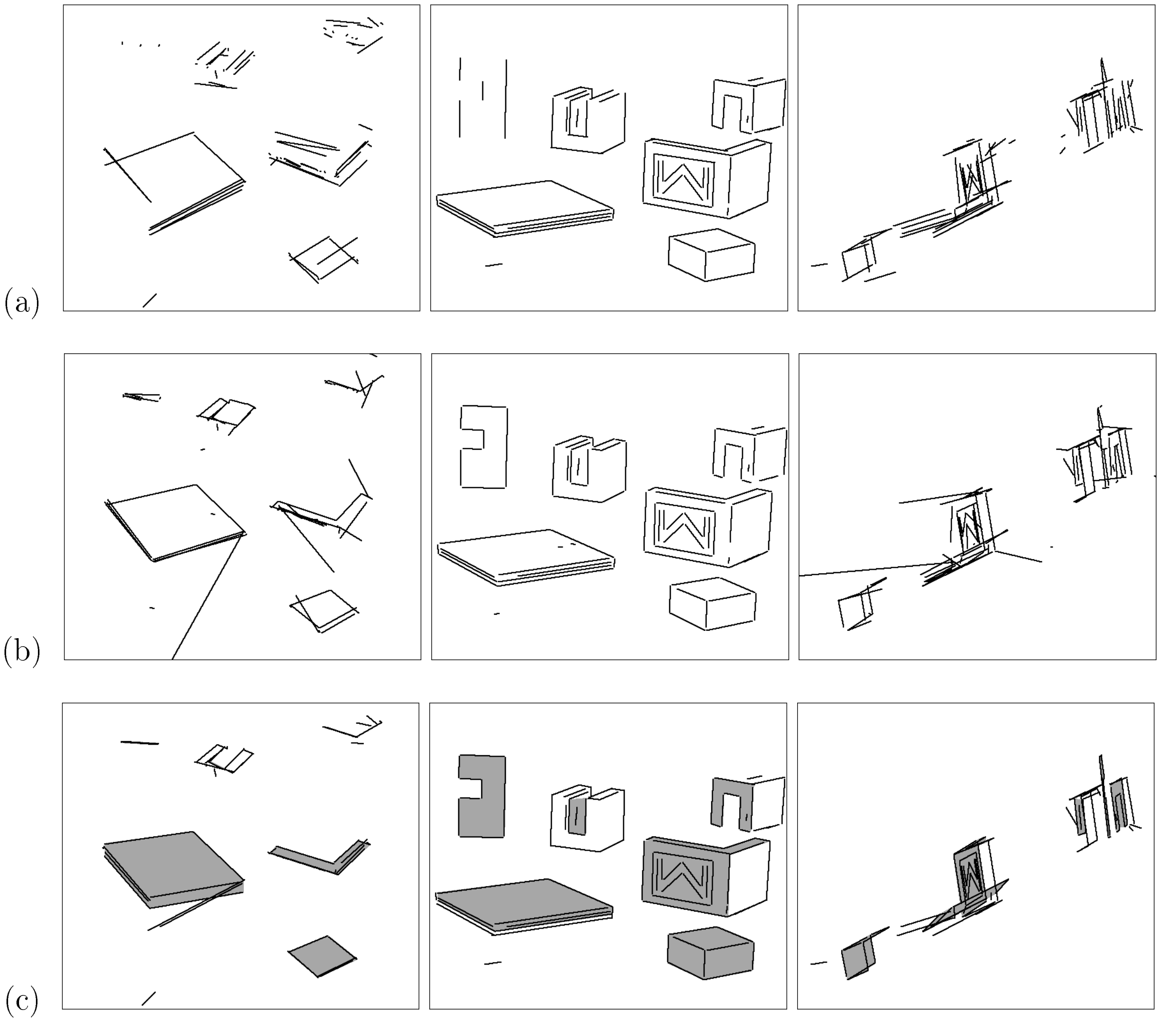
