Behaviours for Active Object Recognition

David Wilkes and John Tsotsos
Department of Computer Science
University of Toronto
6 King's College Rd.
Toronto, Ontario
Canada M5S 1A4
E-mail: wilkes@vis.toronto.edu

Abstract

The concept of active object recognition is introduced and a proposal for its solution is described. It is argued that single-view object recognition is fraught with problems, mainly due to viewpoint-related ambiguities, occlusions and coincidences. Recognition which is *active*, that is, that has the ability to vary viewpoint according to the interpretation status, is both more viable and more closely related to the manner in which humans recognize different objects. An active system may be achieved via a simple modification of the recognition hardware. The camera is mounted on the end of a robot arm on a mobile base. The system exploits the mobility of the camera by using low-level image data to drive the camera to a special viewpoint with respect to an unknown object. From such a viewpoint, the object recognition task is reduced to a two-dimensional pattern recognition problem. This paper describes the behaviour-based approach to camera motion, that ensures robust acquisition of special views of the object to be recognized.

1 Introduction

Current work in 3D object recognition has focussed primarily on the problem of recognizing objects from single, arbitrary views. The fact that there has been work on this problem for many years attests to its difficulty. There are several reasons for the difficulty:

Impossibility of inverting projection The first and most serious problem is that the effects of projection of a three-dimensional world onto two dimensions are irreversible. The three-dimensional structure of the world can not be recovered from a single two-dimensional image without making very restrictive assumptions about the world being imaged.

This means that the system must explicitly account for the effects of projection, given any possibility of variation in camera viewpoint.

Occlusion The second problem is that some of the object's features will be invisible from the viewpoint used to collect the sensor data. Some features will inevitably be occluded by the object itself. Others may be occluded by other objects between the object of interest and the camera.

Detectibility Features may also be missing due to low detectibility. For computer vision, the primary reason is low contrast between scene parts due to the interaction between the existing illumination and surface reflectance and geometry.

False features may be detected due to peculiarities of lighting. The most obvious examples are boundaries between illuminated and shadow areas.

In addition, object features may be missing because they are out of the field of view of the camera.

0277-786X/93/$6.00

Fragility of 3-D inference A variation on the theme of using a single camera viewpoint for recognition is to use two slightly different views, either simultaneously or sequentially. The idea is to reconstruct some aspect of the three dimensional structure of objects given the disparities between the two views.

With two viewpoints similar enough to allow reliable correspondences to be established between the two images, the process of inferring three-dimensional structure is very sensitive to errors in the two-dimensional measurements in each image.

View degeneracy Features may be difficult to identify due to the accidental alignment of spatially distinct scene parts from the camera viewpoint. Such alignment is often referred to as *view degeneracy*. The aligned parts may be part of the same object or different objects. The superposition of features that results can lead to erroneous part counts or bad parameterizations of a model. For example, if two object edges are abutted and collinear, one longer edge is seen in their place.

Given arbitrarily high operator resolution, such alignments would not be a a problem, because they would occur only for a vanishingly small fraction of the possible viewpoints.

Unfortunately, our cameras and feature-extracting operators have finite resolution. A quantitative model of the effects of finite resolution on the probability of view degeneracy appears in [7] and [8]. For our system, the model suggests that an arbitrary view of one of the objects to be recognized has a probability greater than one in three of containing significant degeneracies.

An additional practical problem is that low-level vision modules do not typically make use of all of the information in the image that is available for segmentation of the image into primitives. This aggravates the problems of detectibility and view degeneracy discussed above. For example, features that could in principle be recognized as being alignments of two scene features may not be distinguished in practice. Following the example of abutted collinear lines, the system may in principle have been able to distinguish the lines from each other on the basis of the shading or texture of the surfaces adjacent to each. In practice, such information may well have been discarded as an early step in the line-finding process.

1.1 Benefits of a mobile camera and light source

Bajscy introduced the concept of active perception. She describes the problem as "a problem of intelligent control strategies applied to the data acquisition process" [1]. We propose to apply this concept to the task of object recognition, by changing viewpoint. We recapitulate the problems discussed above, in the context of the new paradigm.

Impossibility of inverting projection Rather than attempting to invert the projection process, we may change the viewpoint on the object to one for which the projection is precisely known and unambiguous.

Occlusion Many instances of occlusion can be eliminated by moving the camera to look around the occluding object. Certain instances of self-occlusion and containment can not be eliminated. For example, the underside of an object resting on the ground can not be seen without moving the object. An object enclosed in a second object can not be seen without opening the second object somehow.

Detectibility Low contrast on occluding contours of objects may possibly be improved by changing the viewpoint. This is the case if the relative reflectance of the surfaces behind the object and composing the object can be changed by a change in viewpoint.

Low contrast and shadow effects may also be overcome by keeping the principal source of illumination near the camera. This serves to throw the darkest shadows behind the imaged objects. It also provides a natural gradient in illumination level with increasing distance from the camera, frequently improving the constrast of lines on an object's occluding contour.

Fragility of 3-D inference We shall see below that a behaviour-based system for varying viewpoint can be constructed that makes very little use of 3-D structure inferred from 2-D images. As a result, the fragility of 3-D inference need not be an issue for an active approach.

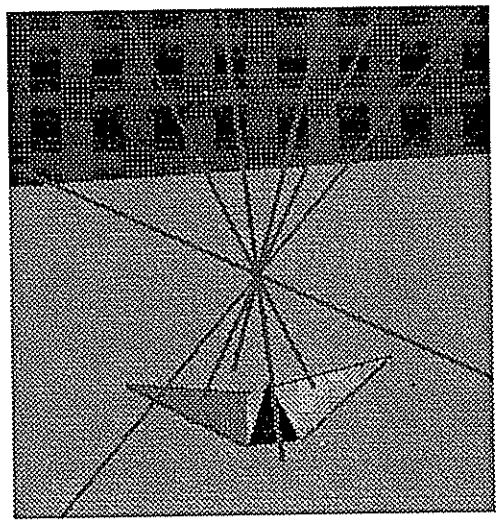

Figure 1: The special views for a particular object, based on the four most prominent feature point pairs (the endpoints of the dark object lines). The straight lines show the directions of each view. The point on such a line, that gives a particular image separation s to the first defining feature pair for the line, is the special viewpoint.

View degeneracy With a changing viewpoint and moderate feature extraction operator size, accidental alignments do not persist over time. Transient accidental alignments could be problematic for an approach that varies viewpoint, if they have an impact on tracking of image features from one frame to another. Given a robust feature tracking system, that can survive occasional bad features that result from transient accidental alignments, such alignments cease to be a problem.

