Human–Robot Interface by Pointing with Uncalibrated Stereo Vision

Roberto Cipolla and Nicholas J. Hollinghurst

Department of Engineering,
University of Cambridge,
Cambridge CB2 1PZ, UK.

Abstract

Here we present the results of an investigation into the use of a **pointing-based interface** for robot guidance. The system requires no physical contact with the operator, but uses **uncalibrated stereo vision** with **active contours** to track the position and pointing direction of a hand in real time. With a **ground plane constraint**, it is possible to find the indicated position in the robot's workspace, by considering only two-dimensional collineations.

Experimental and simulation data show that a resolution of within 1cm can be achieved in a 40cm workspace, allowing simple pick-and-place operations to be specified by finger pointing.

1 Introduction

There has been a lot of interest lately in the use of hand gestures for human–computer interfacing: they are intuitive for the operator, and provide a rich source of information to the machine. This type of interface is particularly appropriate in applications such as virtual reality, multimedia and teleoperation [1, 2, 3].

Most current commercial implementations rely on sensors that are physically attached to the hand, such as the 'DataGlove' [4]. More recently, systems have been proposed using *vision* to observe the hand. Some require special gloves with attachments or markings to facilitate the localization and tracking of hand parts [5, 6], but others operate without intrusive hardware. This is attractive because it is convenient for the user and potentially cheaper to implement.

Here we present an experimental implementation of such a system, concentrating in particular on the case of *pointing* at a distant object. We have developed a simple vision-based pointing system as an input device for a robot manipulator, to provide a novel and convenient means for the operator to specify points for pick-and-place operations. We use *active contour* techniques to track a hand in a pointing gesture, with conventional monochrome cameras and fairly modest computing hardware.

A single view of a pointing hand is ambiguous: its distance from the camera cannot be determined, and the 'slant' of its orientation cannot be measured with any accuracy. Stereo views are used to recover the hand's position and orientation, and yield the line along which the index finger is pointing. In our system, we assume that the user is pointing towards a 'target surface,' which is a horizontal plane. We show how a simple result from projective geometry can be applied to this case, allowing the system to be implemented with *uncalibrated* stereo, that requires no measurements or special assumptions about camera positions and parameters.

2 Geometrical framework

2.1 Viewing the plane

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

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \mathbf{T} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}, \tag{1}$$

where s is a scale factor that varies for each point; and $\mathbf{T}$ is a 3×3 transformation matrix. The system is homogeneous, so we can fix $\mathrm{t}_{33} = 1$ without loss of generality, leaving 8 degrees of freedom. To solve for $\mathbf{T}$ we must observe at least four points. By assigning arbitrary world coordinates to these points (e.g. $(0,0)$, $(0,1)$, $(1,1)$, $(1,0)$), we define a new coordinate system on the plane, which we call *working plane coordinates.*

Now, given the image coordinates of a point anywhere on the plane, along with the image coordinates of the four reference points, it is possible to invert the relation and recover the point's working plane coordinates, which are invariant to the choice of camera location [7]. We use the same set of reference points for a stereo pair of views, and compute two transformations $\mathbf{T}$ and $\mathbf{T}'$, one for each camera.

2.2 Recovering the indicated point in stereo

With natural human pointing behaviour, the hand is used to define a line in space, passing through the base and tip of the index finger. This line will not generally be in the target plane but intersects the plane at some point. It is this point (the *'piercing point'* or *'indicated point'*) that we aim to recover. Let the pointing finger lie along the line l_w in space (see figure 1). Viewed by a camera, it appears on line l_i in the image, which is also the projection of a *plane*, $\mathcal{P}$, passing through the image line and the optical centre of the camera. This plane intersects the ground plane $\mathcal{G}$ along line l_{gp}. We know that l_w lies in $\mathcal{P}$, and the indicated point in l_{gp}, but from one view we cannot see exactly where.

