

Real-time Visual Recovery of Pose using Line Tracking in Multiple Cameras

David W Murray, Ian D Reid and Richard L Thompson
Department of Engineering Science, University of Oxford,
Parks Road, Oxford, OX1 3PJ, UK
[dwm|ian|rlt]@robots.ox.ac.uk

Abstract

This paper describes a system for the recovery in real-time of the pose of moving polyhedral objects using modest hardware. A method of line tracking first introduced by Harris is extended to multiple calibrated cameras, and afforded by robust methods and filtering. The system, which uses three cameras a low-end commercial framegrabber and runs on single PC, has been devised to provide simple visual feedback for the tele-operator of a force reflecting robot manipulator. Experimental results are given which demonstrate the accuracy of the vision system.

1 Introduction

The aim of installing a vision system in a teleoperated workcell is to provide the operator with a simple synthetic view of objects as they move within the workcell, with the view point and view direction under operator control. The overall goal is to demonstrate that human operators can use visual feed-forward to stabilize otherwise marginally stable tasks. Two important choices in the design of such a system are, first, whether to use data-driven structural recovery methods or model-based methods and, secondly, whether to use calibrated vision to recover Euclidean structure or un- or partially-calibrated vision to recover structure modulo unknown projective or affine transformations. For the latter choice, whilst there are certainly tasks in hand-eye coordination for which uncalibrated vision is quite adequate (see for example [1, 3]), the need to function in a Euclidean robot work space and the a priori undefined nature of the operator's tasks suggest that the calibrated route might be more embracing, though more inflexible. For the former choice, though it is in the long term to recover structure before modelling, the need for some robustness and real-time performance on modest hardware still point to a purely model-based approach.

The visual tracking of 3D rigid objects may be subdivided into several sub-problems: first, how to recognize the object; secondly, how to initialize its pose; and thirdly, how to update its pose over time. In this paper we are mainly concerned with the last problem. For the first and second we assume that objects in the scene are recognized by the human operator, who chooses them from a library, and who can initialize the object's pose.

For the third, the method we advance here is that of Harris [4], allowing it to be used with multiple cameras. 3D objects are modelled by a set of control points which lie on edges, which may be either surface creases or surface albedo markings, allowing the corresponding lines to be detected in the image. The method assumes any pose change required between the current estimated pose and the actual pose is sufficiently small (i)

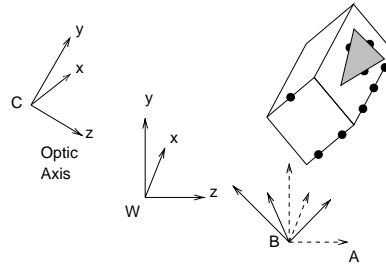


Figure 1: The object is modelled within its own coordinate frame B as a set of control points lying on edges, which may be crease or albedo edges. The world and camera coordinate frames are W and C. The aligned frame A, introduced as a notational convenience, has the same orientation as W but has the same origin as B.

to allow a linearization of the solution, and (ii) to alleviate the problem of finding line to line correspondences. The correspondences used are between predicted point to measured image line, allowing search in 1D rather than 2D within the image.

The paper is ordered as follows. The next section defines the coordinate systems and describes the method of updating pose for multiple cameras. Section 3 gives details of the methods used to determine and filter pose robustly, to initialize pose and to calibrate the system. Section 4 gives experimental results which show the accuracy to which pose can be recovered using this real-time method, and Section 5 gives the results of manipulation experiments.

2 Updating the pose of objects

2.1 Coordinate frames

As noted above, each model is described in its own object frame B by the coordinates of a set of control points. The points may be special features (eg point lights), but more usually are specified to lie on object edges which are geometrically fixed as either crease edges between surfaces or albedo markings on surfaces, as sketched in Figure 1. The underlying object need not be polyhedral.