1.2 Overview of Active Object Recognition

Our approach, active object recognition, divides the recognition problem into two independent subtasks, each of which can be accomplished more robustly than the original task.

The first subtask deals with changes in illumination and imaging geometry by changing the viewpoint to achieve a position called a *special view* of the unknown object. For each object, several special views exist, as shown in figure 1. The motion to a special view is driven by low-level image data. In general, we define a special view to be a position optimizing some function f of the features extracted from the image data and the current position. For example, our implementation defines a special view to be a position maximizing the apparent separation in each of two pairs of feature points, with the pairs possibly sharing one feature point, over all positions at a fixed distance from the object (the distance is that at which the first feature-point pair's separation has a target value, as seen from the special view orientation). Moving to such a viewpoint has much in common with robotic docking maneuvers that maintain a certain bearing with respect to an object, based on low-level image data (Wunsche, as reported by Dickmanns and Graefe [4]). Also related is the interesting work in 2D-data-driven control by Zheng et al. [9].

In order for the viewpoint change to be carried out, the difficult problem of correspondence of image primitives between frames must be solved. The problem is complicated by the possibility of accidental alignments of tracked primitives, infrequent imaging of the object and large changes in viewpoint between frames. We present a tracking method robust to these problems in [8]. The method allows arbitrary rotations of the camera between frames, and a high degree of variability in the individual primitives extracted from the image data. Our approach is to describe each image primitive using the set of surrounding primitives, expressed in local, primitive-based coordinates. The resulting tokens are described by a mixture of simple primitive-generating probability distributions.

Given a successful strategy for moving to a special view, we are left with a two-dimensional recognition problem, with remaining variation in object appearance due to two main factors.

The first factor is discussed in depth in [8]. It is the effect of the relative positions of other objects, including the background and parts of the object to be recognized. This affects the detectability of various object features, causing certain object features to be missed, or grossly distorted (as is the case with a partially occluded line). Features that in fact belong to other objects may be erroneously grouped with features of the desired object, resulting in the

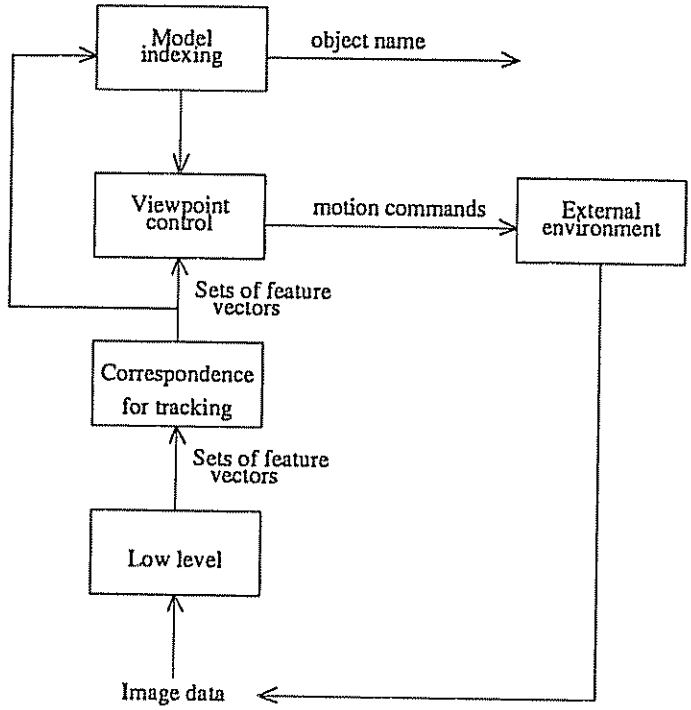

Figure 2: High-level view of the system

apparent "invention" of new features.

The second factor is error in the acquisition of a special view. This can be due to error in both camera and light source positioning, and also due to the presence of secondary, uncontrolled light sources. Positioning errors result primarily in minor perturbations of feature parameter values. Lighting variations can result in missing features.

In order to deal with these remaining sources of variation, we present a method of storing and probabilistically retrieving objects based on *noisy sets* of features in [8].

Figure 2 recapitulates our approach. Starting with image data, a low level image processing module extracts multi-parameter features of a type chosen to be appropriate for the application domain. A set of conditions on the features is chosen that defines a particular position of the camera with respect to the object. The system drives either the camera or the object or both to positions that achieve this viewpoint, restarting with a different set of conditions in case of failure. For small, movable objects, it might be appropriate to move the object rather than the camera. From the first successfully achieved special view, the object is recognized by matching the feature set extracted from the image at this position with stored feature sets, using an indexing scheme.

2 Theory

In order for our approach to recognition to make sense, we must be able to solve our subproblem of motion to a special view extremely robustly. In the theory section of this paper, we will argue for a solution involving a set of interleaved, simple, data-driven behaviours, and show how this is applied to our particular definition of a special view. The implementation section gives details of the approach; then, we describe the performance of the motion subsystem in various trials.

3 Theory

3.1 Special views and how to get there

The design of a viewpoint control scheme must satisfy a number of requirements.

First, our definition of a special view must be such that only a small number of discrete points surrounding each object are special viewpoints for the sensor. There should be few enough points that it is not problematic to store the projection of the object at each point. On the other hand, there must be enough special views that there is a high probability that at least one will be accessible in spite of occlusion of certain parts of the object by the surface on which it rests, and possibly by other objects. Also, the special views should be such that there is high probability that view degeneracy will not interfere with recognition from the special view.

Second, we must be able to specify a robust method for driving the sensor to a special view from any position within sight of the object. In general, this will involve translation of the sensor through a 3D coordinate space defined by the relevant object features. A particular special view may be inaccessible whenever the object features defining it are not detected, due to occlusion or other problems. Thus, the system must be able to seek alternate special views in case of a failure to acquire a particular chosen view.

It is important to note that it is *not* possible to determine the sensor position of a special view with respect to an object based on an arbitrary single image, for strictly projective sensors such as cameras. This follows directly from work by Burns *et al.* [2] and Clemens and Jacobs [3] concerning projective invariants. Burns *et al.* have shown in particular that there are no sets of scalar quantities that are invariant for all projections of a general 3D set of points into an image. If the coordinates of a special viewpoint could be recovered from an arbitrary single image of an object, they would be such an invariant.

Thus, the special view will need to be recovered from a sequence of two or more images.

The motion strategy must include some means of keeping tracked object features within the view of the sensor, and at a range appropriate for their continued detection. There are two possible approaches to achieving this. One approach would use knowledge of the trajectory of the sensor with respect to the tracked features to re-aim the sensor at the features after each quantum of sensor translation.