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

Repeating the above procedure with the second camera C$'$ gives us another view l'_i of the finger, and another line of constraint l'_{gp}. The two constraint lines will intersect at a point on the target plane, which is the indicated point. Its position can now be found relative to the four reference points. Figure 2 shows the lines of pointing in a pair of images, and the intersecting constraint lines in a 'canonical' view of the working plane (in which the reference point quadrilateral is transformed to a square). This is a variation of a method employed by Quan and Mohr[8], who present an analysis based on cross-ratios.

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

3 Tracking a pointing hand

3.1 Background

A large number of systems have been proposed for visual tracking and interpretation of hand and finger movements. These systems can broadly be divided into:

- those concerned with gesture identification (e.g. for sign language), which compare the image sequence with a set of standard gestures using correlation and warping of the templates [9], or classify them with neural networks [10];
- those which try to reconstruct the pose and shape of the hand (e.g. for teleoperation) by fitting a deformable, articulated model of the palm and finger surfaces to the incoming image sequence [11].

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

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

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

3.2 Tracking mechanism

We use a type of active contour model [12, 13, 14] to track the image of a hand in the familiar 'pointing' gesture, in real time. Our tracking mechanism was chosen mainly for its speed, simplicity, and modest demand for computer resources. A pair of trackers operate independently on two stereo views.

Anatomy

Each tracker is based on a template, representing the shape of the occluding contours of an extended index finger and thumb (see figure 3). At about 50 sites around the contour (represented in the figure by dots) are *local edgefinders* which continuously measure the normal offsets between the predicted contour and actual features in the image; these offsets are used to update the image position and orientation of the tracker.

The tracker's motion is restricted to 2D affine transformations in the image plane, which ensures that it keeps its shape whilst tracking the fingers in a variety of poses [15]. This approach is suitable for tracking planar objects under weak perspective; however it also works well with fingers, which are approximately cylindrical.

The positions of these sampling points are expressed in affine coordinates, and their image positions depend on the tracker's *local origin* and two *basis vectors*. These are described by six parameters, which change over time as the hand is tracked.

Dynamics

At each time-step, the tracker searches for the maximum image gradient along each sampling interval, which is a short line segment, normal to and centred about a point on the active contour. This yields the *normal offsets* between points on the active contour and their corresponding edge segments in the image.

The offsets are used to estimate the affine transformation (translation, rotation, scale and shear) of the active contour model, which minimises the errors in a least-squares sense. A first order temporal filter is used to predict the future position of the contour, to improve its real-time tracking performance. The filter is biased to favour rigid motions in the image, and limits the rate at which the tracker can change scale — these constraints represent prior knowledge of how the hand's image is likely to change, and increase the reliability with which it can be tracked.

To extract the hand's direction of pointing, we estimate the orientation of the index finger; the base of the thumb is tracked merely to resolve an *aperture problem* [17] induced by the finger's long thin shape.

4 Implementation

4.1 Equipment

The system is implemented on a Sun SPARCstation 10 with a Data Cell S2200 frame grabber. Images are provided by two monochrome CCD cameras, which view the operator's hand and the working area from a distance of about 1.6 metres. The angle between the cameras is about 30°. A Scorbot ER-7 robot arm is also controlled by the Sun (figure 4).

4.2 Experiment

Setup

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

Performance

Figure 5 shows the system in operation. The corners of the white rectangle are the four reference points, and the overlaid square shows the position of the indicated point. Movements of the operator's hand cause corresponding movements of this point in real time.

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

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

4.3 Accuracy evaluation

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

Finger tracker uncertainty

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

These offsets indicate that the image position of the finger's midline can be determined to subpixel accuracy (standard deviation typically $\sigma = 0.3$ pixels), and the orientation to an accuracy of 0.6°.

From this uncertainty measure we calculate $\pm 2\sigma$ bounds on lines l_i and l'_i; and, by projecting these onto the ground plane, estimate the uncertainty in the indicated point.