As sketched in Figure 1, an object's pose is one or another representation of the the transformation $\{R_B^W, t_{BW}\}$ that takes points in frame B into points into a fixed world frame W. For example, using a non-homogeneous rotation matrix and translation representation $X^W = R_B^W X^B + t_{BW}$ where t_{BW} denotes the origin of B in W. The pose of a camera C relative to the world coordinate system is defined in the inverse manner as the rotation and translation $\{R_W^C, t_{WC}\}$ where $X^C = R_W^C X^W + t_{WC}$. It is a matter of convenience below to introduce an aligned frame A that is aligned with W but has its origin coincident with the object frame B: $X^A = R_B^W X^B$.

The method of establishing and calibrating the coordinate frames is deferred to Section 3.

2.2 Recovering change of pose

The image operations are of the sort routinely used in visual contour tracking. As each new image is captured, the current estimate of pose (or the predicted estimate if the rate of change of pose is modelled) is used to project the visible control points and their lines onto the image, and a search is initiated from each control point in a direction perpendicular to

the projected line in order to find any point on the actual imaged line in the new image. The new pose is estimated by minimizing the control point to image line displacements.

Consider an object whose position at some instant is described in the world frame by

$$\mathbf{X}^W = \mathbf{R}_B^W \mathbf{X}^B + \mathbf{t}_{BW} = \mathbf{X}^A + \mathbf{t}_{BW} .$$

The object's angular and rectilinear velocities are $\boldsymbol{\Omega}$ and \mathbf{v} respectively, so that in a small time $\delta\tau$ the object will move to a new position

$$\mathbf{X}^{W'} = \mathbf{X}^A + (\boldsymbol{\Omega}\delta\tau)\wedge\mathbf{X}^A + \mathbf{t}_{BW} + \mathbf{v}\delta\tau .$$

By writing the product of the time interval and velocity screw as

$$\delta\mathbf{s} = \delta\tau[v_x, v_y, v_z, \Omega_x, \Omega_y, \Omega_z]^\top$$

and composing the matrix

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & Z^A & -Y^A \\ 0 & 1 & 0 & -Z^A & 0 & X^A \\ 0 & 0 & 1 & Y^A & -X^A & 0 \end{bmatrix}$$

the new position can be written as

$$\mathbf{X}^{W'} = \mathbf{X}^W + \mathbf{G}\delta\mathbf{s} .$$

To obtain the measurement equation, we need to transform the new positions into the camera frame, and thence project into the image. The camera is an idealized device with unity aspect ratio and unity focal length. In the camera frame, the modified coordinates are related to the originals by

$$\mathbf{X}^{C'} = \mathbf{X}^C + \mathbf{R}_W^C \mathbf{G}\delta\mathbf{s} .$$

The projection into the idealized image¹ is $\mathbf{x} = -\mathbf{X}^C/Z^C$ (where we will keep \mathbf{x} as a 3-vector $(x, y, -1)^\top$ for the moment). The change in image coordinate as the object is moved is thus

$$(\mathbf{x}' - \mathbf{x}) = -\left[\mathbf{X}^{C'}/Z^{C'} - \mathbf{X}^C/Z^C\right] .$$

If the change in depth is small, so that $Z^{C'} = (1 + \alpha)Z^C$, with $\alpha \ll 1$, then

$$(\mathbf{x}' - \mathbf{x}) \approx -\frac{1}{Z^C} \left[\mathbf{X}^{C'} - \mathbf{X}^C - \alpha\mathbf{X}^{C'}\right] = -\frac{1}{Z^C} \left[\mathbf{R}_W^C \mathbf{G}\delta\mathbf{s} - \alpha(\mathbf{X}^C + \mathbf{R}_W^C \mathbf{G}\delta\mathbf{s})\right] .$$

