3D Model Acquisition by Tracking 2D Wireframes

M. Brown, T. Drummond and R. Cipolla
{96mab|twd20|cipolla}@eng.cam.ac.uk

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

Abstract

This paper presents a semi-automatic wireframe acquisition system. The system uses real-time (25Hz) tracking of a user specified 2D wireframe and intermittent camera pose parameters to accumulate 3D position information. The 2D tracking framework enables the application of model-based constraints in an intuitive way which interacts naturally with the Kalman filter formulation used. In particular, this is used to introduce feedback from the current 3D shape estimate to improve the robustness of the 2D tracking. The scheme allows wireframe models of simple edge based objects to be built in around 5 minutes.

1 Introduction

Computer vision can be used to obtain 3D models of objects for use in graphics, reverse engineering and for applications such as model-based tracking. One method is to acquire a dense set of depth measurements e.g. from optical flow and structure from motion techniques, or laser range finding. However, many applications require more efficient representations than a simple depth map. Hence, the data must be segmented e.g. into planar or quadric surface patches [1], and constraints such as orthogonality imposed in the high-level description [2]. Multi-view, feature based techniques overcome the problem of segmentation by computing the depth of salient points only, thereby assuming a simple underlying representation. Matching of points between images is typically achieved by a feature extraction step, followed by application of geometric constraints. For example, corner features might be extracted using the Harris corner detector, and the fundamental matrix used for point matching between 2 views [3, 4]. Alternatively, edges might be extracted using the Canny edge detector, and the trifocal tensor used for line matching between 3 views [5]. Snake tracking reduces the complexity of maintaining correspondence by searching for edges in a local window [6]. Edges have the advantage that they can be rapidly tracked, using multiple 1D searches (perpendicular to the edge), rather than using 2D search e.g. for corners. This approach is utilised in this work to allow frame rate (25Hz) 2D tracking as a mechanism for preserving correspondence.

Previous approaches to model acquisition using tracking have used single line segments [7, 8]. This paper shows how nets of connected line segments – or *2D wireframes*,

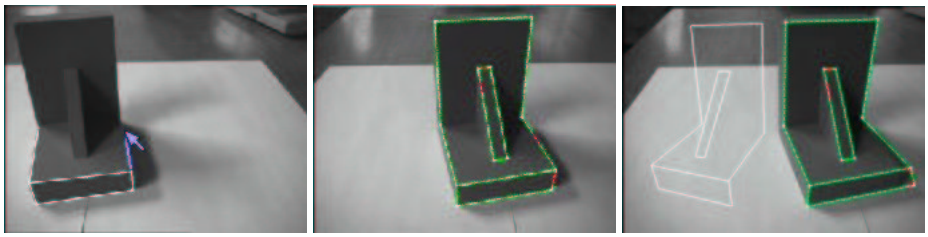


Figure 1: Typical operation of the model acquisition system – user input, tracking, reconstruction

can incorporate high level user constraints to reduce the number of degrees of freedom and hence improve tracking. Kalman filtering is used to accumulate 3D information from observations made with noisy sensors, as described in [9]. This information is used in a second Kalman filter, which accumulates 2D tracking information. In contrast to the independent covariance matrices maintained for features in systems such as [7, 8, 10], this filter uses a *fully covariant* representation. This allows the application of a rich set of model-based constraints upon the tracking, utilising the 3D information.

A significant motivation for this work has been the production of models for model-based tracking systems such as [11, 12]. These systems make use of 6 degree of freedom Euclidean motion constraints to enable robust tracking in the presence of noise. The 'model acquisition system' presented here may be seen as a method for bootstrapping model-based tracking, by combining it with model building.

2 Model Acquisition System

An example of typical system operation is shown in figure 1. A robot mounted camera, with known internal parameters, provides video images of the object. The user specifies the wireframe to be acquired via a point and click interface. The user then moves the robot, after which it provides an update of the pose of the camera. The model acquisition system tracks the object, and uses the sparse camera updates to accumulate 3D position information. The system output is a 3D wireframe model of the object. In our case the objects of interest are composed of straight line segments and may be represented by straight line 3D wireframes.

