Man–Machine 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 **gesture-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. With a **ground plane constraint**, it is then possible to find the indicated position in the robot's workspace, by considering only two-dimensional collineations.

With feedback to the user, points can be indicated to an accuracy of about 1cm in a 40cm workspace. The system is initialised by observing just 4 points on the plane.

1 Introduction

A number of systems have been proposed in the past for human–computer interaction based on hand gestures and pointing. Some systems required the user to wear a special glove or magnetic sensors [1, 2, 3]. Others using image processing have required calibration for each user's individual hand shape and posture [4].

We have developed a stereo vision 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 [5] to track a hand in a pointing gesture, with conventional monochrome cameras and fairly modest image-processing 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 needed to recover the hand's position and orientation, and yield the line along which the index finger is pointing. We seek to recover the point where this line intersects a planar table-top. The use of such a "ground-plane constraint" effectively reduces the problem to a two-dimensional one.

2 Theory

2.1 Viewing the plane

Consider a pinhole camera vision system viewing a plane. The viewing transformation for each camera is a plane-to-plane projection between some world coordinate system (X, Y) on the ground plane, 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 $\mathrm{t}_{33} = 1$, leaving 8 degrees of freedom. To solve for $\mathbf{T}$ we must observe at least four reference points; and, 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 in 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 [6]. 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 index finger is used to define a line in space, passing through the base and tip of the index finger. This line will not generally be in the ground plane but intersects the plane at some point. It is this *piercing point* [7] 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 the l_w lies in $\mathcal{P}$, and the indicated point in l_{gp}, but from one view we cannot see exactly where.

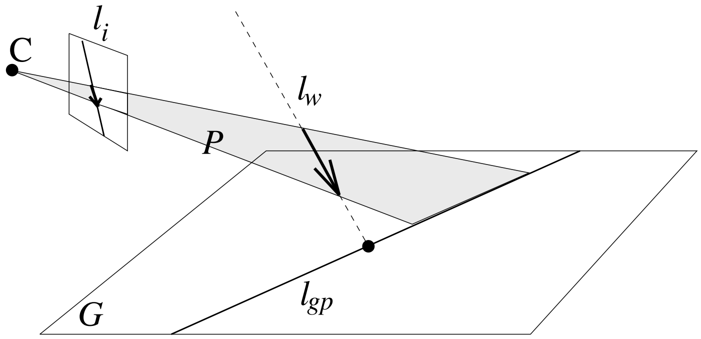
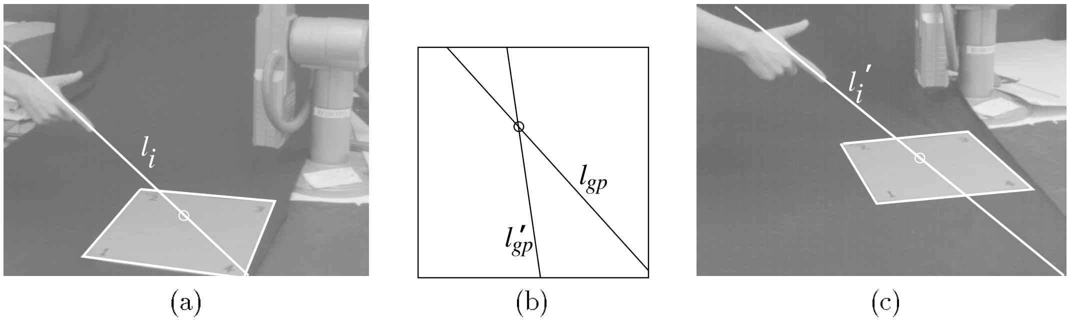


Figure 1: Projection of the finger's image line l_i onto the ground plane yields a constraint line l_{gp} on which the indicated point must lie.

Note that l_i is an image of 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 table.

Repeating the above procedure with the second camera C$'$ gives us another view l_i' of the finger, and another line of constraint l_{gp}'. The two constraint lines will intersect at a point on the ground plane, which is the indicated point. Its position can now be found relative to the four reference points (figure 2).



(a) (b) (c)

Figure 2: 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.

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.