But α may be expressed as

$$\alpha = (Z^{C'} - Z^C)/Z^C = Z^{C-1}[001]\mathbf{R}_W^C \mathbf{G}\delta\mathbf{s}$$

and so

$$(\mathbf{x}' - \mathbf{x}) \approx -\frac{1}{Z^C} \left[\mathbf{R}_W^C \mathbf{G}\delta\mathbf{s} - \frac{1}{Z^C}[001]\mathbf{R}_W^C \mathbf{G}\delta\mathbf{s} \mathbf{X}^C\right] = -\frac{1}{Z^C} \left[\mathbf{R}_W^C \mathbf{G}\delta\mathbf{s} + [001]\mathbf{R}_W^C \mathbf{G}\delta\mathbf{s} \mathbf{x}\right] .$$

¹The minus sign sets the ideal image plane behind the optic centre.

The above could be used to recover the change of pose if point to point matches could be established between several points \mathbf{x} and their correspondences \mathbf{x}' , for example when the control points are corners or lights. However in our work the control points used typically lie on lines or curves, and we thus only obtain information on point-to-line or point-to-curve matches because of the aperture problem — we know that \mathbf{x}' lies on a particular line or curve, but not exactly where.

All the information is preserved if the inner product is formed between $(\mathbf{x}' - \mathbf{x})$ and the unit normal $\hat{\mathbf{n}}$ to the curve or line at the point \mathbf{x} . The measurement equation becomes

$$\hat{\mathbf{n}}^\top (\mathbf{x}' - \mathbf{x}) = -\frac{1}{Z^C} \hat{\mathbf{n}}^\top [\mathbf{I}_3 + \mathbf{x}[001]] \mathbf{R}_W^C \mathbf{G} \delta \mathbf{s}$$

or, using 2-vectors for image quantities,

$$\hat{\mathbf{n}}^\top (\mathbf{x}' - \mathbf{x}) = -\frac{1}{Z^C} \begin{bmatrix} n_x & n_y \end{bmatrix} \begin{bmatrix} 1 & 0 & x \\ 0 & 1 & y \end{bmatrix} \mathbf{R}_W^C \mathbf{G} \delta \mathbf{s}$$

As Figure 2 illustrates, $\hat{\mathbf{n}}^\top (\mathbf{x}' - \mathbf{x})$ is the perpendicular distance between the curves, which we will assume has a radius of curvature much greater than this distance. Note that the choice of the positive direction of $\hat{\mathbf{n}}$ is arbitrary.

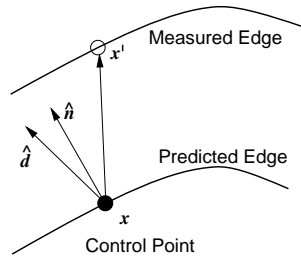


Figure 2: Because of the aperture problem, only the perpendicular distance along $\hat{\mathbf{n}}$ is measurable. It is not necessary to search for the edge along $\hat{\mathbf{n}}$: it is quicker to search along one of the eight cardinal directions $\hat{\mathbf{d}}$, here a diagonal.

2.3 Extension to several cameras

Because the pose updating is model-based, there is no particular need when extending the method to multiple cameras to establish correspondence between control points, and each camera can be treated independently in terms of measurement. The independence of the cameras has no particular impact on image search, as it is already 1-dimensional. Moreover, because the image search is between point and a *line* and involves the aperture problem, there is no possibility of reducing the search to zero dimensions.

A second benefit of independence between views is that it provides basic robustness to complete obscuration and loss of image without recourse to costly checking procedures. This is of value because it allows *unmodelled* objects to enter the workspace.