A difficulty with this is that knowledge of the 3D scene geometry is also required. This knowledge is only available after a significant amount of camera translation has already occurred.

The safest alternative is to begin motion by centering the midpoint between two relevant object features, then moving away from the centred point to estimate the object distance. If the distance to the object is large relative to the distance between the features, then the distance to the features may be estimated as in figure 3.

We replace the further feature point by its projection onto a plane perpendicular to the optical axis. Then

$$\frac{D}{d_0} = \frac{r_0}{f} \tag{1}$$

and

$$\frac{D}{d_1} = \frac{r_1}{f} \tag{2}$$

Thus,

$$\frac{D}{d_1} - \frac{D}{d_0} = \frac{r_1}{f} - \frac{r_0}{f} \tag{3}$$

leading to

$$D = \frac{d_1 d_0 (r_1 - r_0)}{f(d_0 - d_1)} \tag{4}$$

and the current distance estimate r_1 is available from equation 2.

Then the relevant features may be guaranteed to remain within the image following a translation of the sensor. The idea is to start with the features centred in the image and a known minimum distance from a feature to the image border. The size of the camera translation is then chosen to guarantee an upper bound on the translation of the features in the image that is less than the allowable image distance. The required computations are similar to the above.

So far, our arguments have specified a requirement for a feature-centering operation, and constrained the camera motion to preserve approximately the distance of the chosen features from the camera. This corresponds to restricting sensor motion to lie on a sphere at a fixed distance from the object features of interest. To specify the rest of the viewpoint control requires that we choose a particular definition of a special view.

We would like our definition of a special view to use as few object points as possible, so that it does not depend on overly sophisticated feature detection.

As discussed previously, it is also important to avoid positions of view degeneracy. The best that we can do in this regard is to avoid degeneracy with respect to the view-defining features.

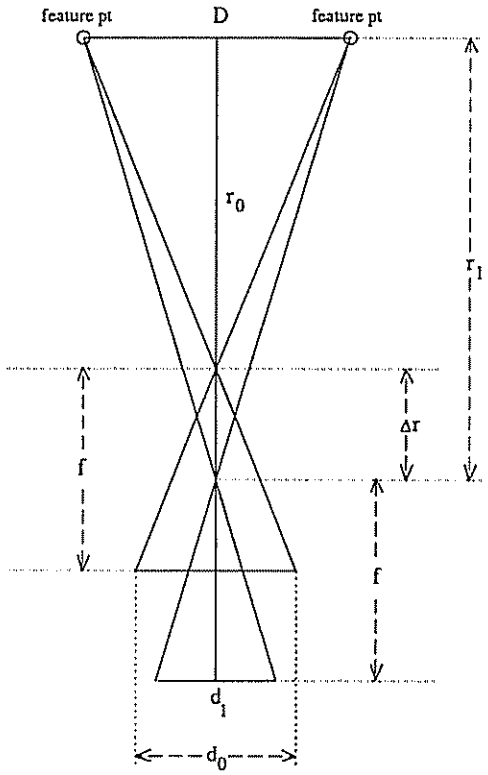

Figure 3: Distance estimation with a step away from object

Thus, we define a *special view* with respect to an object based on three non-collinear feature points on an object. The points must be of fixed position with respect to the object. Let the points be P_0, P_1 and P_2. Let d_{ij} be the distance in the current image between points P_i and P_j. A special view is a position at which d_{01} and d_{02} are both maximized, over all angles of view, at a particular distance r from the center of the line joining P_0 and P_1. The parameter r is not chosen explicitly, since it is not an image-data-based quantity. Instead, a desired value for d_{01} is chosen that compromises between having d_{01} large enough to allow subtle tilt differences to be detected, and small enough that the tracked feature points will not be displaced beyond the edges of the frame from one image to the next, as outlined above, for a reasonable translation step size. Then initial steps towards or away from the midpoint between P_0 and P_1 bring d_{01} to an initial projected length equal to the desired length. The distance r' at which this condition holds is used for attempts to acquire the special view. Final steps away from the chosen features, when at the orientation of the special view, will bring the sensor to the desired position. Should d_{01} grow too large or small during view acquisition, r' is re-chosen.

Figure 4 shows a geometric construction giving the two special views with respect to a particular choice of feature points. L_{ij} is the line joining points P_i and P_j. Shown in the figure are the planes that are perpendicular bisectors of L_{01} and L_{02}. Their intersection is a line. The point or points at which this line cuts a sphere at a fixed distance r from the centre of L_{01} is a special view.

In order not to suffer from having too many possible special views, it is necessary to order somehow the feature points in the initial image. Undoubtedly, the ordering of the points will vary as the viewpoint changes, so the appearance of the object from multiple likely special views should be stored, to increase the probability of successful recognition from the chosen view.

Given our definition of a special view, we can now specify directions of camera translation on the sphere to determine the view. We take advantage of the fact that our definition of a special view has two independent conditions, based on each of the two point-pairs in use. We may maximize the separation of the first point pair without attempting to track the third point at all, and defer choice of a third point until we are on the plane defined by the first condition. This increases the robustness of the process to correspondence errors and feature detection failures.

Estimating the position maximizing the separation of the first pair of points requires that we determine from the