A naive approach to this problem is to consider tracking and model building as separate tasks, and we present this first. They are, in fact, very much intertwined – an improved 3D model enables improved tracking, which in turn improves the 3D model. By also allowing user input in the loop, previously unseen parts of the model can be reconstructed. This facilitates a framework for continuous building and tracking of 3D objects, bootstrapping a 3D model from high level user constraints. This is described in the remainder of the paper.

2.1 2D Wireframes

Object edges, which appear as intensity discontinuities in the video image, are used for tracking. Edges have the advantage that they can be rapidly tracked, using multiple 1D

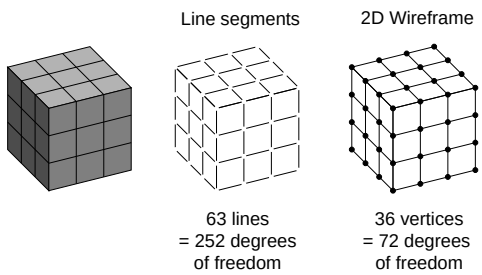


Figure 2: A Rubics Cube example

searches (perpendicular to the edge), rather than using 2D search e.g. for corners. Another advantage is that many measurements can be made along an edge, allowing it to be accurately localised.

A simple approach to object tracking is to use independent line segments. This has the disadvantages that lines in the epipolar plane cannot be localised due to the aperture problem, and that each new line segment adds 4 degrees of freedom. By connecting the line segments to form a 2D wireframe, these problems are reduced. In addition to enabling epipolar lines to be tracked, initialising the tracker as a 2D wireframe rather than a set of line segments causes a significant reduction in the number of degrees of freedom. For example, a Rubics cube from a general viewpoint has 63 visible edge segments and 36 visible vertices. Hence, a line segment representation has $63 \times 4 = 252$ degrees of freedom, but a 2D wireframe representation has only $36 \times 2 = 72$ degrees of freedom (see figure 2) – making tracking much easier.

2.2 Tracking

A simple approach to tracking is to use least squares – minimising the sum of the squared edge measurements from the wireframe (see figure 3). To formulate this as a linear least squares problem, the partial derivative of the edge measurements with respect to the vertex image positions is computed. The linear change in measurement d_i due to the change in vertex image position $\mathbf{w}_j = (u_j, v_j)^T$ is given by

$$\delta d_i = -\frac{l}{L} \delta \mathbf{w}_j \cdot \hat{\mathbf{n}}$$

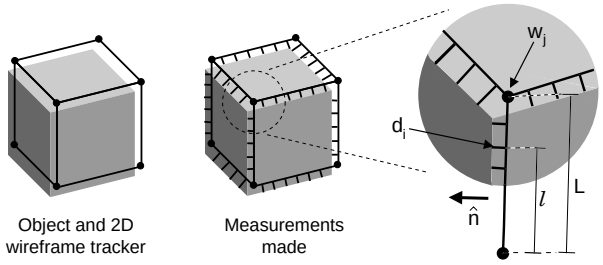


Figure 3: Edge measurements

Where l/L is the fractional distance of the measurement along the line, and $\widehat{\mathbf{n}}$ is the unit line normal. Hence

$$\frac{\partial d_i}{\partial \mathbf{w}_j} = -\frac{l}{L}\widehat{\mathbf{n}}$$

Stacking the vertex image motions $\delta\mathbf{w}_j$ into the P-dimensional vector $\mathbf{p}$, and the measurements into the D-dimensional vector $\mathbf{d}_0$ we may write

$$\mathbf{d} - \mathbf{d}_0 = \mathbf{M}\mathbf{p}$$

where $\mathbf{d}$ is the new measurement vector due to the motion $\mathbf{p}$, and $\mathbf{M} = \frac{\partial \mathbf{d}}{\partial \mathbf{p}}$ is the $D \times P$ measurement matrix. In our experiments, the number of measurements D is typically 100, and the number of wireframe vertices $P/2$ is typically 20. The overconstrained set of linear equations is now solved by least squares, minimising the sum squared measurement error $|\mathbf{d}|^2$. Note that in general the least squares solution $\mathbf{p}$ is not unique, it can contain arbitrary components in the right nullspace of $\mathbf{M}$, corresponding to displacements of the vertex image positions which do not change the measurements. Regularising by adding a small constant to the diagonal of $\mathbf{M}$ prevents instability, ensuring that there is a small measurement change for each direction in P space.