With several points i and cameras c we form the system

$$\begin{bmatrix} \vdots \\ \mathbf{a}_{ic} \\ \vdots \end{bmatrix} \delta \mathbf{s} = \begin{bmatrix} \vdots \\ m_{ic} \\ \vdots \end{bmatrix}$$

where $m_{ic} = \hat{\mathbf{n}}_{ic}^\top (\mathbf{x}'_{ic} - \mathbf{x}_{ic})$ and \mathbf{a}_{ic} is the row vector

$$-\frac{1}{Z_i^c} \hat{\mathbf{n}}_{ic}^\top [\mathbf{I}_3 + \mathbf{x}_{ic} [001]] \mathbf{R}_W^c \mathbf{G}_i$$

This linear system, $\mathbf{A}\delta\mathbf{s} = \mathbf{m}$, is solved by first applying a robust estimator (least median of squares) to eliminate outliers, and then using singular value decomposition on the remaining inliers.

3 Implementation

The above method has been implemented for polyhedral objects viewed by up to three cameras. The three monochrome video streams are synchronized and captured as the RGB planes of a PCI bus colour framegrabber. The framegrabber is hosted by a 166 MHz Pentium running under the QNX real-time OS.

Exploitation of this relatively simple method of pose updating relies on finding good point to line matches in the image, which in turn requires methods to be in place for calibration, pose initialization, image search, and pose filtering.

As the last of these is most closely related to pose updating, we deal with this first, and then take the others in order.

3.1 Robust determination and filtering of absolute pose

As noted above the linear system $\mathbf{A}\delta\mathbf{s} = \mathbf{m}$, is solved by first applying a robust estimator (least median of squares) to eliminate outliers, and then using singular value decomposition on the remaining inliers.

In a number of comparative tests in the vision literature, random sampling methods have proved the most successful for robust estimation of properties. Here, because the standard deviation is not known *a priori*, robust estimation is performed using Rousseeuw's Least Median of Squares (LMS) algorithm, rather than Fischler and Bolles' RANSAC. In repeated trials, the minimal groups of 6 matches are randomly selected to determine a value for $\delta\mathbf{s}$, and this value used to determine the median of the magnitudes of the deviations $e_i = |m_i - m_i^{\text{fitted}}|$. The solution with the smallest median is used to estimate the standard deviation of the data exploiting the fact that $\sqrt{\text{med}|e_i|}/\Phi^{-1}(0.75)$ is an asymptotically consistent estimator of σ when the e_i are distributed like $N(0, \sigma^2)$, where Φ is the cumulative distribution function for the Gaussian pdf.

$$\sigma = 1/\Phi^{-1}(0.75) \sqrt{\text{med}|e_i|} = 1.48 \sqrt{\text{med}|e_i|} .$$

Then measurements are split between inliers and outliers using

$$i \in \begin{cases} \text{inliers} & \text{if } |e_i| \leq 1.96\sigma \\ \text{outliers} & \text{otherwise} \end{cases} .$$

In experiment, the σ derived was typically of order 0.03 in the ideal image with focal length unity. Our cameras have focal length around 3000 pixels. Now $|e_i|$ is a measure of the distance from the predicted edge to the actual edge. Using the Rousseeuw formula in reverse we find $|e_i| \sim 1.2$ pixel in the physical image. This is entirely commensurate with the edge search mechanism, which operates only to ± 1 pixel accuracy. The outliers are excluded to the final least squares fit for the change in pose $\delta\mathbf{s}$. The fit is made using singular value decomposition.

The *change* in pose is used to recompute *absolute* pose. Whilst translation requires a simple addition,

$$\mathbf{t}(\tau + \delta\tau) = \mathbf{t}(\tau) + \delta\tau\mathbf{v} ,$$

updating the angle-axis to better than first order is more involved. It is of course possible to write down angle-axis update rules, but they can be expressed using standard quaternion notation as

$$\begin{aligned} \delta\hat{\mathbf{q}} &= (\cos(|\delta\mathbf{a}|/2), \sin(|\delta\mathbf{a}|/2)\hat{\delta\mathbf{a}}) \\ \hat{\mathbf{q}}(\tau + \delta\tau) &= \delta\hat{\mathbf{q}} \cdot \hat{\mathbf{q}}(\tau) . \end{aligned}$$

from which $\mathbf{a}(\tau + \delta\tau)$ is found by back-transformation.