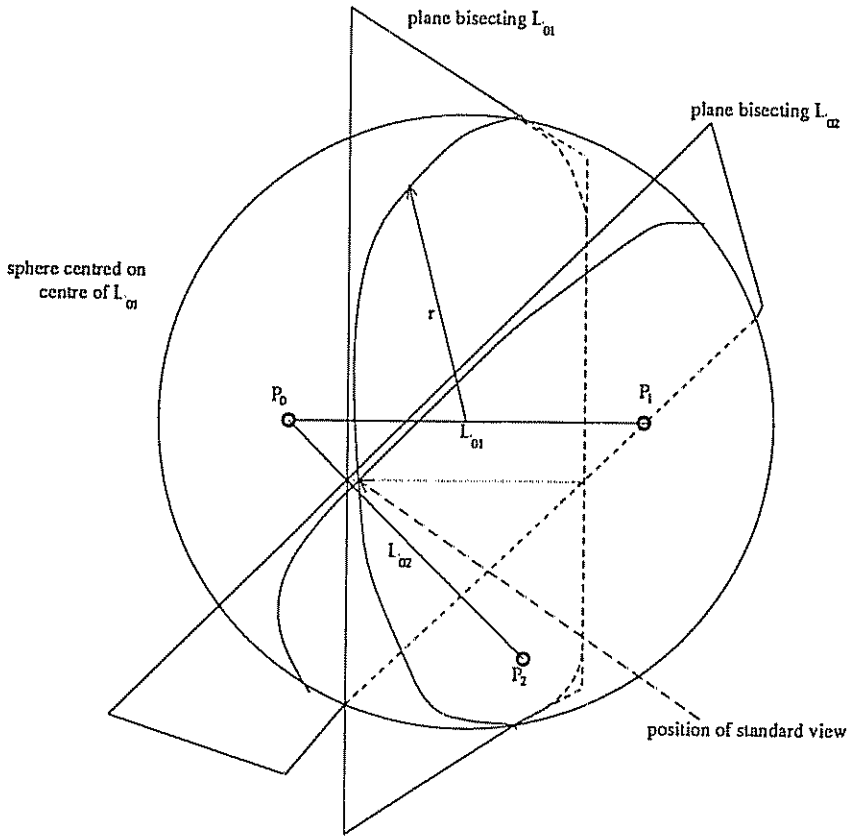

Figure 4: A special view

sensor data how the separation varies as a function of sensor position. Moves parallel to the image of the line joining the feature pair maximize the change in line length per unit sensor motion. Thus, translational moves are made in such directions.

Once the position maximizing the separation of the first point pair is found, moves must be made to determine how the separation of the second point pair varies with position. In order to maintain the first condition, these moves *must* be perpendicular to the image of the line joining the first point pair.

To summarize, the motion prescribed thus consists of *rotations* of the sensor to keep the chosen features centred, *radial translations* to keep the sensor at an appropriate distance from the features and *tangential translations* parallel, then perpendicular to a line joining a single feature pair. Each of these motions may be expressed as a simple behaviour driven by the current image data. The radial translations also require distance estimation using the previous frame's data.

3.2 The executive

The three interleaved behaviours derived above require an *executive* process to control them. The executive must decide when to start, restart or terminate the behaviours. It is also needed to compute the location of a special view from samples of d_{01} or d_{02} collected from images acquired during motion. Finally, the executive must decide on which features to use and how many special views to acquire in order to achieve reliable recognition.

The location of a position maximizing one of the d_{0i}, $(i = 1, 2)$, given a set of length samples from different positions on the viewing sphere, requires that a curve describing the length variation be fit to the data using robust statistics. The maximum of the curve is used as the special view position.

Our definition of a special view has the interesting property that one may drive to an alternate special view from any current viewpoint, by simply choosing a new feature point to include in place of an old one in the defining set. Thus, if some condition causes a particular view to be abandoned, the acquisition of a new view may proceed from the current location. This property fails to hold for certain classes of objects consisting of only a few planar faces (e.g.

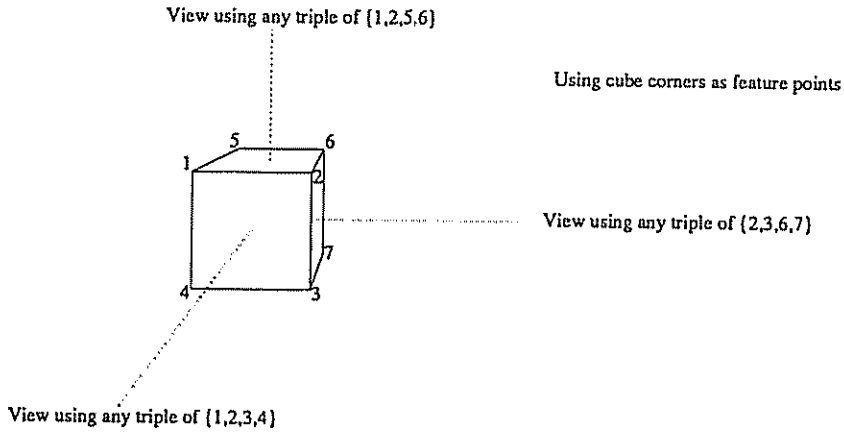

Figure 5: Special views from all visible features may be similar

cubes). Given our definition of a special view, most special views on such objects have the sensor aiming direction perpedicular to one of the planes, and the majority of the visible feature points lie in the chosen plane. Thus, choice of new feature points tends to lead to a new special view similar to the previous one. This is illustrated in figure 5.

A second means of acquiring an alternate special view is to ensure somehow that the new viewpoint will not use defining features similar to the features of the old viewpoint. One way to do this is to begin seeking the new viewpoint from a position *minimizing* d_{01} on the sphere at distance r'. A drawback of this scheme is that the required large viewpoint change may be costly.

There are several conditions that may lead the executive to begin acquisition of a new special view:

- Detection of a critical correspondence failure. This is possible when the robust statistical methods for determining the maximum of d_{0i} detect too many outlier data points.
- Continued acquisition of the current special view becomes physically impossible. This may occur, for example, when the camera attempts to drive through a piece of furniture, or when the view is occluded.
- The views acquired so far have not resulted in sufficient certainty about object identity.

This list of conditions lead us to an interesting view: that using the current state of the image interpretation in some non-trivial way to determine where to look next (as suggested by Bajcsy [1]) may not be a reasonable approach for us. We only wish to decide where to look next when something is wrong with our current state. Using a state known to be problematic to determine further actions is probably little better than a fresh (random) start, and may possibly be worse.

3.3 The behaviours

We now present details of the behaviours.

The centering behaviour keeps L_{01} centred in the image, using a simple control loop based on the current centering error. Figure 6 illustrates this control loop. With each frame grabbed from the camera and processed to extract line segments, our tracking algorithm is invoked to determine which segment corresponds to L_{01}. The control loop generates its error signal from the difference in position between the centre of the current L_{01} segment and the image centre.

The line-following behaviour moves the image plane of the camera parallel to the image of L_{01}. This involves computing a small step in the desired direction, and executing the step with the robot arm. Figure 7 illustrates this open-loop control.