Figure 6 shows the results for three different configurations of the cameras, with a 95% confidence ellipse drawn around the indicated point. The constraint line uncertainties are much the same in each trial, but the uncertainty on the indicated point varies according to the separation between the stereo views: when the cameras are close together, the constraint lines are nearly parallel and tracker uncertainty is very significant (figure 6a); as the baseline is increased and the stereo views become more distinct, the constraint lines meet at a greater angle and accuracy is improved (figure 6c).

Reference point uncertainty

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

We used Monte Carlo simulations (based around real-world configurations of cameras, hand and table) to assess the impact of this uncertainty on the coordinates of the indicated point. The results (table 1) show that this source of error is less significant than the tracker uncertainty, and confirm that the system is not especially sensitive to errors in the reference point image coordinates. Again, the errors are most significant when the camera separation is small.

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

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

Experimental accuracy

Ground truth about the position and orientation of a human finger is, of course, very difficult to measure without intrusive equipment that could interfere with our stereo vision system. We therefore tested the accuracy of the pointing system using an artificial pointing device (figure 7). The test pointer was a white cylinder, about 15cm long, bounded by black end stops and wrapped around a rod which could be positioned by the robot arm to an accuracy of about 1mm. Whilst not identical to a human hand, it had approximately the same dimensions and was tracked in a similar manner.

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

4.4 Robot control application

The proposed application for this stereo pointing system is to control a robot manipulator as it grasps and places small objects on a flat table-top.

Setup

Here the reference points are defined by observing the robot gripper itself as it visits 4 points in a plane (using active contours similar to those which track the pointing hand [16]). This not only defines the working coordinate system but relates it to the robot's own world coordinate system. Finger-trackers are then initialised as before.

Performance

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

When the distance between hand and workspace is large, the system is sensitive to small changes in index finger orientation (as one would expect). To reduce this sensitivity, the operator maintains a steep angle to the horizontal, and points from a distance of less than 50cm from the plane, whilst still keeping his or her hand clear of the robot. One can then comfortably position the gripper with sufficient accuracy to pick up small objects (figure 8).

5 Conclusion

Pointing can be used successfully to specify positions on a two-dimensional workspace for a robot manipulator. The system is simple and intuitive for an operator to use, and requires practically no training.

This method for computing the indicated point proves to be robust in the presence of tracker uncertainties and noise. Its accuracy depends on the geometry of the stereo cameras, and is best when they are at least an angle of 30° apart. The system does not require camera calibration because all calculation takes place in the image and ground planes. By tracking at least 4 points on the plane it could be made invariant to camera movement.

The main problem for this system is tracking a pointing hand reliably in stereo. At present, this is only possible in an environment where there is a strong contrast between the hand and the background. Our current system requires the index finger and thumb to be kept rigid throughout operation. Tracking speed is limited by our hardware (a single Sun SPARCstation) and could be improved by adding faster computing or image processing equipment. Improvements to hardware performance would allow more sophisticated tracking mechanisms (such as stochastic deformable models [18]) to be incorporated, permitting more degrees of freedom for hand gesturing.

Although subjective pointing direction depends upon eye as well as hand position, it is not necessary to model this phenomenon. Instead, by providing the operator with feedback about the *objective* pointing direction (e.g. having a robot follow the pointing hand in real time), the hand can be aligned with any desired object on the working plane. Points can then be indicated with sufficient accuracy to guide simple pick-and-place operations that might otherwise have been specified using a teach pendant.

Acknowledgements

We are indebted to Mr. Masaaki Fukumoto and Dr. Yasuhito Suenaga of the NTT Human Interface Laboratories, Yokosuka, Japan, who collaborated with Roberto Cipolla on an earlier implementation of this stereo pointing algorithm.

We gratefully acknowledge the financial support of EPSRC, and the donation of a robot manipulator by the Olivetti Research Laboratory, Cambridge.