Note also that the least squares solution $\mathbf{p} = -(\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T\mathbf{d}_0$ is equivalent to a Maximum Likelihood estimate with the assumption that $\mathbf{d}$ is drawn from a zero mean, white Gaussian process. The likelihood probability density function (pdf) is a P-dimensional Gaussian with mean $\mathbf{m}_l = -(\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T\mathbf{d}_0$ and covariance matrix $\mathbf{C}_l = \mathbf{M}^T\mathbf{M}$.

2.3 Model Building

Tracking the 2D wireframe preserves the correspondence of the vertices, whose 3D position can be calculated from 2 views using simple triangulation. Observations from more than 2 views can be combined by maintaining a 3D pdf for each vertex $p(\mathbf{X})$. This is updated on the basis of the tracked image position of the point, and the known camera. New tracked image positions are calculated from the tracking step, and assumed to be correct up to white Gaussian noise in the image plane.

A 3D pdf that corresponds to this 2D pdf has surfaces of constant probability defined by rays through a circle in the image plane. We approximate this as a 3D Gaussian of infinite variance in the direction of the ray through the image point, and equal, finite, variances in the perpendicular plane (see figure 4). This is the likelihood of the tracked

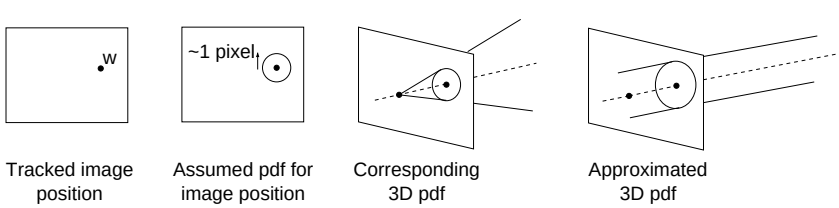


Figure 4: 3D pdf from observation

point position, conditioned on the current 3D position estimate – $p(\mathbf{w}|\mathbf{X})$, expressed in terms of 3D position with mean $\mathbf{m}_l$ and covariance matrix $\mathbf{C}_l$.

The 3D likelihood pdf is multiplied by the prior pdf to get the posterior pdf:

$$p(\mathbf{X}|\mathbf{w}) = \frac{p(\mathbf{w}|\mathbf{X})p(\mathbf{X})}{p(\mathbf{w})}$$

Since $\mathbf{X}$ is Gaussian with mean $\mathbf{m}_p$ and covariance matrix $\mathbf{C}_p$, and $\mathbf{w}|\mathbf{X}$ is Gaussian, mean $\mathbf{m}_l$, covariance matrix $\mathbf{C}_l$, $\mathbf{X}|\mathbf{w}$ is also Gaussian with mean $\mathbf{m}$ and covariance matrix $\mathbf{C}$, where

$$\begin{aligned} \mathbf{C}^{-1} &= \mathbf{C}_p^{-1} + \mathbf{C}_l^{-1} \\ \mathbf{C}^{-1}\mathbf{m} &= \mathbf{C}_p^{-1}\mathbf{m}_p + \mathbf{C}_l^{-1}\mathbf{m}_l \end{aligned}$$

In the next iteration, $p(\mathbf{X}|\mathbf{w})$ is used as an estimate for $p(\mathbf{X})$, therefore

$$\begin{aligned} \mathbf{C}_{n+1}^{-1} &= \mathbf{C}_n^{-1} + \mathbf{C}_l^{-1} \\ \mathbf{C}_{n+1}^{-1}\mathbf{m}_{n+1} &= \mathbf{C}_n^{-1}\mathbf{m}_n + \mathbf{C}_l^{-1}\mathbf{m}_l \end{aligned}$$

where n is the iteration number. These are the Kalman filter equations which are used to maintain 3D pdfs for each point.