3 Implementation and evaluation

3.1 Equipment

The system is implemented on a Sun SPARCstation 10 with a Data Cell S2200 frame grabber. Images are provided by two PULNiX monochrome CCD cameras, which view the operator's hand and the working area from a distance of about 1.6 metres. The angle betwen the cameras is about 30°. A Scorbot ER-7 robot arm is also connected to the Sun (figure 3).

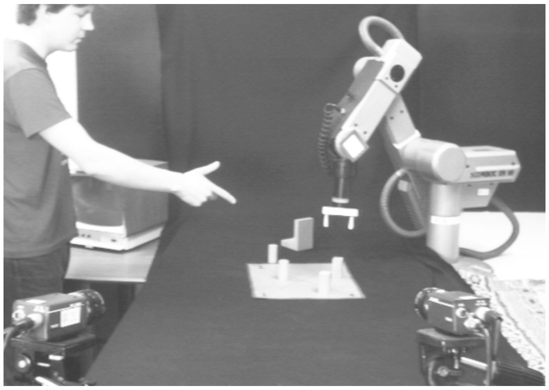
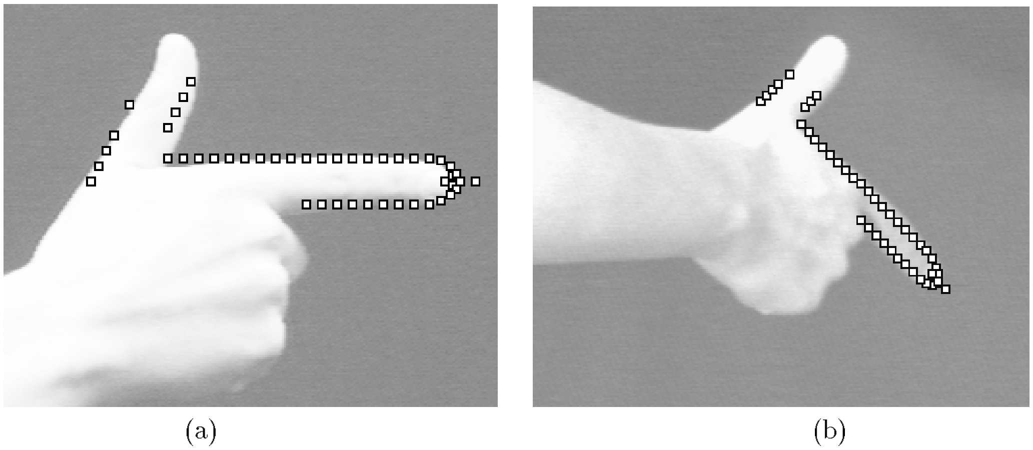


Figure 3: Arrangement of cameras, table and robot

3.2 Tracking Mechanism

We use a type of active contour tracker [5] to track the image of a hand in the familiar 'pointing' gesture. The tracker is based on a template, representing the shape of the occluding contours of an index finger and thumb (figure 4). About 50 *local edgefinders* (represented in the figure by dots) continuously measure the normal offset between these predicted contours and actual features in the image; these offsets are used to update the image position and orientation of the tracker.



(a)

(b)

Figure 4: 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).

The tracker's motion is restricted to 2D affine transformations in the image plane, which ensures that it keeps its shape whilst tracking the finger in a variety of poses [8] (this approach is best suited to tracking small planar objects, but also works well with fingers, which are cylindrical). Additional constraints bias it in favour of rigid motions in the image plane and limit the rate at which it can change scale. These constraints represent prior knowledge of how the hand's image is likely to move, and increase the reliability with which it can be tracked.

The pointing direction is assumed to be the orientation of the index finger; the base of the thumb is tracked merely to resolve an *aperture problem* [9] induced by the finger's long thin shape. We have deliberately avoided tracking the main part of the hand because this has a complicated shape which can vary significantly from one person to another.

3.3 Experiments

Pointing Experiment

In this experiment, the corners of a coloured rectangle on the tabletop are used to define the working coordinate system. The pair of finger-trackers (one for each camera) are 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 tabletop. In our implementation, the position and orientation of the finger, and the indicated point on the plane, are updated about 10 times per second.

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. Subjective estimates of accuracy are in the order of 2–4cm.