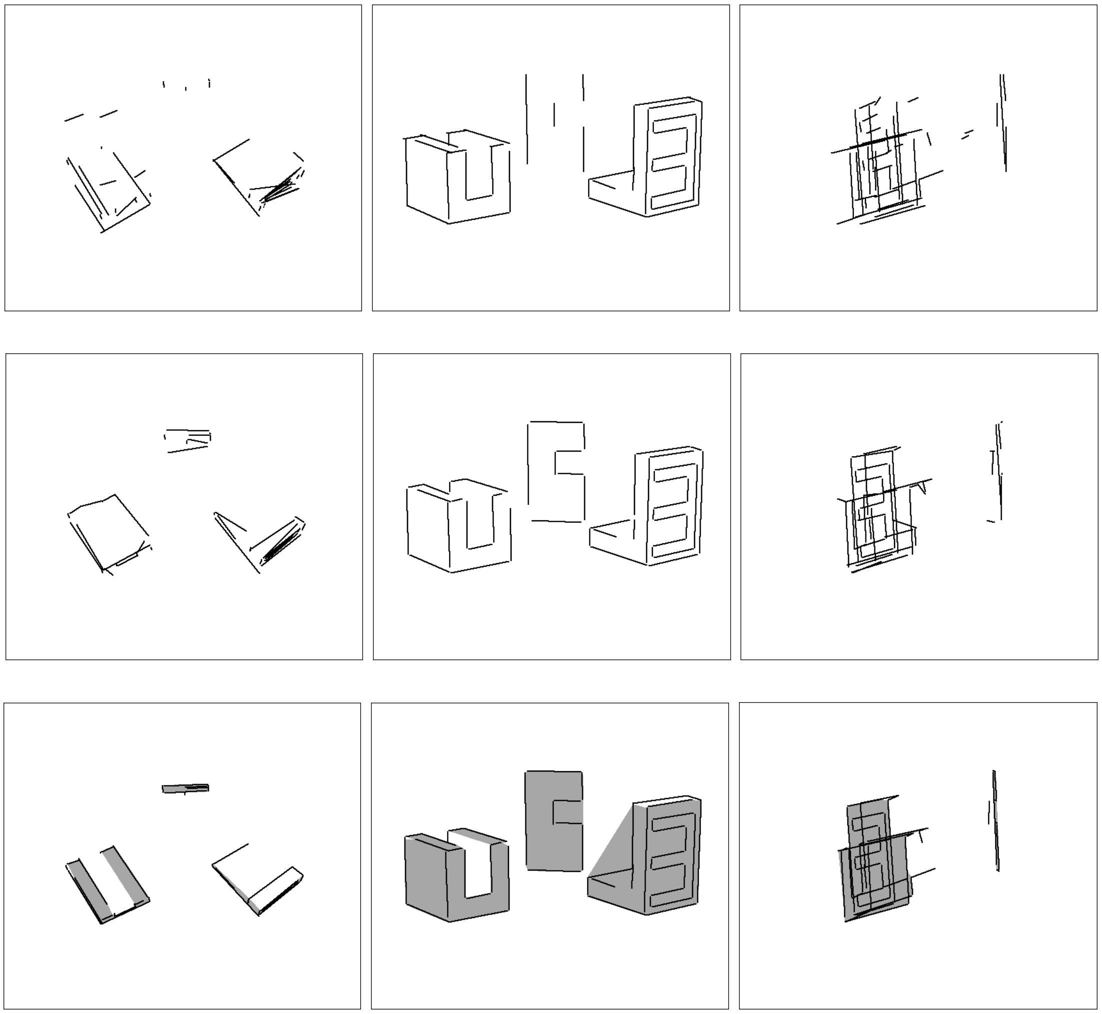