In the present implementation each new absolute pose measurement is combined with a running estimate using a constant velocity Kalman Filter with twelve components in the state

$$\mathbf{p}_\tau = (t_x, t_y, t_z, a_x, a_y, a_z, \dot{t}_x, \dot{t}_y, \dot{t}_z, \dot{a}_x, \dot{a}_y, \dot{a}_z)^\top .$$

Although similar to other filters for pose tracking in the literature, its performance has proved less than satisfactory in our teleoperative applications. More recent studies by Heuring and Murray [6, 7] show that muscular motions (specifically those of the human head) are better modelled by describing the velocity as Ornstein-Uhlenbeck process, rather than the Wiener process implicit in the constant velocity model.

3.2 Calibrating the system

The location of the world coordinate frame \mathbf{W} is arbitrary, but to gain benefits from statistical centring its origin should be placed near to the point of closest approach of the optic axes of the cameras being used to view the scene. (There is of course no requirement for the optic axes to meet at a point.) To establish \mathbf{W} 's location with respect to camera j , and to derive the imaging properties of camera j , we need to measure the image coordinates \mathbf{x}_i^j of a number of non-coplanar points whose 3D positions $\mathbf{X}_i^{\mathbf{W}}$ are specified in the world coordinate frame, and to minimise the image error

$$\min_{\mathbf{u}_j} \sum_i \left[\mathbf{x}_i^j - \mathbf{x}(\mathbf{u}_j, \mathbf{X}_i^{\mathbf{W}}) \right]^2$$

After compensating for radial distortion and neglecting the skew between image axes, the parameter vector \mathbf{u} has ten degrees of freedom, comprising four intrinsic camera parameters (focal lengths and principal point f_x, f_y, x_0, y_0) and six extrinsic parameters (three rotational represented by angle-axis vector \mathbf{a} and three translational components \mathbf{t}):

$$\mathbf{u} = (f_x, f_y, x_0, y_0, a_x, a_y, a_z, t_x, t_y, t_z)^\top .$$

A starting estimate for this non-linear minimization is obtained using a QR decomposition of the projection matrix ([5], itself found using the usual linear methods. Care is taken to statistically centre the scene and image data prior to fitting.

Two methods of specifying the world points are used. The first, on which the results in this paper are based, uses the usual chequerboard pattern imposed on a cube. The second uses the telerobotic arm as a moving pointer.

3.3 Model pose initialization

Once added to the list of active objects, pose is initialized using stereo in any two of the three available views, with correspondence specified by the operator. More recent work by Heuring [6] uses search constrained by motion.

4 Experimental results

Figure 3 shows views from the three cameras of a polyhedral object in the workspace. The axes shown in the view from camera (0) are the object axes. Experiments are performed (i) to explore the relative and absolute accuracies of pose recovery, (ii) to derive the frequency response, and (iii) to explore the efficacy of the virtual view in a manipulation task.

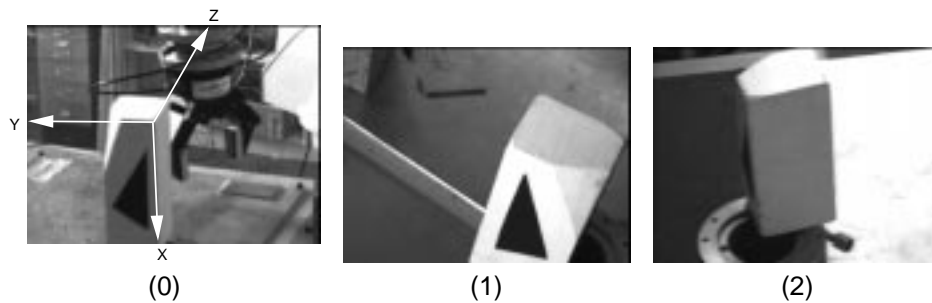


Figure 3: Views from cameras 0, 1 and 2.