3 Tracking and Model Building Combined

The processes of tracking and model building are very much intertwined – a 3D model can be used to improve tracking, which in turn provides an improved 3D model. Firstly we review the technique of model-based tracking using 6 degree of freedom Euclidean motion constraints. Secondly, we introduce a probabilistic framework that allows the weighted application of model-based constraints to the 2D tracking, and integrates naturally with the Kalman filter formulation.

3.1 Model-based 2D Tracking

A rigid body has 6 degrees of freedom corresponding to Euclidean position in space (3 translations and 3 rotations). A wireframe of $P/2$ points has a P-dimensional vector of image positions. The 6 degrees of freedom of Euclidean position correspond to a 6-dimensional manifold in this P-dimensional space. This can be linearised about the image position vector to give a 6D subspace of Euclidean motion. Projecting the image motion vector $\mathbf{p}$ onto this subspace constrains the image point motions to correspond to Euclidean motion of the object (or camera).

For a normalised camera moving with velocity $\mathbf{U}$ and rotating with angular velocity $\boldsymbol{\omega}$ about it's optical centre, the velocity of an image point $(\dot{u}, \dot{v})^T$ is given by

$$\begin{pmatrix} \dot{u} \\ \dot{v} \end{pmatrix} = \left[\frac{1}{Z_c} \begin{pmatrix} -1 & 0 & u \\ 0 & -1 & v \end{pmatrix} \quad \begin{matrix} uv & -(1+u^2) & v \\ 1+v^2 & -uv & -u \end{matrix} \right] \begin{pmatrix} U_1 \\ U_2 \\ U_3 \\ \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix}$$

where Z_c is the depth in camera coordinates, and (u, v) are the image coordinates. Stacking the image point velocities $(\dot{u}, \dot{v})^T$ into the P-dimensional vector $\dot{\mathbf{p}}$ gives

$$\dot{\mathbf{p}} = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 & \mathbf{v}_4 & \mathbf{v}_5 & \mathbf{v}_6 \end{bmatrix} \begin{pmatrix} U_1 \\ U_2 \\ U_3 \\ \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix}$$

The 6 P-dimensional vectors $\mathbf{v}_i$ form a basis for the 6D subspace of Euclidean motions in P space.

Model-based tracking by projection of the image point motion vector $\mathbf{p}$ onto this subspace gives great improvement when the model is good – converting a P degree of freedom tracking problem into a 6 degree of freedom one. However, initially the accuracy of the model, and hence the accuracy of the subspace of it's Euclidean motion, is poor. Therefore, it is not desirable to fully project to this subspace from the start. We would like to accumulate 3D information from all of our observations, and progressively apply stronger constraints as the quality of this information improves.

3.2 Probabilistic Framework for 2D Tracking

The weighted application of constraints to the 2D tracking is achieved using a second Kalman filter. This P-dimensional filter takes inputs $\mathbf{C}_l$ and $\mathbf{m}_l$ from the image measurements (section 2.2), and uses a prior covariance matrix, $\mathbf{C}_p$. Encoding the constraints in this full, $P \times P$ prior covariance matrix enables the weighted application of a rich set of model-based constraints.

A Euclidean motion constraint can be included by using a covariance matrix of the form $\mathbf{C}_p = \sum_i \lambda_i^2 \mathbf{v}_i \mathbf{v}_i^T$. This comes from writing $\mathbf{p}$ as a weighted sum of Euclidean motions, $\mathbf{p} = \sum_i \lambda_i \mathbf{v}_i$. Then, if λ_i are assumed to be independent,

$$\mathbf{C}_\mathbf{p} = E(\mathbf{p}\mathbf{p}^T) = \sum_i \lambda_i^2 \mathbf{v}_i \mathbf{v}_i^T$$

Qualitatively, this says that the variance of the image motion is large in the directions corresponding to Euclidean motion, and 0 in all other directions. The weights λ_i can be used to vary the strength of the constraints – increasing λ_i increases the weight of the prior with respect to the likelihood.