The distance-correcting behaviour attempts to keep the camera roughly on the surface of a sphere centred on L_{01}, at the fixed distance r from its centre. This is done by the control loop shown in figure 8. The control loop generates its error signal using the angle θ travelled by the centre of L_{01} during one step of the first behaviour. Figure 9 shows the geometry used to derive the error signal. Let r_1 be the initial, desired distance of the centred L_{01}

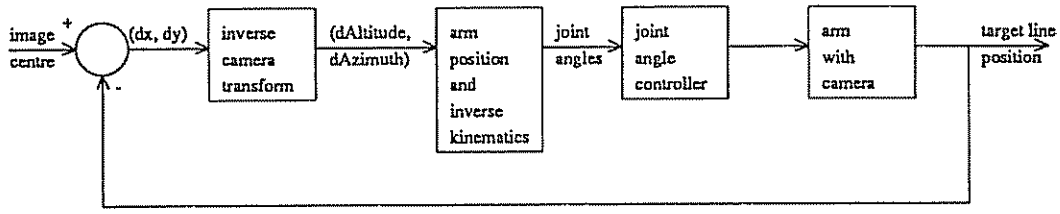

Figure 6: The centering control loop

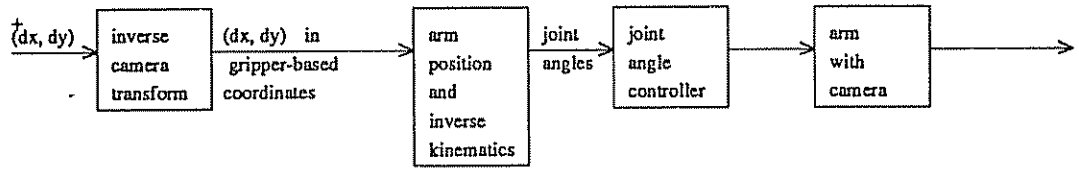

Figure 7: The line-following control

from the optical centre, and r_2 be the distance after a line-following step of length D. Then we have that

$$r_1 = \frac{D}{tan\theta} \tag{5}$$

$$r_2 = \frac{D}{sin\theta} \tag{6}$$

We wish to drive the camera back to the distance r_1, so the desired motion is by an amount dr towards the recentred L_{01}, where

$$dr = r_2 - r_1 \tag{7}$$

$$= \frac{D}{sin\theta} - \frac{D}{tan\theta} \tag{8}$$

$$= \frac{Dtan\theta - Dsin\theta}{sin\theta tan\theta} \tag{9}$$

$$= \frac{Dtan\theta cos\theta - Dsin\theta cos\theta}{sin^2\theta} \tag{10}$$

$$= \frac{Dsin\theta\,(1 - cos\theta)}{sin^2\theta} \tag{11}$$

$$= \frac{D\,(1 - cos\theta)}{sin\theta} \tag{12}$$

Note that this is the only use made by the behaviours of any inferences about object depth.

4 Implementation

The implementation is based on low-level processing which extracts four-parameter line segments from image data.

Figure 10 shows our CRSPlus five-degree-of-freedom robot arm mounted on a TRC Labmate mobile base. The arm has a revolute waist, prismatic shoulder, elbow and wrist, with a second revolute joint just before the gripper. We mount a camera and light in the gripper. By solving the inverse kinematics of the arm, we are able to specify the five joint angles necessary to bring the camera to a specified position in the base coordinates of the robot, with a specified aiming direction. Note that in order to achieve a particular position and aiming direction, the camera may take on an arbitrary rotation about the optical axis of the lens. The base is used to move the whole arm under program control to recentre the arm in its workspace when a move of the arm would take it outside of its workspace.

The executive chooses a prominent line in the image to be maximized. The current implementation simply uses line length as the measure of prominence. Alternative measures could be based on the degree of support for the line, or could choose the closest line to some target location supplied by a higher-level system. Since our system is

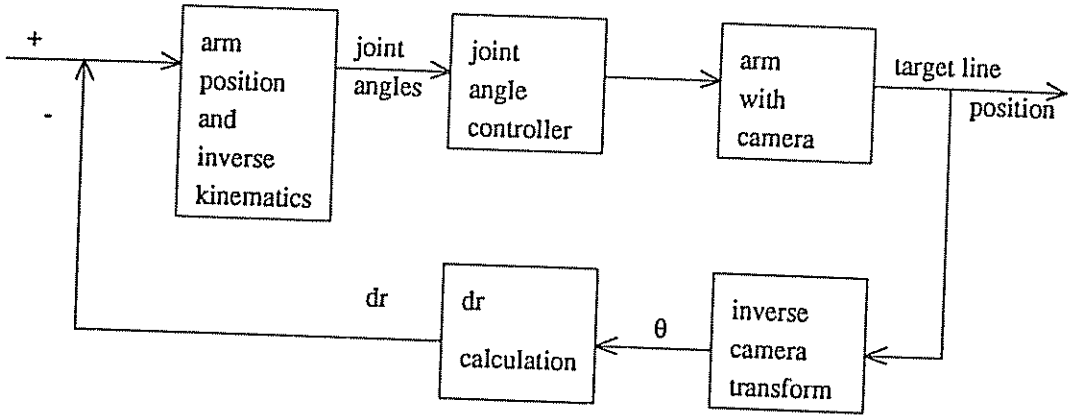

Figure 8: The distance-correcting control loop

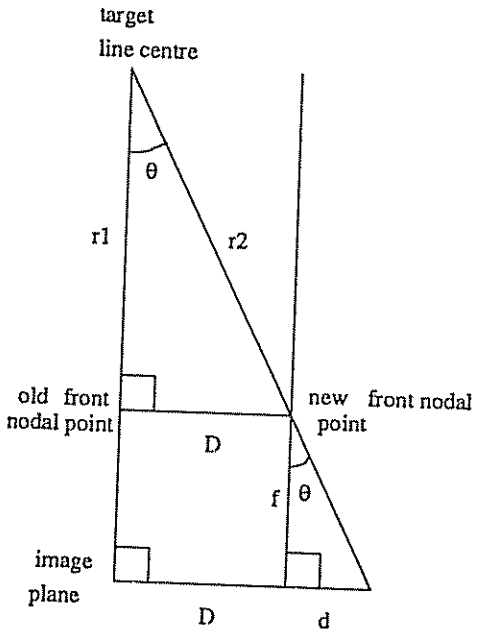

Figure 9: The distance-correcting loop error calculation

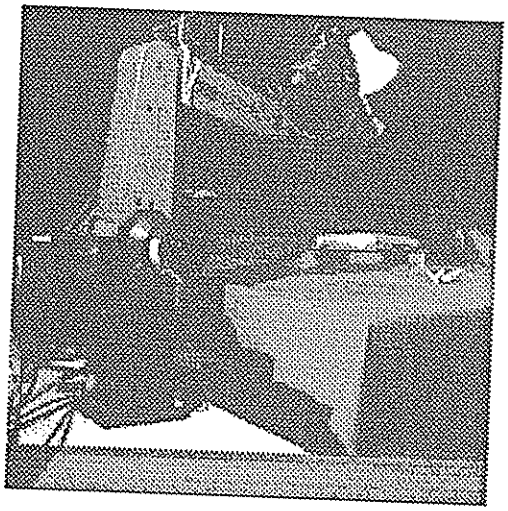

Figure 10: The robot arm, mobile base and camera

initially very conservative about accepting support points for the line segments, length serves well as a measure of the likelihood that the line will persist. The chosen line is used as L_{01}.