References

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

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

[3] T. Baudel and M. Beaudouin-Lafan. Charade: remote control of objects with freehand gestures. *Communications of the ACM*, vol. 36 no. 7 pp 28–35, 1993.

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

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

[6] R. Cipolla, Y. Okamoto and Y. Kuno. Robust Structure from Motion using Motion Parallax. *Proc. 4th Int. Conf. on Computer Vision*, pp 374–382. Berlin, 1993.

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

[8] L. Quan and R. Mohr. Towards structure from motion for linear features through reference points. *Proc. IEEE Workshop on Visual Motion*, pp 249–254. Princeton, New Jersey, 1991.

[9] T. Darrel and A. Pentland. Space–time gestures. *Computer Vision and Pattern Recognition*, pp 335–340, 1993.

[10] R. Kjeldsen. Visual hand gesture interpretation. *IEEE computer society workshop on non-rigid and articulate motion.* Austin, Texas. November 1994.

[11] J. Kuch and T. Huang. Virtual gun: a vision based human computer interface using the human hand. *Proc. IAPR Workshop on Machine Vision Applications MVA'94*, pp 196–199. Tokyo, December 1994.

[12] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: active contour models. In *Proc. 1st Int. Conf. on Computer Vision*, pages 259–268. London, June 1987.

[13] R. Cipolla and A. Blake. Surface shape from the deformation of apparent contours. *Int. J. Computer Vision*, vol. 9 no. 2 pp 83–112, 1992.

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

[15] 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*, pp 66–75. Berlin, 1993.

[16] N. J. Hollinghurst and R. Cipolla. Uncalibrated stereo hand–eye coordination. *Image and Vision Computing*, vol. 12 no. 3 pp 187–192, 1994.

[17] S. Ullman. *The interpretation of visual motion.* MIT Press, 1979.

[18] C. Kervrann and F. Heitz. Robust tracking of stochastic deformable models in long image sequences. *1st IEEE Int. Conf. Image Processing*, pp 88–92. Austin, Texas. November 1994.

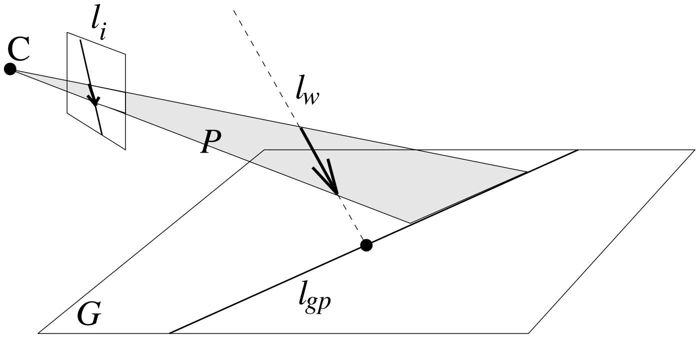
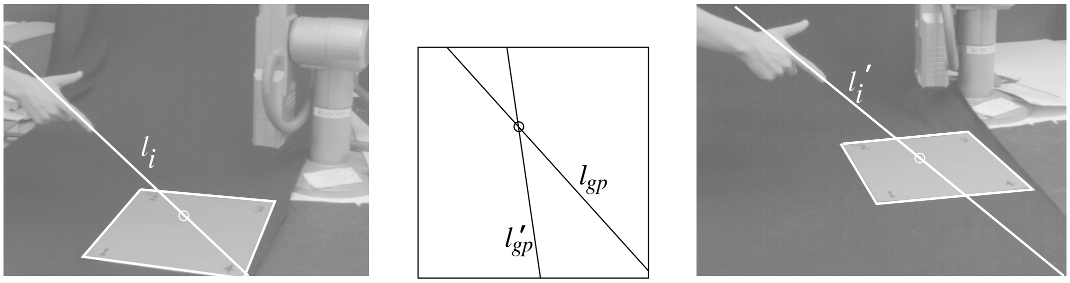


Figure 1: **Relation between lines in the world, image and ground planes.** Projection of the finger's image line l_i onto the ground plane yields a constraint line l_{gp} on which the indicated point must lie.



