Robust Model-based Motion Tracking Through the Integration of Search and Estimation

David G. Lowe

Technical Report 92–11
May 1992

Computer Science Department
University of British Columbia
Vancouver, B.C., V6T 1Z4, Canada

Email: lowe@cs.ubc.ca

This paper has been accepted for publication in the *International Journal of Computer Vision*, August 1992 issue.

Robust Model-based Motion Tracking through the Integration of Search and Estimation

David G. Lowe

Computer Science Department
University of British Columbia
Vancouver, B.C., V6T 1Z4, Canada

ABSTRACT

A computer vision system has been developed for real-time motion tracking of 3-D objects, including those with variable internal parameters. This system provides for the integrated treatment of matching and measurement errors that arise during motion tracking. These two sources of error have very different distributions and are best handled by separate computational mechanisms. These errors can be treated in an integrated way by using the computation of variance in predicted feature measurements to determine the probability of correctness for each potential matching feature. In return, a best-first search procedure uses these probabilities to find consistent sets of matches, which eliminates the need to treat outliers during the analysis of measurement errors. The most reliable initial matches are used to reduce the parameter variance on further iterations, minimizing the amount of search required for matching more ambiguous features. These methods allow for much larger frame-to-frame motions than most previous approaches. The resulting system can robustly track models with many degrees of freedom while running on relatively inexpensive hardware. These same techniques can be used to speed verification during model-based recognition.

Introduction

With recent improvements in model-based vision algorithms and computer hardware performance, it will soon be possible to build low-cost, high-reliability systems for model-based motion tracking. Such systems can be expected to open up a wide range of applications in robotics by providing machines with real-time information about their environment. This paper describes a number of techniques for efficiently matching parameterized 3-D models to image features. The matching methods are robust with respect to missing and ambiguous features as well as measurement errors. The initial application is in a system for real-time motion tracking of articulated 3-D objects. With the future addition of an indexing component, these same techniques can be used as a component of general model-based recognition as well as motion tracking.

There are two types of errors that must be accounted for during the recognition process: matching errors and measurement errors. Each type of error has very different characteristics and is best handled by separate computational mechanisms. In the past, most model-based vision systems have been designed to minimize the influence of one of these classes of error, but there has been little work on methods for simultaneously accounting for both. This paper describes some methods for the integrated treatment of matching and measurement errors. In particular, allowance for matching errors improves the estimation for unknown model parameters by removing outliers, while accurate computation of variance in measurements can be used to limit the amount of search during matching.

Matching errors occur due to the mislabeling of image features that allows incorrect image features to be brought into correspondence with model features. As correct and incorrect matches are typically independent features of the scene, the location of an incorrect match does not provide any useful information regarding the location of the correct match. The standard method for handling matching errors in model-based vision is to perform a search, in which different combinations of potential matches are individually evaluated for consistency (Brooks 1981; Grimson & Lozano-Pérez 1987). The drawback of this approach is its computational cost, which grows exponentially as larger subsets of features are considered. However, this cost can be minimized through the probabilistic selection of the matches that are most likely to be correct. As reliable verification of an overdetermined interpretation allows the search to terminate when a correct set of matches is found, the average search time is minimized by performing the search in decreasing order of probability of correctness.

Measurements of the locations of correctly matched features have a very different distribution of errors. These errors are most easily modeled as having a Gaussian distribution, which can be represented with a mean and variance. The individual feature errors can be used to compute the variances and covariances for all model parameters. The residual of the data fitting can be used to evaluate the consistency of matches. The optimal estimation of model parameters from initial matches provides information for the probabilistic evaluation of the correctness of later matches, thereby minimizing matching errors as well as measurement errors.

Previous approaches

Most previous work on model-based motion tracking has assumed that velocity or acceleration is slow relative to the frequency of image acquisition, allowing each feature to be tracked according to its spatiotemporal continuity. When the location of features in each new frame can be accurately predicted from previous frames, there is little or no ambiguity in matching. By using the averaging properties of overdetermined systems, it is possible to tolerate occasional incorrect matches as long as the errors are limited in size by a small search window, so such systems can achieve reliable performance for frame-to-frame motion of up to several pixels. One of the earliest systems for 3-D model-based motion tracking was reported by Gennery (1982), which tracked Sobel edges within a 5-pixel range of predicted edges. The prediction included velocity extrapolation and filtering. In separate work, Gennery (1981) also examined the probabilistic evaluation of feature matches to a model. Verghese et al. (1988; 1990) describe a system for real-time tracking of rigid 3-D objects, based on the assumption that features are spatiotemporally dense (i.e., move less then one pixel from frame to frame). Bray (1990) has developed a system that individually tracks each image edge over short distances and uses the motion of these individual edges to solve for combined object motion. Perhaps the most dramatic demonstration of the approach of using spatiotemporal continuity is the work of Dickmanns & Graefe (1988) on the use of Kalman filtering as a framework for the real-time control of vehicles and aircraft from moving image sequences. He has demonstrated the ability to drive a van on normal roads at speeds up to 100 km/hour by tracking the road boundaries with sets of correlation-type feature detectors. Another example of the application of Kalman filtering to motion tracking is described by Wu et al. (1989).