Combining tracking and model building is intended to provide a smooth transition between loosely constrained, P dof tracking, and highly constrained, 6 dof model-based tracking. To do this we permit errors due to incorrect estimation of depth, weighted by the uncertainty in the depth of the 3D point. From the image motion equations (section 3.1), only the component of image motion due to camera translation depends on depth. Therefore, for a 1 standard deviation error in the inverse depth, $\sigma_{\frac{1}{Z_c}}$, the image motions are

$$\begin{pmatrix} \dot{u} \\ \dot{v} \end{pmatrix} = \sigma_{\frac{1}{Z_c}} \begin{bmatrix} -1 & 0 & u \\ 0 & -1 & v \end{bmatrix} \begin{pmatrix} U_1 \\ U_2 \\ U_3 \end{pmatrix}$$

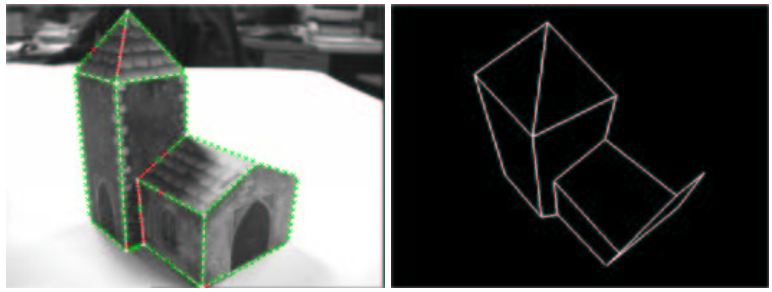


Figure 5: Church – real time tracking and 3D reconstruction

Stacking the image point velocities $(\dot{u}, \dot{v})^T$ into the P-dimensional vector $\dot{\mathbf{p}}$ gives

$$\dot{\mathbf{p}} = \sum_j \begin{bmatrix} \mathbf{e}_{1j} & \mathbf{e}_{2j} & \mathbf{e}_{3j} \end{bmatrix} \begin{pmatrix} U_1 \\ U_2 \\ U_3 \end{pmatrix}$$

where $\mathbf{e}_{ij}$, $i = 1, 2, 3$ are the P-dimensional vectors for motion due to making a 1 s.d. error in the estimate of the inverse depth of point j. Writing $\mathbf{p}$ as a weighted sum of these vectors, $\mathbf{p} = \sum_i \sum_j \mu_i \mathbf{e}_{ij}$, and therefore $\mathbf{C}_p = \sum_i \sum_j \sum_k \mu_i^2 \mathbf{e}_{ij} \mathbf{e}_{ik}^T$. Neglecting terms due to coupling between points gives $\mathbf{C}_p = \sum_i \sum_j \mu_i^2 \mathbf{e}_{ij} \mathbf{e}_{ij}^T$, which has the desired effect of allowing variance proportional to $\sigma^2_{\frac{1}{Z_c}j}$ in the $\mathbf{e}_{ij}$ directions. The depth variance for each point can be computed from its 3D pdf by $\sigma_{Z_c} = \mathbf{u}^t \mathbf{C} \mathbf{u}$, where $\mathbf{u}$ is a unit vector along the optical axis and $\mathbf{C}$ is the 3D covariance matrix. Then, assuming that σ_{Z_c} is small compared to Z_c, $\sigma_{\frac{1}{Z_c}} \approx \frac{\sigma_{Z_c}}{Z_c^2}$

Hence, the final form of the prior covariance matrix is

$$\mathbf{C}_p = \sum_{i=1}^{6} \lambda_i^2 \mathbf{v}_i \mathbf{v}_i^T + \sum_{i=1}^{3} \sum_{j=1}^{P/2} \mu_i^2 \mathbf{e}_{ij} \mathbf{e}_{ij}^T$$

This allows image motion due to Euclidean motion of the vertices in 3D, and also due to errors in the depth estimation of these vertices.