Up to sixteen samples of each line at equally spaced angles on the sphere are fit with a parabola to determine the position at which the line has maximal length. The variation in length is actually described by the cosine of the tilt angle, but a parabola provides an adequate approximation.

The fit is a least-squares fit. By itself, this would not be appropriate, since it assumes Gaussian errors on the line length measurements and that the parabola is the correct underlying model (see, for example, Press et al. [6]). We allow for the larger tails in the distribution of length measurements, and other variations in the line segments at a given camera position, by checking each data point to see if it is an outlier. The check is based on the discrepancy between the data point's value as predicted by the other data points, and its measured value.

Having found a position at which L_{01} has maximal length, the camera is returned to that position, and the tracking token set from the neighbourhood of the maximal position is restored. This allows line tracking to continue from this point, which can be at a significantly different viewpoint on the object than the position of the last length sample.

As discussed earlier, the locus of points at which L_{01} has maximal length for a given viewing distance is the intersection of the plane that bisects the line and is perpendicular to the line with the sphere at the given distance. Thus, we may maintain the maximal length of L_{01} by moving perpendicularly to its image. The next stage of the camera motion thus interleaves three behaviours as before, but with L_{01}-following replaced by motion perpendicular to L_{01}, in order to get length samples for a second line, L_{02}.

The executive chooses L_{02} based on two criteria. It must not be parallel to L_{01}, and it should have an endpoint close to one belonging to L_{01}, to increase the likelihood that the two lines belong to the same object. A position on the sphere at which L_{02} has maximal length is found using another fit of a parabola to length measurements. The final result is a camera position on the sphere jointly maximizing the lengths of L_{01} and L_{02}.

There are a number of potential sources of difficulty for the camera motion to a special view.

The line tracking strategy may occasionally run into difficulty due to the disappearance of one of L_{01} or L_{02}. In such a case, there will be one of three results. One possibility is that the disappearance is for a small enough number of frames that the original line is recovered when it reappears. In this case, the one or two erroneous length samples that result are typically discarded as outliers in the quadratic fit. Another possibility is that a different physical edge is permanently substituted for the edge that disappeared. In this case, the quality of the special view found depends on whether the points for one of the two physical edges that were sampled are discarded as outliers, or retained as part of the fit of the parabola to the length data. The latter case will generally result in a failed recognition attempt, and a subsequent restart of the motion behaviours. Finally, it is possible that no correspondent for the line being sampled is found. In this case, the algorithm restarts from current position, at the most recent viewpoint, with new features.

Care must also be taken to make sure that the motion will not attempt to drive the camera through any solid obstacles. Our solution to this problem is to restrict the arm position to the quarter-sphere above and in front of the object. Solutions that detected the collision, and then did something sensible, would be preferable.

Our current executive restarts from a random position on the accessible portion of the viewing sphere at distance r' from the object, whenever a restart is required.

5 Trials

Our experiments use a set of eight origami objects, as shown in figure 11.

Figure 12 depicts the camera motions involved in arriving at a single special view. The positions of lines used as L_{01} and L_{02} were measured relative to the initial base position of the robot. The lines used as L_{01} and L_{02} are shown, as well as other model lines. The translucent sphere shows the distance r' from the centre of L_{01}. The camera motions and the ideal standard view direction are shown as thick lines.

The ideal camera motion consists of two arcs on the surface of the transparent sphere. One arc corresponds to the motions parallel to L_{01} (collecting its length samples). The other arc corresponds to motions perpendicular to L_{01} (collecting L_{02} length samples). In practice, occasional correspondence failures cause direction changes on the arcs.

Figure 13 tabulates some detailed motion trials, that demonstrate the robustness of the system to missed correspondences and other line-finding errors. The columns labelled L_{01} and L_{02} show which actual object edges were believed to be each of L_{01} and L_{02} at each step. For example, the trial with object number 4 shows that L_{01} was

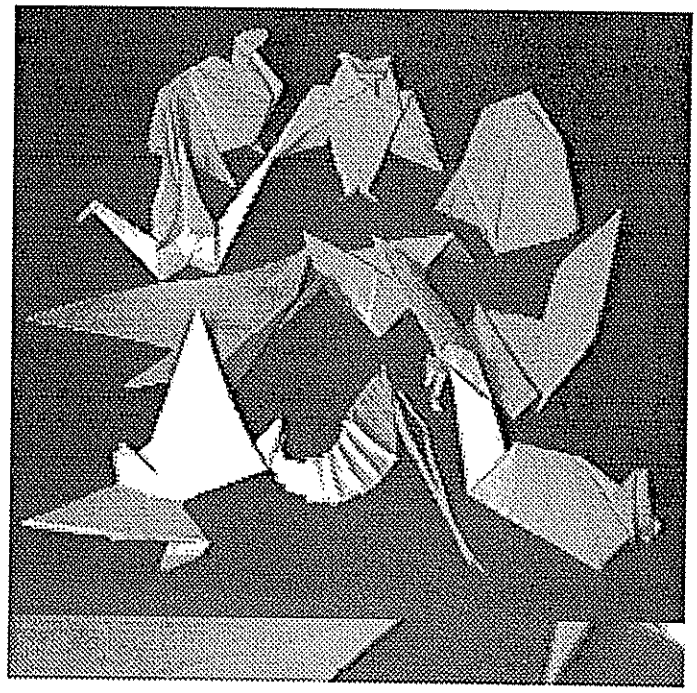

Figure 11: The object set.

associated with a particular edge in 23 of the 24 camera frames in the motion sequence. The correspondence module incorrectly associated a different edge with L_{01} on the final frame. L_{02} was associated with a certain edge in 9 of 12 frames, with an incorrect correspondence persisting for 3 frames during the fourth to sixth frames.

The robust curve fit manages to find a reasonable maximal-length position in many cases in which a different actual line was taken to be the tracked line at some point. The important thing is that one object edge is taken to be L_{01} (or L_{02}) for a clear majority of the camera frames. The effect on camera motion in the case of a minority of missed correspondences for L_{01} is best expressed in terms of the two affected behaviours. The *line-following* behaviour follows the wrong line, taking the camera motion somewhat off-course for a while. Since a line mistaken for L_{01} tends to have similar parameters to the correct line, the overall effect is typically small. The *centering* behaviour centres the wrong line for a while.

The main strength of the behaviour-based motion is that *a system with little internal state can accommodate errors in the low level processing.* The behaviours are able to continue operating the face of a missed line segment, since their state does not depend on which line is being tracked or centered. This immediately allows the entire system to work in the presence of what would otherwise be catastrophic (but nonetheless typical) errors.