Robot guidance experiment

For this experiment, the reference points are defined by observing the robot gripper as it visits 4 points in a plane (this not only defines the working coordinate system but relates it to the robot's own world coordinate system). Finger-trackers operate as before, but now the robot is instructed to move repeatedly to where the hand is pointing, providing the operator with direct feedback of the system's output.

By observing this feedback from the robot, the operator is able to position the gripper to within 1cm: sufficient accuracy to instruct it to pick up a small wooden block standing in its workspace (figure 5).

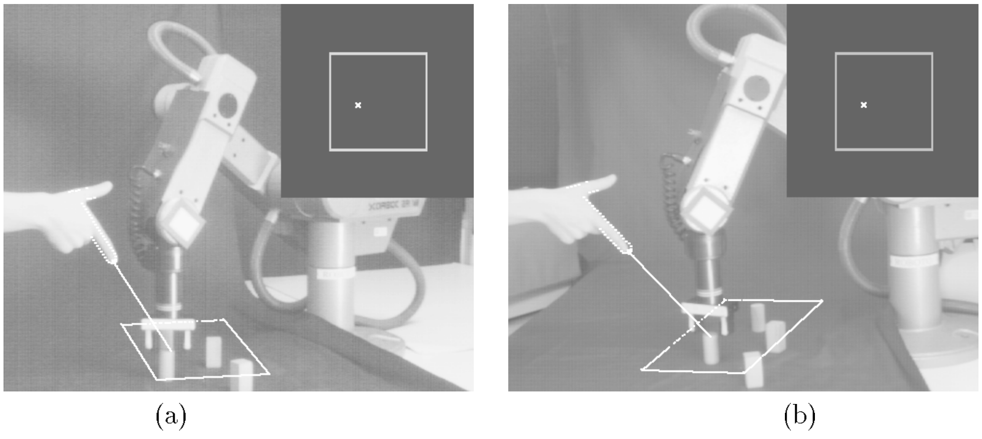

(a) (b)

Figure 5: Gestural control of robot position for grasping, seen in stereo. The four reference points (white rectangle) were defined by the robot's gripper in a plane 50mm above the table.

3.4 Accuracy evaluation

Finger trackers

We can derive 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. This takes into account the effects of image noise and occlusion, as well as pixel quantisation effects. Image position of the finger's midline can be measured to sub-pixel accuracy (standard deviation estimate typically 0.3 pixels), and the orientation to an uncertainty of about 0.6°.

From this uncertainty measure we calculate $\pm 2\sigma$ perturbations of the position and orientation of lines l_i and l'_i; and, by projecting these onto the ground plane, estimate the uncertainty in the position of the indicated point. Figure 6 shows the results for different camera configurations, with a 95% confidence ellipse around the indicated point: 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 the constraint lines meet at a greater angle and accuracy is improved (figure 6c).

Reference point coordinates

For the first experiment, in which the reference points are defined by hand, we assume an uncertainty of 1 pixel standard deviation. 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 that the tracker uncertainty. Again, the errors are most evident when the camera separation is small.

Angle between the cameras	Working plane coordinate error (with finger tracker noise)	Working plane coordinate error (with reference point noise)	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, σ derived from 'residual offsets'; (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).

4 Conclusion

This method for resolving the pointing direction proves to be usable and stable in the presence of normal image noise. The uncertainty in locating the indicated point depends on the camera configuration, and is most accurate when they are at least 30° apart. The system does not require full camera calibration because all calculation takes place in the image and ground planes. By tracking 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 constrast between the hand and the background. Our system also 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 purpose-built image processing equipment.

With visible feedback to the operator (e.g. having a robot follow the pointing hand in real time), points can be indicated to a resolution of about 1cm: sufficient accuracy to guide simple pick-and-place operations.