4.1 Relative accuracy of single and multiple cameras

The object's x -axis was first aligned with the world frame's x -axis. The object was then rotated about this axis in steps of 10° and the rotational and translational components of pose recovered.

The rotational results for three cameras and one camera are given in Figures 4(a) and (b) respectively. To obtain a measure of error, each point and error bar in the graph represent the average and standard deviation of four measurements. For three cameras, the slope of the plot of recovered rotation angle vs. set rotation angle was 1.005 ± 0.009 , and the mean change in rotation between steps was found to be $(10.1 \pm 2.1)^\circ$. (The zero offset of 33° arises trivially because no attempt was made to align the y - and z -axes.) For the single camera (camera 1), pose recovery fails at about 140° because of lack of data — but this is particular to this example. Of more general interest is that there is increased error in the recovered rotation, but not dramatically so: the mean change in angle is $(9.6 \pm 4.0)^\circ$.

More revealing is the comparison between recovered translational parameters, shown in Figure 5. For three cameras, the t_y and t_z are, as expected, close to constant and zero throughout, and t_x is close to a constant offset. (Again the offset arises trivially because no attempt was made to align the origins of the x -axes.) However, the components recovered from camera 0 alone are evidently erroneous but highly correlated.

Taking the values recovered in the three-camera experiment as veridical, the error in the translation δt_j was found over the rotation range -20° to $+130^\circ$. The eigenvalues of

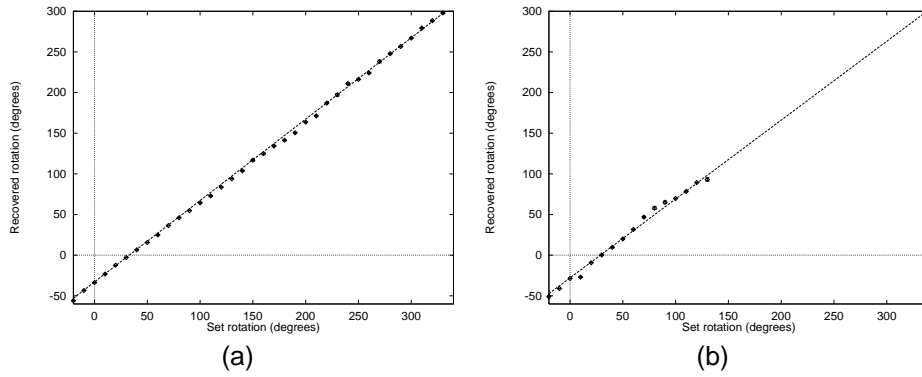


Figure 4: Recovered rotation versus set rotation for (a) 3 cameras and (b) 1 camera.

the scatter matrix

$$M = \sum_j [\delta \mathbf{t}_j - \bar{\delta \mathbf{t}}][\delta \mathbf{t}_j - \bar{\delta \mathbf{t}}]^T$$

were found as $\lambda_1 = 1.2 \times 10^{-1}$, $\lambda_2 = 2.6 \times 10^{-1}$, and $\lambda_3 = 1.9 \times 10^{+2}$, indicating that the translation was determined with similar accuracy along the directions of eigenvectors \mathbf{e}_1 and \mathbf{e}_2 , but with much poorer accuracy along the direction of the third eigenvector in the W frame, $\mathbf{e}_3 = (+0.711, -0.428, +0.558)$. Multiplying this by the rotation matrix R_W^C for the single camera (camera 1) gives the direction in the camera's frame as

$$\mathbf{e}_3^C = (-0.201, -0.416, +0.980).$$

This, as expected, is close to the optic axis of the single camera.