The column labelled "base positions" in the table shows the number of distinct robot arm base positions needed to complete the camera motion.

Notice that one of these trials failed because all of the lines detected in the image were nearly parallel. In this case, there was no suitable candidate for L_{02}. This problem could be overcome by constructing a line connecting endpoints of different lines in the parallel group; such a line could be considered to be L_{02}.

Figure 14 illustrates the motions for a complete recognition run. We have chosen a fairly typical run, consisting of two special-view acquisitions. The same model segment was used for L_{01} in each case, but different edges were used for L_{02}. The first view acquisition was close to ideal, in that the error is only a few degrees, and no correspondences were missed. Notice that the final scaling steps for the first view acquisition are not exactly parallel to the ideal special view axis, and so the error was increased slightly by the scaling steps. This is not normally a significant component of the total error, since usually view acquisition is not so accurate. The second acquisition, although it resulted in a successful index lookup, was perturbed from the ideal by a couple of factors. First, the randomly chosen starting point was a position of view degeneracy in which L_{01} was collinear with an adjacent edge. the result was that L_{01} appeared longer than it really was, and the motion steps were done at a significantly larger distance than in the first acquisition. Secondly, there were a number of missed detections of the line used for L_{02} that were not detected by the robust parabola-fit, so the position chosen to maximize the length of L_{02} was somewhat in error.

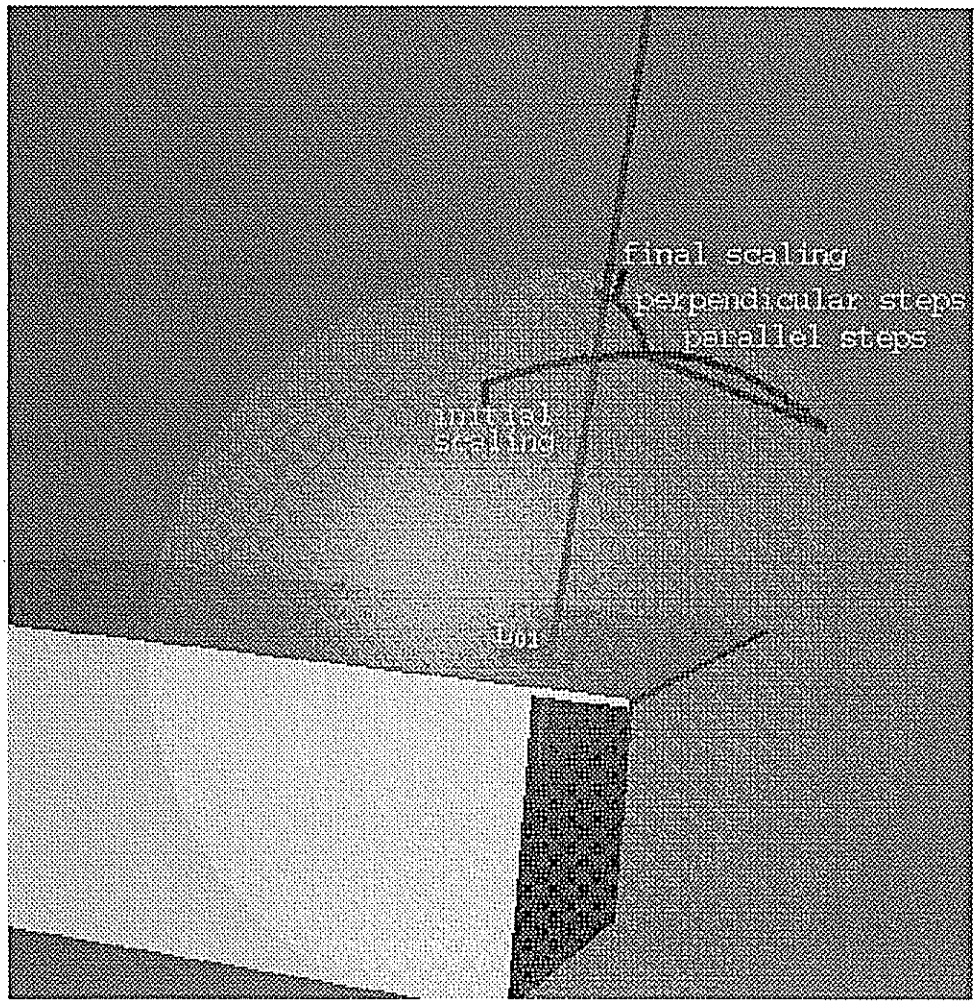

Figure 12: Example of a single motion to a standard view.

object	L_{01}	L_{02}	base positions	comments
1	1(24 positions)	2(12 positions)	3	approx. 15 degree error
2	2(1),1(8),4(1),1(14)	3(5),5(1),3(6)	3	30 degrees
3	1(24)	2(1),3(5),4(6)	3	Poor fit (40 degrees), restart
3	1(24)	4(9),5(3)	4	10 degrees
4	1(23),4(1)	2(3),3(3),2(6)	2	less than 10 degrees
5	1(20),3(4)	2(4),1(4),3(4)	4	no good L_{02}: all lines parallel
5	1(24)	4(12)	3	10 degrees
6	1(24)	2(1),3(11)	2	15 degrees
7	1(10),2(2),1(12)	3(12)	4	less than 10 degrees