(a) (b) (c)

Figure 2: **Pointing at the plane.** By taking the lines of pointing in left and right views (a, c), transforming them into the canonical frame defined by the four corners of the grey rectangle (b), and finding the intersection of the lines, the indicated point can be determined; this is then projected back into the images.

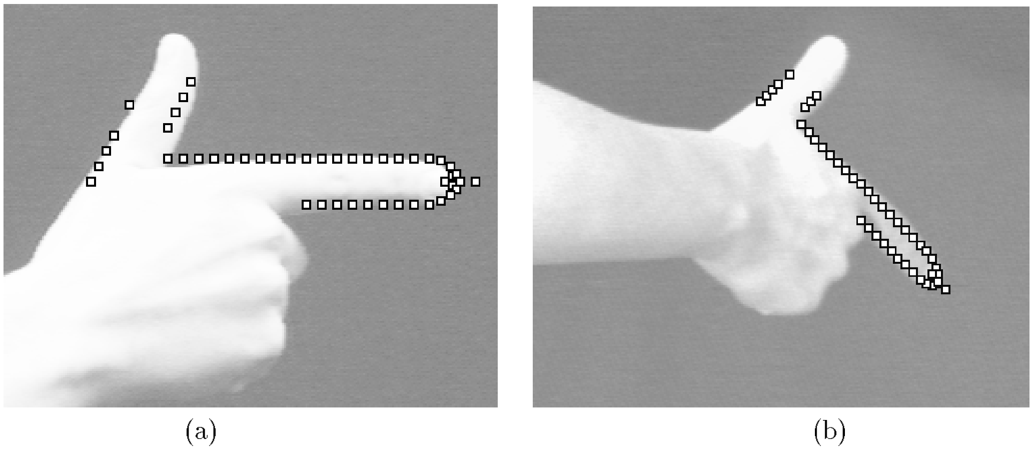


Figure 3: **The finger-tracking active contour**, (a) in its canonical frame (b) after an affine transformation in the image plane (to track a rigid motion of the hand in 3D). It is the index finger which defines the direction of pointing; the thumb is observed to facilitate the tracking of longitudinal translations which would otherwise be difficult to detect.

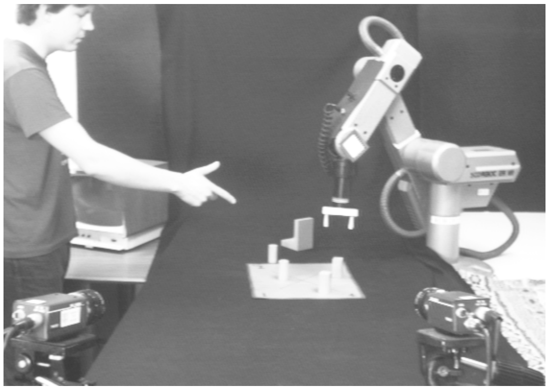


Figure 4: **Arrangement of the stereo cameras, table and robot.** The cameras view the robot, workspace and operator's hand from a distance of about 1.6m.