Figure 5.17: Reconstruction of the `blocks` scene illustrated by synthetic top, cyclopean and side views, using: (a) epipolar matching only, (b) corresponding endpoints, (c) facet grouping and junctions.



(a)

(b)

(c)

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.

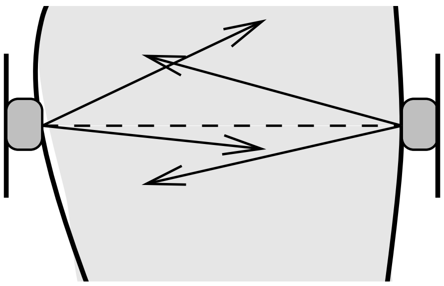


Figure 6.1: Grasping with a parallel gripper: the grasp is successful if the line joining the contacts lies within the *friction cone* of each (i.e. the contact surfaces are nearly orthogonal to the gripper's axis).

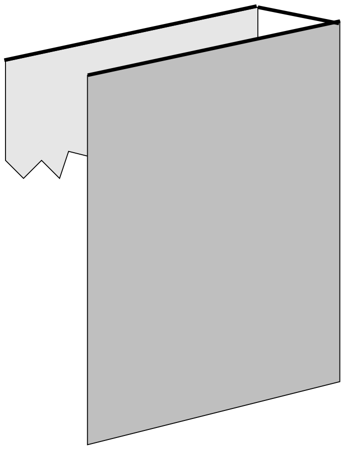


Figure 6.2: Inference of parallel surfaces from parallel edges: if two edge segments on a 3-D connected component are parallel, one lies on a visible surface boundary (dark grey) and the other is behind it, we hypothesise that there is a parallel surface (light grey).

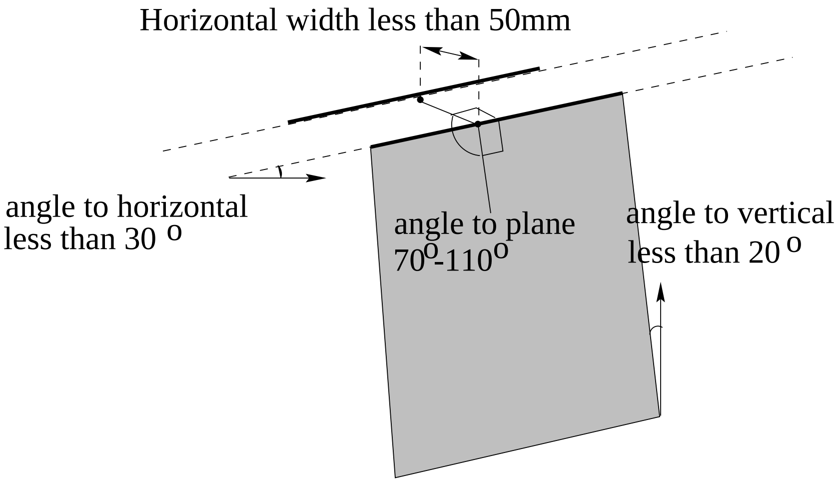


Figure 6.3: Summary of the conditions to be met by a pair of points on a facet and edge pair, to be considered a feasible grasp for our robot.

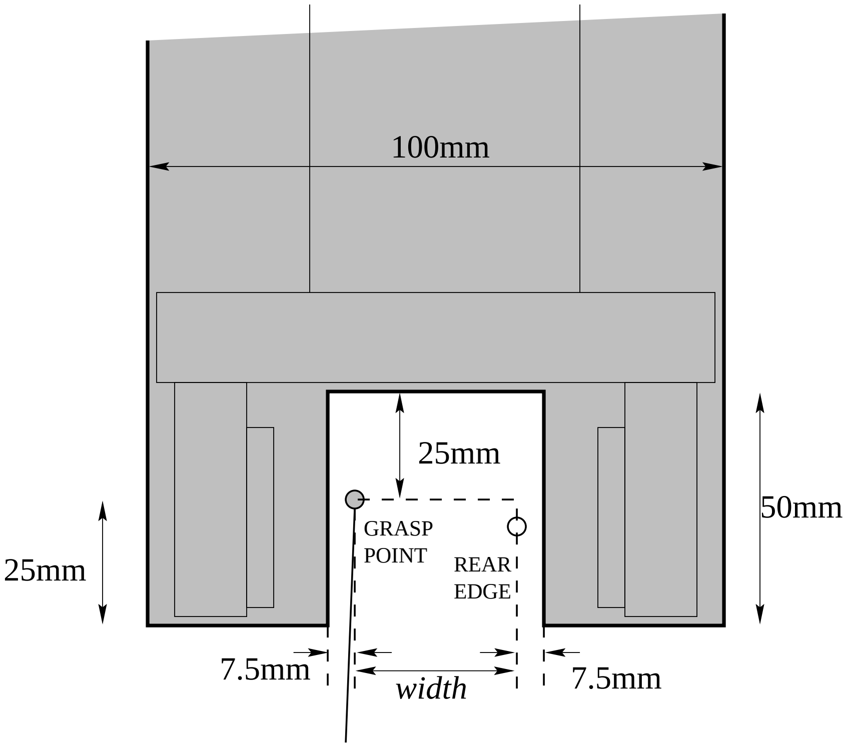


Figure 6.4: Cross section of the robot gripper model used for collision checking. The gripper approaches vertically from above. No visible features may intersect the shaded region.