Figure 13: Sample camera motion trials

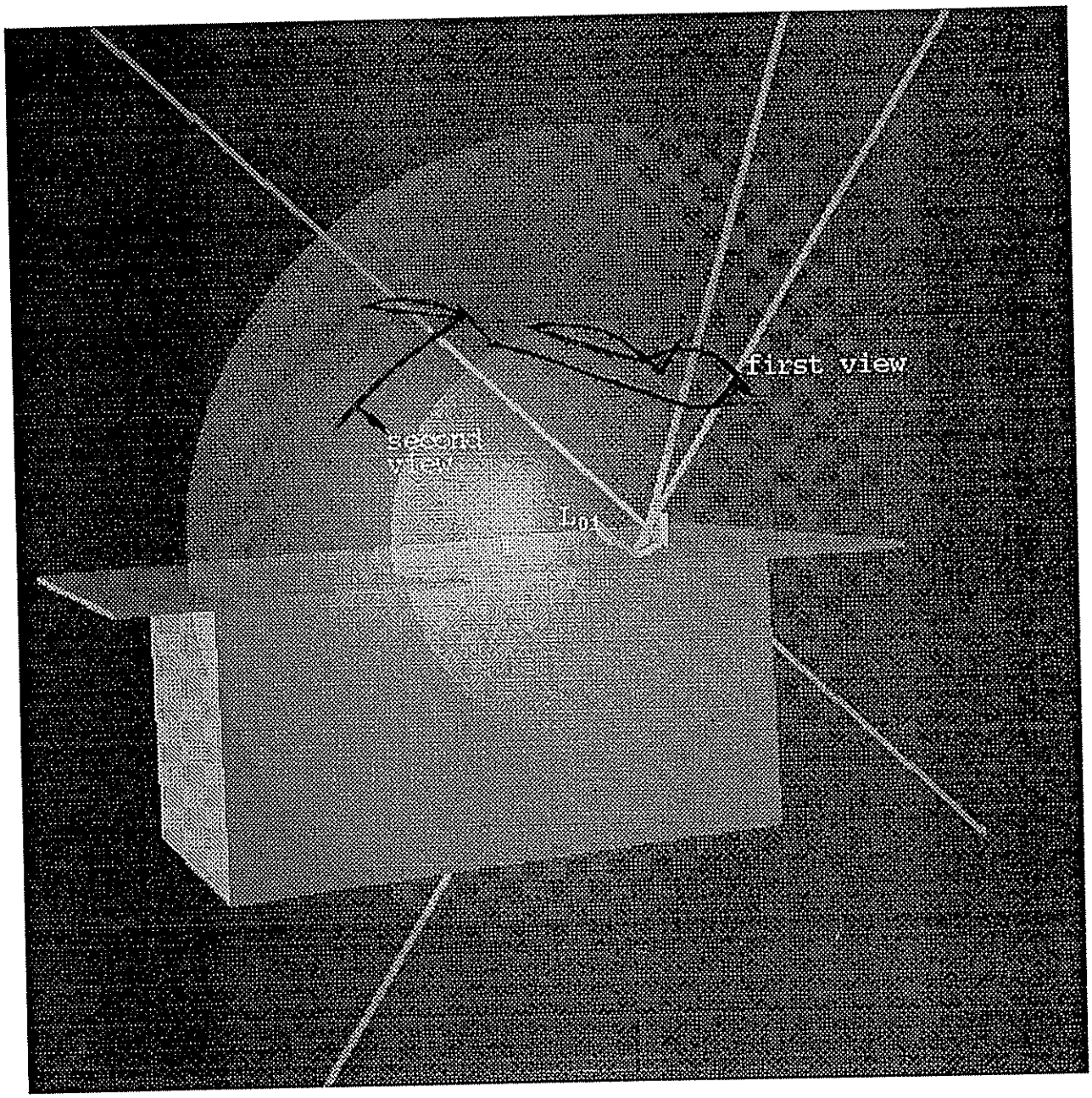

Figure 14: A complete recognition run

6 Conclusions and Future Work

We have demonstrated that robust viewpoint control is possible, reducing 3D object recognition to viewpoint control plus 2D recognition. There are several directions for future work.

Occlusion and clutter The current system does not address the issues of occlusion and/or clutter in the scene. As discussed by Grimson[5], the problem of correctly grouping model features belonging to a single object amidst clutter is hard. We wish to determine a method for adapting each of our indexing and tracking schemes to perform efficiently this grouping.

We believe that the problem is one of producing a model that distinguishes feature variation due to measurement error from feature variation due to extraneous scene parts.

Use of different image primitives More interesting object sets could be attempted with a more sophisticated low-level system.

The camera motion strategy operates on the distances between two selected pairs of point features. In our case, the point features were the endpoints of line segments extracted from the data. As long as the points correspond to stable positions on the object to be recognized, they could be points derived from any low-level primitive. All that is required is a set of three non-collinear feature points, with the distance between two selected pairings of the points to be maximized.

Real time implementation The majority of the processing time in the current prototype implementation is devoted to low-level image processing. We are interested in pursuing a second implementation, in which dedicated image processing hardware does as much of the initial feature extraction as possible.

Object motion instead of camera motion As the speed of computation continues to increase, the system will become increasingly limited in speed by the need for large mechanical motions. One solution to the problem is to reduce the amplitude of the mechanical motions. In the case of small objects, this may be achieved by moving the object instead of the camera. There are at least two possible approaches to this. Each uses several cameras to reduce the range of motion needed to observe all object viewpoints. The first method provides rotations of the object by grasping it with a robot arm end effector, in order to rotate the object with respect to the camera. The second uses a tilting turntable as a base for the object.

Acknowledgements The authors wish to thank the Natural Sciences and Engineering Research Council of Canada, the Information Technology Research Centre, and the Canadian Institute for Advanced Research for financial support. The authors are members of the Institute for Robotics and Intelligent Systems (IRIS) and wish to acknowledge the support of the Networks of Centres of Excellence Program of the Government of Canada and the participation of PRECARN Associates Inc. The second author is the CP (Unitel) Fellow of the Canadian Institute of Advanced Research.

References

[1] Bajcsy, R., "Active Perception vs Passive Perception," *Proc. IEEE Workshop on Computer Vision: Representation and Control*, Bellaire, Michigan, 1985.

[2] Burns, J.B., Weiss, R., Riseman, E.M., "View variation of point set and line segment features," *Proc. DARPA IU Workshop*, 1990.

[3] Clemens, D.T. and Jacobs, D.W., "Space and Time Bounds on Indexing 3-D Models form 2-D Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 10, 1991.

[4] Dickmanns, E.D., and Graefe, V., "Dynamic Monocular Machine Vision and Applications of Dynamic Monocular Machine Vision," *Tech. Report UniBwM/LRT/WE 13/FB/88-3*, Institút fur Meβtechnik, Universitat der Bundeswehr Munchen, July 1988.

[5] Grimson, W.E.L., "The Combinatorics of Object Recognition in Cluttered Environments using Constrained Search," *Proc. Second International Conference on Computer Vision*, IEEE Computer Society Press, Washington, 1988.

[6] Press, W.H., Flannery, B.P., Teukolsky, S A. and Vetterling, W.T., *Numerical Recipes, The Art of Scientific Computing*, Cambridge University Press, New York, 1986.

[7] Wilkes, D., Dickinson, S., Tsotsos, J., "Quantitative Modelling of View Degeneracy," *Proc. Eight Scandinavian Conference on Image Analysis*, Tromso, Norway, 1993.

[8] Wilkes, D., *Active Object Recognition*, Ph.D. Thesis, Department of Computer Science, University of Toronto, 1993 (expected).

[9] Zheng, J.Y., Chen, Q., Kishino, F., and Tsuji, S., "Active Camera Controlling for Manipulation," *Proc. Computer Vision and Pattern Recognition '91*, IEEE Computer Society Press, Washington, 1991.