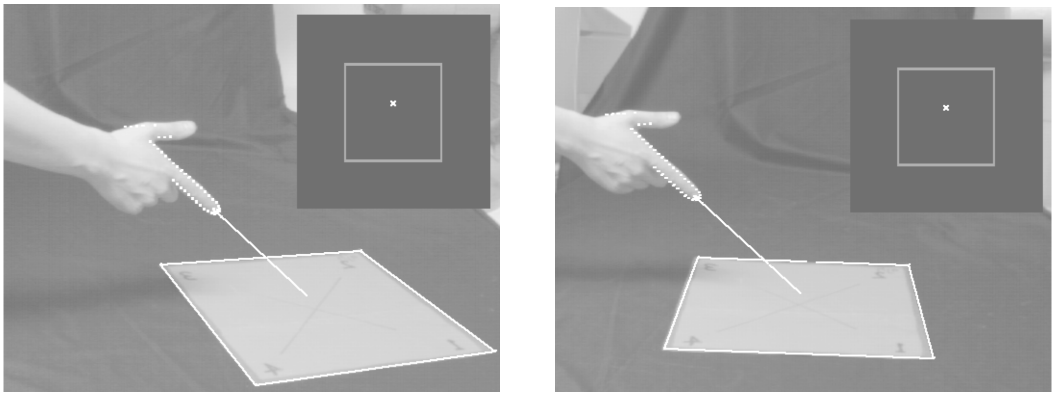


Figure 5: **Stereo views of a pointing hand**. The two views are shown side by side. In each view an active contour is tracking the hand. The inlaid square is a representation of the indicated point in working plane coordinates.