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.

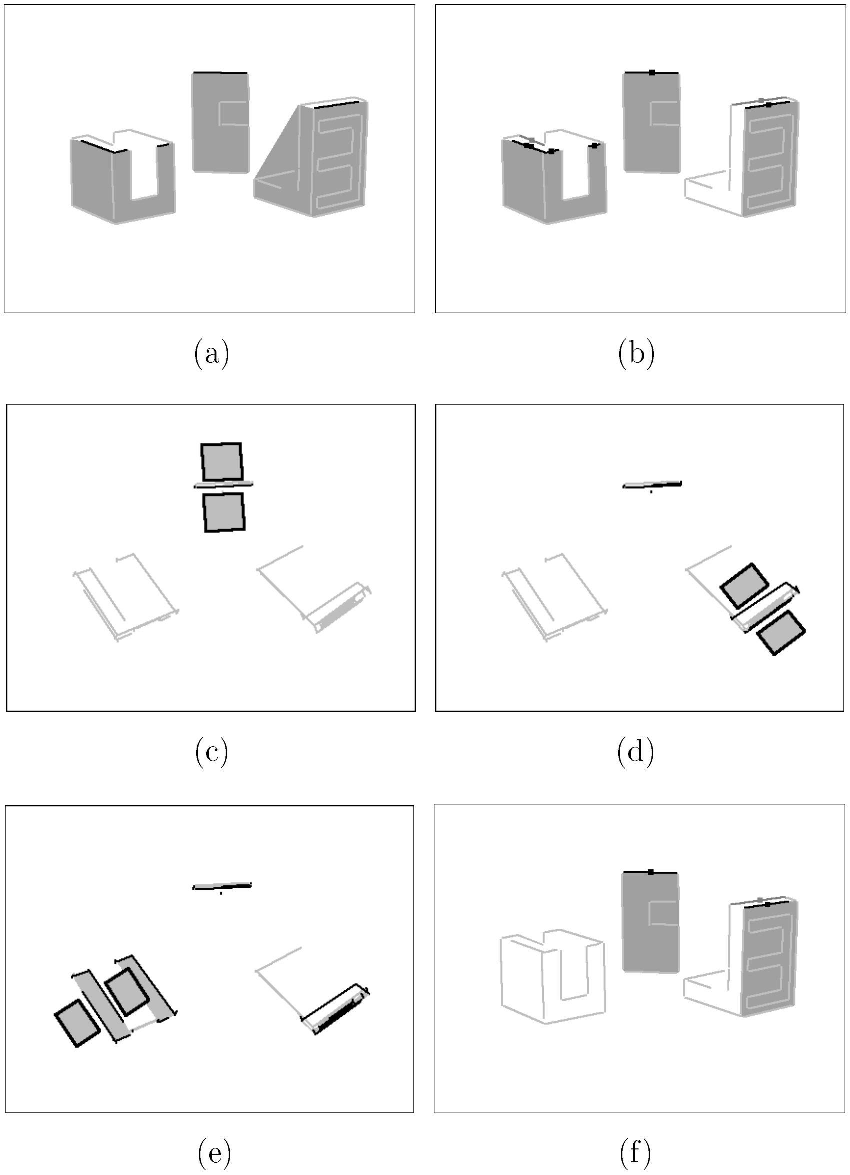


Figure 6.5: Parallel grasp planning on stereo reconstruction of the `test` scene: (a) near-vertical facets and near-horizontal upper edges; (b) feasible grasp sites; (c,d) plan views of successful grasp sites showing finger placements; (e) grasp site where finger would collide with object; (f) collision-free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