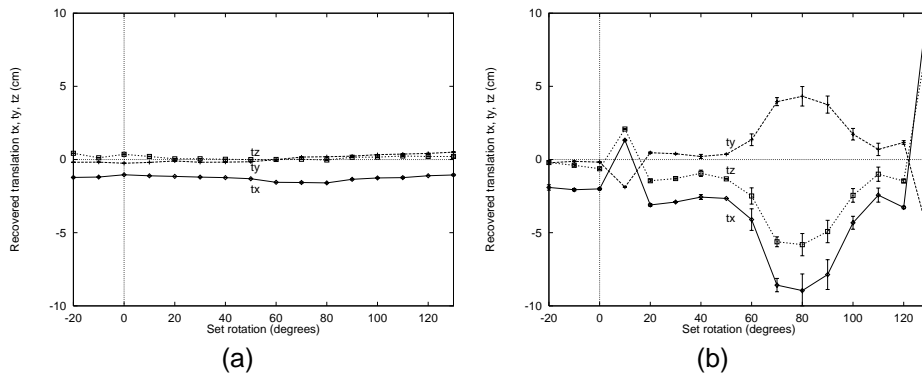


Figure 5: Recovered translation components versus set rotation for (a) 3 cameras and (b) 1 camera to the same scale.

A second comparative run was made using translational rather than rotational movement. The object was moved some 200 mm in steps of (10 ± 0.5) mm along a straight line close to the world frame's z -direction and certainly in the world's y, z -plane. Figures 6(a) and (b) compare the recovered translation components for 3 cameras and 1 camera

respectively. Again, each point gives the mean and standard deviation of several measurements. For three cameras, the step size recovered was (9.8 ± 0.9) mm, whereas that for one camera was (9.2 ± 4.2) mm. Assuming the values obtained using three cameras to be veridical, errors were computed for the one camera case and, using the same eigen-analysis, the errors found to lie predominantly along the optic axis of the single camera.

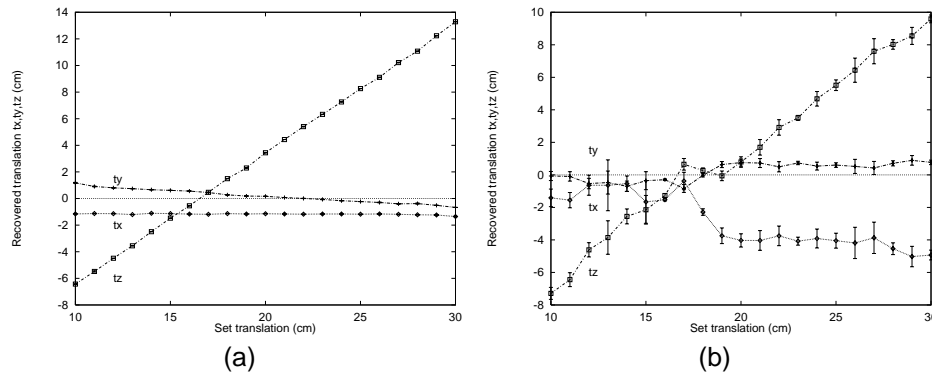


Figure 6: Recovered translation components versus set translation for (a) 3 cameras and (b) 1 camera to the same scale.

4.2 Teleoperation experiments

The system has been used for a variety of simple manipulation experiments. Figure 7(a) shows views of the recovered wireframes of a peg and hole during an insertion task and Figure 7(b) shows the inserted peg from a number of view points. The obvious misalignment is real, and not systematic noise — the peg has sufficient clearance around it when in the hole to twist by several degrees.

5 Discussion

We have described a system for the recovery in real-time of the pose of moving objects using modest hardware. We make use of multiple calibrated cameras which we have demonstrated gives a significant improvement in the accuracy of the recovered pose over single camera methods such as [4, 9].

In the context of teleoperations the system provides two major benefits. In many cases, the information required for remote manipulation is not so much dependent on the quality of visual representation as upon *viewpoint*. Our system provides a prompt and rapidly updated representation which can be used to generate novel viewpoints to assist the operator.