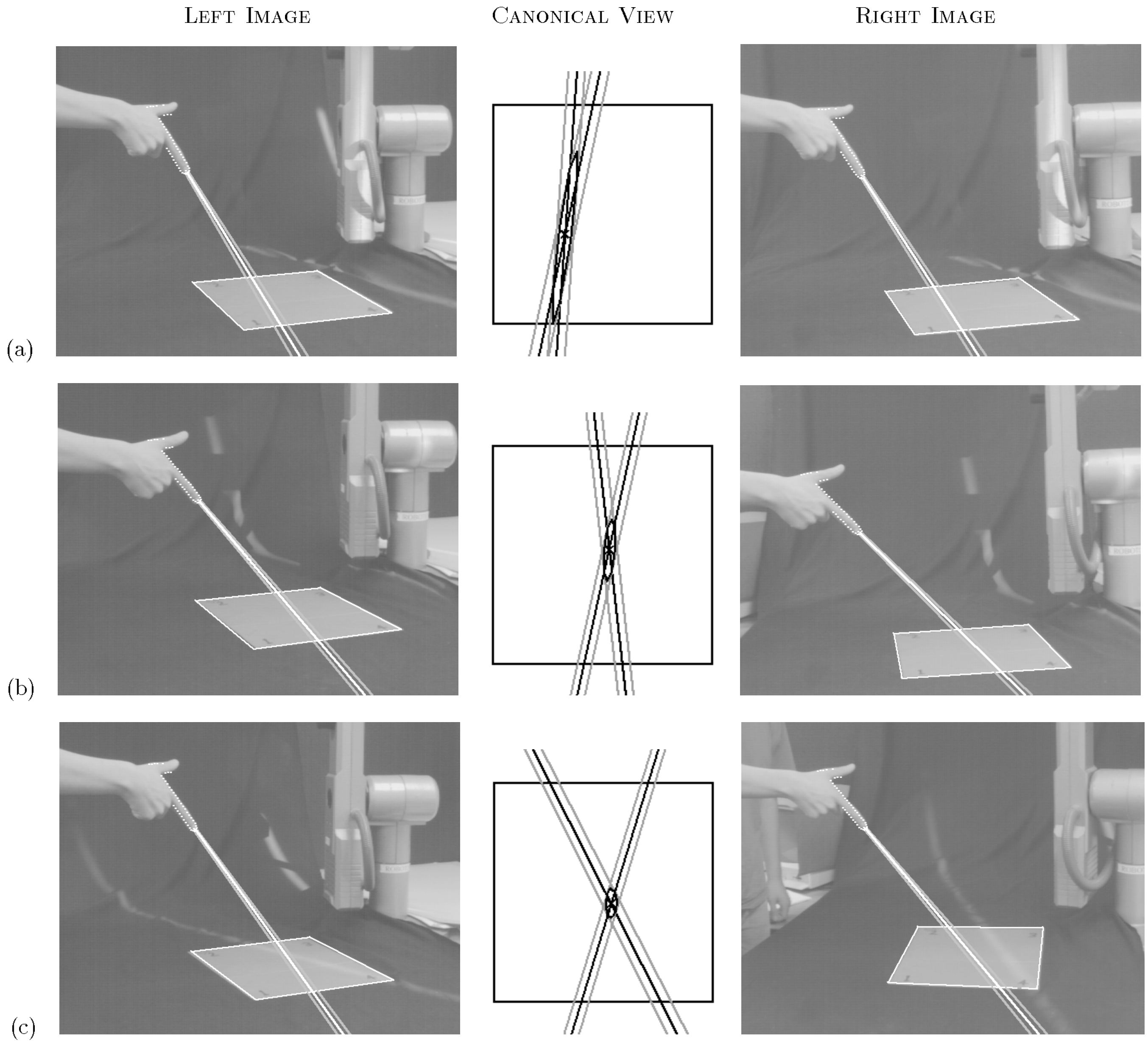


Figure 6: **Indicated point uncertainty for 3 different camera configurations.** 2σ bounds for the pointing lines, their projections into working plane coordinates, and error ellipses for the indicated point, when the angle between stereo views is (a) 7° (b) 16° (c) 34°. The uncertainty is greatest when the camera angle is small and the constraint lines nearly parallel.

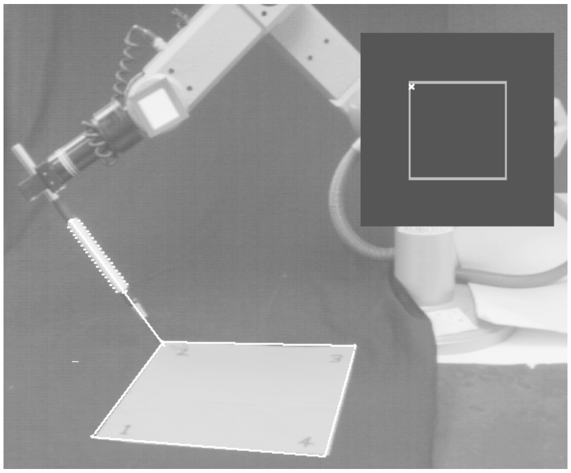
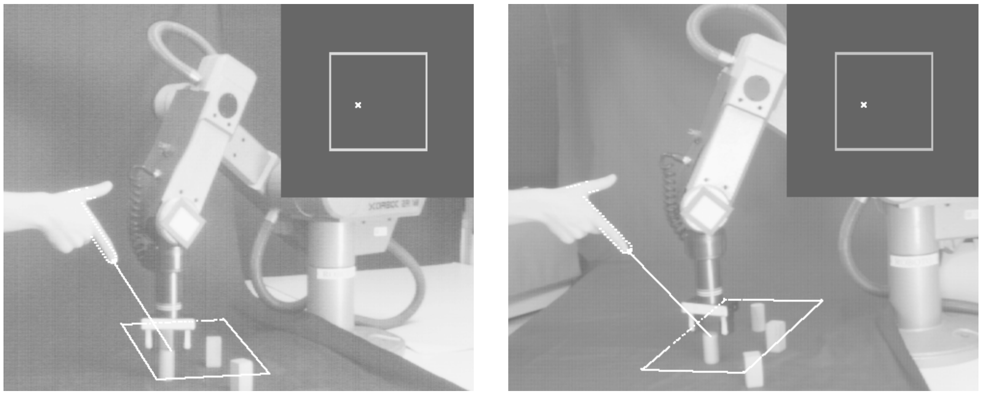


Figure 7: **Mechanical pointing device used to test the accuracy of our system.** We aligned the rod with known points on the workspace, and recorded its coordinates as recovered by the vision system.



(a) (b)

Figure 8: **Gestural control of robot position for grasping, seen in stereo.** The robot gripper servos to the position indicated by the pointing hand; here it is being instructed to align itself with the small wooded block to be grasped. The four reference points (white rectangle) were defined by the robot's gripper in a plane 50mm above the table.