4 Results

4.1 Reconstructed models

Figures 5 and 6 give examples of models acquired using the system. The left image shows a frame from the tracking sequence, and the right image shows the reconstructed model from a novel view. The 'Church' model of figure 5 was generated in about 3 minutes from 10 observations along single camera trajectory. The 'ME' block in figure 6 was built in 2 steps. Firstly, the 'M' face was tracked for a low level camera motion. The 3D position information accumulated allowed this face to be robustly tracked as the camera was moved to view the 'E' face. In the second step, the 'E' face was added. The second camera motion used loosely constrained tracking of the new 'E' face, and tightly

constrained tracking of the ‘M’ face. New 3D information for the ‘E’ face was generated, whilst existing 3D information for the ‘M’ face was refined. This process took about 6 minutes.

Initial stability of the tracking proved to be a critical factor. For some objects, the ‘bootstrapping’ process could not get a foothold due to the inability to track small motions correctly e.g. if the edges had insufficient contrast. Another cause of error were the unwanted edges due to object texture and shadows. These caused tracking errors, which led to the accumulation of incorrect 3D information. Robustness could be improved in future by adding other user defined constraints into the 2D tracking framework, as we will describe in section 5.

4.2 Accuracy of models

The ‘ME’ block of figure 6 was reconstructed using about 20 observations as described above. To assess the accuracy of the model, the rms error in the angles and ratios of lengths for the body of the block were computed:

angles:	rms error = 2.30 degrees
ratios of length:	rms error = 2.73 %

The coplanarity of the points in the ‘M’ and ‘E’ planes was also determined:

‘M’ plane:	rms error = 0.85mm
‘E’ plane:	rms error = 0.74mm

4.3 Convergence of models

Figures 7 and 8 illustrate propagation of 3D pdfs and evolution of model structure. The ellipses in figure 7 are contours of constant probability density, at 100 standard deviations from the mean. Figure 8 is side on compared to the views of figure 7. It shows the structure of the model emerging from the initial planar hypothesis. Note that the pdfs shrink as the model structure improves – causing stronger Euclidean motion constraints in the tracking system.

To assess the rate of convergence, 40 observations were made as the camera was moved at 5mm intervals over a 200mm baseline, approximately 600mm from the block in figure 7. The rms error of the 3D position of the points was computed, and is plotted against iteration number in figure 9.

5 Further Work

The 2D tracking framework described in section 3.2 is extensible to the application of other model-based constraints. This can be achieved by incorporating additional full covariance matrices into the Kalman filter e.g. $\mathbf{C}^{-1} = \mathbf{C}_l^{-1} + \mathbf{C}_p^{-1} + \mathbf{C}_{other}^{-1}$. For example, a plane to plane transformation has 8 dof. Hence, $n\,(> 4)$ points in the image give $2n - 8$ constraints, which can be included as a rank $2n - 8$ matrix $\mathbf{C}_{other}^{-1}$ to provide a planarity constraint.

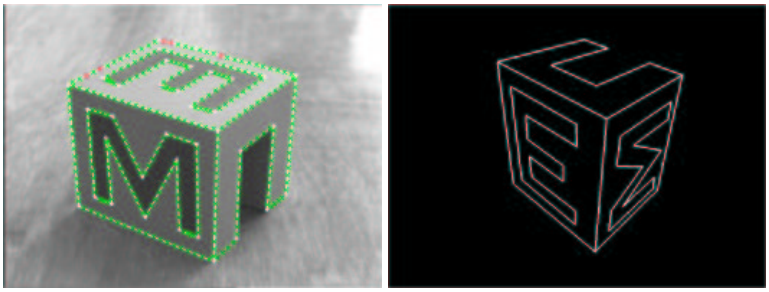
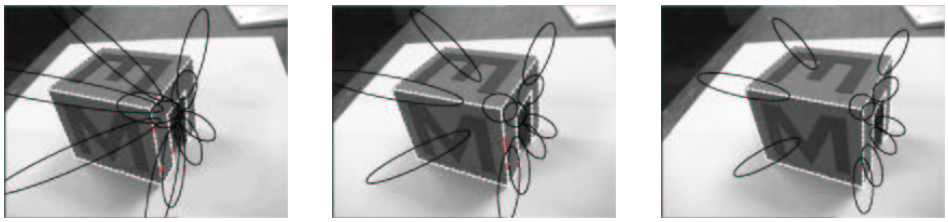