Furthermore, visual data provides information which is complementary to the force data which is typically available. In particular, the accurate position and velocity information can be used as a *predictor* of impending collisions. For teleoperative force-feedback systems, the aim of an impact controller is to allow the demands of the operator to be followed, which are often designed to impact with the environment, but to prevent a large impact transient. Some previous work has addressed the issue of transition control [8, 10]. We are currently investigating the use of our vision system with recent results due to

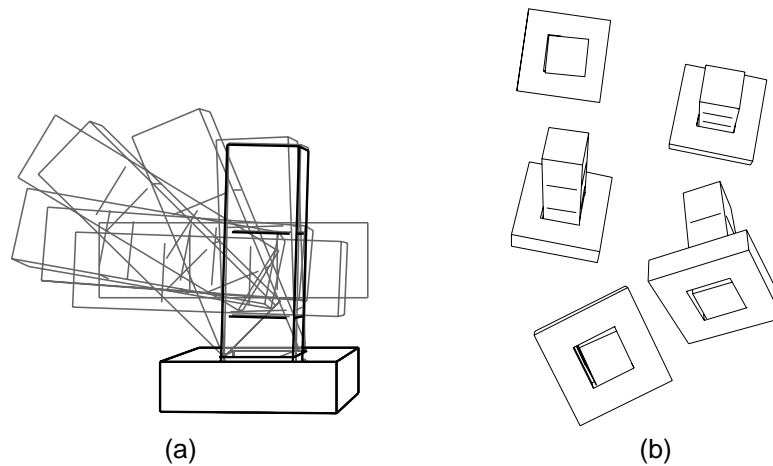


Figure 7: (a) Recovered wireframes during an insertion task. (b) Views of the peg in the hole.

Daniel and McAree [2] who showed how a transition controller can be designed which uses vision as a feed-forward sensor during transition control such that the robot trajectory is modified to maximise the performance of the system yet leave the force response unchanged (controlled by a tele-operator once impact has occurred).

Acknowledgements

This work is supported by EPSRC Grant GR/L15005 and RLT is supported by and EPSRC Research Studentship.

References

- [1] R Cipolla and N J Hollinghurst. Human-robot interface by pointing with uncalibrated stereo vision. *Image and Vision Computing*, 16(1):171–178, 1996.
- [2] R. W. Daniel and R. Mcaree. Transition control. Technical report, Department of Engineering Science, University of Oxford, 1997.
- [3] G D Hager, S Hutchinson, and P Corke. A tutorial introduction to visual servo control. *IEEE Transactions on Robotics and Automation*, 12(5):661–670, 1996.
- [4] C Harris. *Tracking with Rigid Models*, pages 59–73. MIT Press, Cambridge MA, 1992.
- [5] R. I. Hartley. Self-calibration from multiple views with a rotating camera. In *Proc. 3rd European Conf. on Computer Vision, Stockholm*, volume 1, pages 471–478, 1994.
- [6] J.J. Heuring. *Merging Computer Telepresence and Computer Vision*. PhD thesis, Department of Engineering Science, University of Oxford, 1998.
- [7] J.J Heuring and D.W. Murray. Modelling velocities as an Ornstein-Uhlenbeck process for 3d pose tracking. Submitted to IEEE Trans on PAMI, 1998.
- [8] Y. F. Li. A sensor-based robot transition control strategy. *International Journal of Robotics Research*, 15(2):128–136, April 1996.

- [9] D. G. Lowe. Robust model-based motion tracking through the integration of search and estimation. *International Journal of Computer Vision*, 8(2):113–122, August 1992.
- [10] B. J. Nelson and P. K. Khosla. Force and vision resolvability for assimilating disparate sensory feedback. *IEEE Trans. Robotics and Automation*, 12(5):714–731, October 1996.