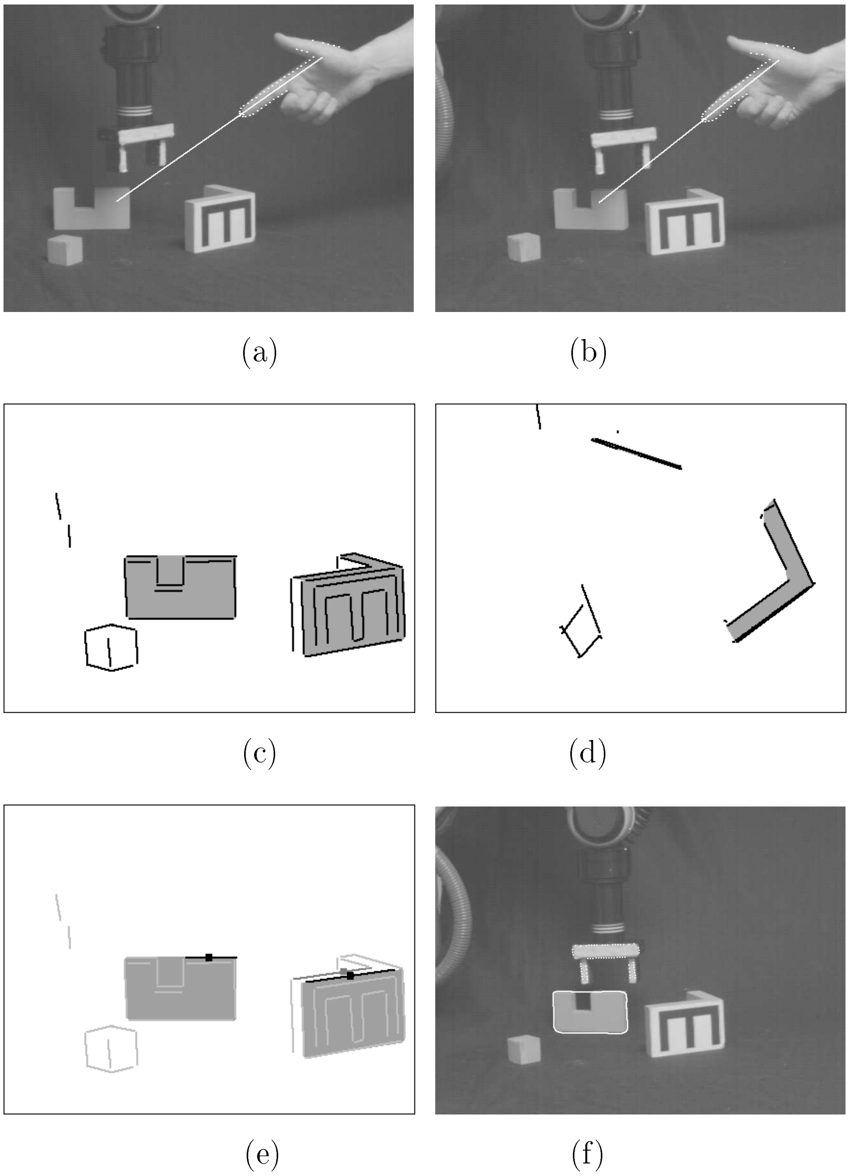


Figure 6.6: Visually guided grasping of an object: (a,b) stereo views of a pointing hand indicating the object to be grasped; (c,d) reconstruction of the scene; (e) permissible grasp sites; (f) alignment using visual feedback.

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).

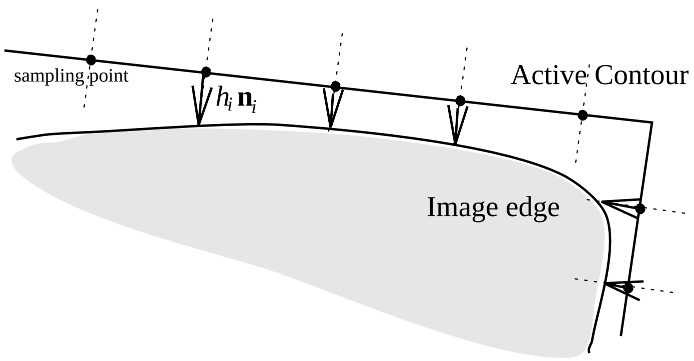


Figure A.1: An active contour: The image is sampled in segments normal to the predicted contour (dotted lines) to search for the maximal gradient. The offsets between predicted and actual edges (arrows) are combined globally to guide the active contour.

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 $\mathbf{t}$ and $\mathbf{A}$ 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/`).