Figure 6: ME block – constructed in 2 stages exploiting weighted model-based tracking constraints



(a) Iteration 2 (b) Iteration 5 (c) Iteration 20

Figure 7: Propagation of 3D pdfs – the ellipses are 100 standard deviations from the mean vertex positions

(a) Iteration 2 (b) Iteration 5 (c) Iteration 20

Figure 8: Evolution of the model from initial planar hypothesis

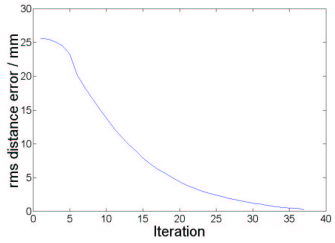


Figure 9: Convergence of the model

6 Conclusions

This paper has presented a semi-automatic wireframe acquisition system using real-time (25Hz) tracking. To do this, a 2D tracking framework enabling the natural integration of model-based constraints into the Kalman filter formulation was developed. The system has been used to acquire 3D models of simple edge based objects in around 5 minutes.

References

[1] A. Fitzgibbon, D. Eggert, and R. Fisher. High-level CAD Model Acquisition from Range Images. *Computer-Aided Design*, 29(4):321–330, 1997.

[2] N. Werghi, R. Fisher, A. Ashbrook, and C. Robertson. Object Reconstruction by Incorporating Geometric Constraints in Reverse Engineering. *Computer-Aided Design*, 31(6):363–399, 1999.

[3] O. Faugeras. *Three-Dimensional Computer Vision – A Geometric Viewpoint*. MIT press, 1993.

[4] P. Beardsley, P. Torr, and A. Zisserman. 3D model acquisition from extended image sequences. In *Proceedings of 4th European Conference on Computer Vision (ECCV'96)*, volume II, pages 683–695, Cambridge, April 1996. Springer-Verlag.

[5] R.I. Hartley. Lines and Points in Three Views and the Trifocal Tensor. *International Journal of Computer Vision*, 22(2):125–140, 1996.

[6] A. Blake and M. Isard. *Active Contours*, chapter 5, pages 97–113. Springer-Verlag, 1998.

[7] J. Crowley, P. Stelmaszyk, T. Skordas, and P. Puget. Measurement and Integration of 3D Structures by Tracking Edge Lines. *International Journal of Computer Vision*, 8(1):29–52, 1992.

[8] L. Matthies, T. Kanade, and R. Szeliski. Kalman Filter-based Algorithms for Estimating Depth from Image Sequences. *International Journal of Computer Vision*, 3:209–236, 1989.

[9] J. Porrill. Optimal Combination and Constraints for Geometrical Sensor Data. *International Journal of Robotics Research*, 7(6):66–77, 1988.

[10] C. Harris. Geometry from Visual Motion. In A. Blake, editor, *Active Vision*, chapter 16, pages 263–284. MIT press, 1992.

[11] T. Drummond and R. Cipolla. Real-time Tracking of Complex Structures with On-line Camera Calibration. In *Proceedings of British Machine Vision Conference (BMVC'99)*, pages 574–583, Nottingham, 1999.

[12] D. Lowe. Robust Model-based Motion Tracking Through the Integration of Search and Estimation. *International Journal of Computer Vision*, 8(2):113–122, 1992.