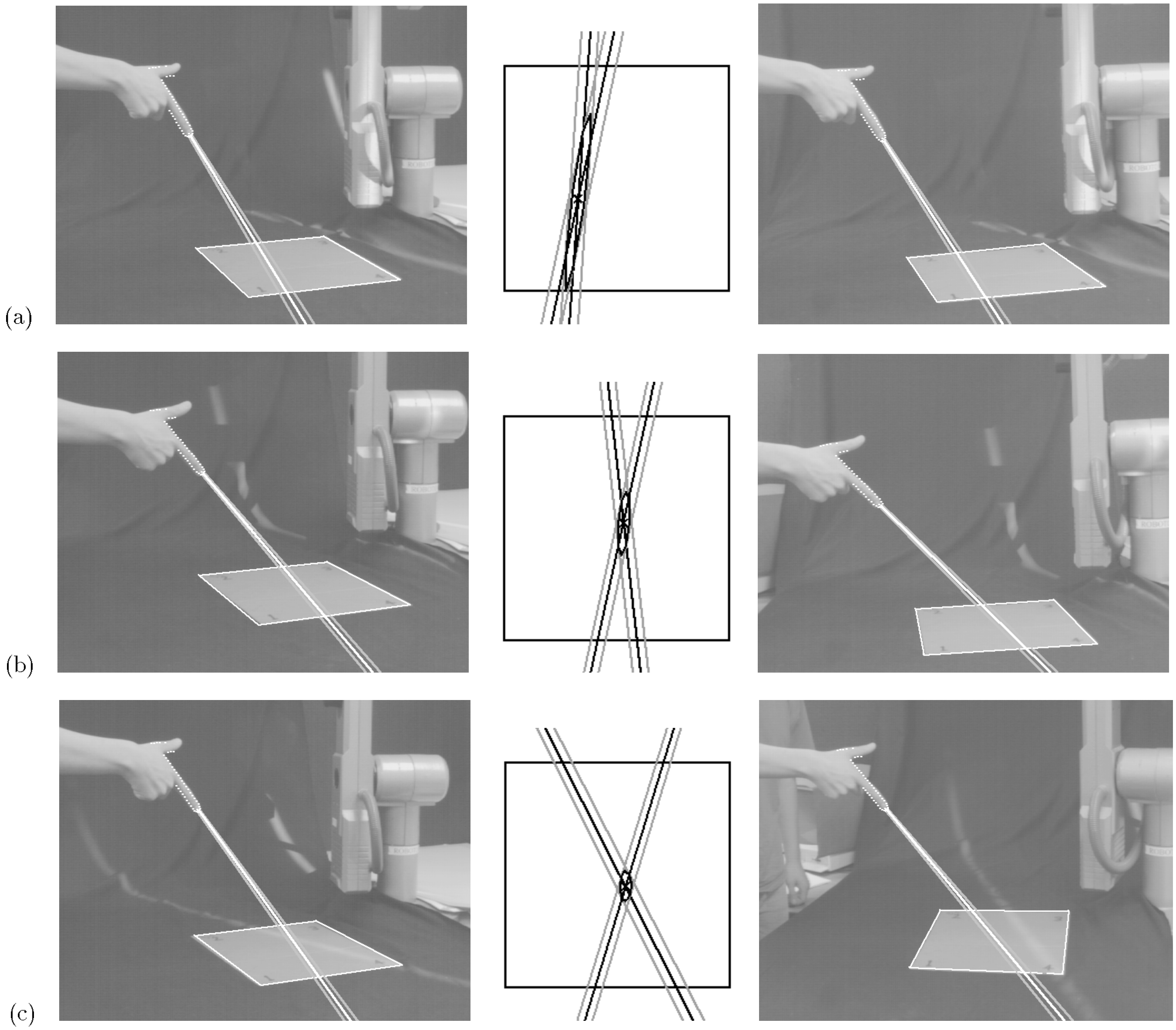


Figure 6: 2σ bounds for the pointing lines and their projections into working plane coordinates, when the angle between stereo views is (a) 7° (b) 16° (c) 34°. The uncertainty is greatest when the angle is small and the constraint lines nearly parallel.

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 algorithm.

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

References

[1] R. A. Bolt. 'Put-that-there': Voice and gesture at the graphics interface. *ACM-SIGGRAPH*, vol. 14 no. 3, pp 262–270, 1980.

[2] D. Wiemer and S. G. Ganapathy. A synthetic visual environment with hand gesturing and voice input. *Proc. CHI'89*, pp 235-240, 1989.

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

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

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

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

[7] L. Quan and R. Mohr. Towards structure from motion for linear features through reference points. *Proc. IEEE Workshop on Visual Motion*, 1991.

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

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