The system described in this paper incorporates a search process to allow for the possibility of errors in feature matching, in addition to using detailed propagation of error bounds in feature measurements. The iterative matching procedure allows the most reliable matches to improve the probability of correctly matching other features. These methods allow the range of motion from frame to frame to be greatly increased without loss of reliability and with only modest increases in computation. As such, it draws on work in model-based recognition (Lowe 1985, 1987), which can be seen as the limiting condition when there are no bounds on motion from frame to frame. The major difference is that tracking begins its search from a predicted location while recognition requires a more powerful indexing method to generate matching hypotheses from image features in any location. Previous work on matching for recognition has placed much less emphasis on the propagation of parameter variance estimates during model verification. Each task can benefit from both matching techniques, so there will no doubt be an eventual merging of systems for recognition and tracking. Thompson & Mundy (1988) describe the use of motion prediction to constrain a different type of recognition algorithm, based on the clustering of vertex matches in an affine transform space. An approach to motion tracking using a Hough transform space around the current object position is described by Stephens (1990).

Modeling of measurement errors

The search process to be described later is used to eliminate incorrect matches (outliers) from the solution set. Therefore, it is reasonable to base the quantitative parameter solving on the assumption of normally distributed measurement errors. The parameters that must be solved for include the orientation and position of each object as well as the position of any articulated object components.

A number of previous motion tracking systems have used a Kalman filter to smooth these parameter estimates over time across a number of image frames. This is appropriate in applications such as aeronautics, where it is possible to put precise limits on the range of accelerations that can be expected. However, in typical robotics applications there are few useful limits on expected accelerations (e.g., when objects are bumped or collide), so that the smoothing performed by the Kalman filter would be either misleading or ineffective. For example, velocity smoothing would cause an object dropped on a table to "bounce" into the table before recovering. Given the overconstrained information usually available from each image frame, it is possible to replace the Kalman filter with a more efficient form of velocity prediction and stabilization with prior variances. The stabilization is important during early stages of matching when only a few features are a part of the solution, and it is useful for estimating the initial probability of correctness for each potential feature match.

To perform a least-squares fit to the data for the non-linear unknown parameters, we use the Gauss-Newton method augmented by stabilization with prior variances. Each iteration of the Gauss-Newton method solves the following matrix equation:

$$\begin{bmatrix} \mathbf{J} \\ \mathbf{W} \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{e} \\ \mathbf{0} \end{bmatrix}$$

where $\mathbf{x}$ is the unknown vector of corrections to be made to each parameter, $\mathbf{e}$ is the vector of errors between matched image features and model predictions, and $\mathbf{J}$ is the Jacobian matrix of errors with respect to parameters. $\mathbf{W}$ is a diagonal weighting matrix used to stabilize the solution, in which each weight is inversely proportional to the prior standard deviation, σ_i, expected for parameter i:

$$W_{ii} = \frac{1}{\sigma_i}$$

This system is minimized by solving the corresponding normal equations:

$$\begin{bmatrix} \mathbf{J^T} & \mathbf{W^T} \end{bmatrix} \begin{bmatrix} \mathbf{J} \\ \mathbf{W} \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{J^T} & \mathbf{W^T} \end{bmatrix} \begin{bmatrix} \mathbf{e} \\ \mathbf{0} \end{bmatrix}$$

Which multiplies out to

$$\left(\mathbf{J^T J} + \mathbf{W^T W}\right) \mathbf{x} = \mathbf{J^T e}. \tag{1}$$

Since $\mathbf{W}$ is a diagonal matrix, $\mathbf{W^T W}$ is also diagonal but with each element on the diagonal squared. This means that stabilization can be accomplished by first forming $\mathbf{J^T J}$ and then adding small numbers to the diagonal. This method almost always converges to the accuracy limits of the data within one or two iterations for the motion tracking problem, as the initial parameter estimates are quite good. Full details on the development of the above solution and stabilization methods have been given in an earlier paper (Lowe 1991).

If the number of error measurements derived from the data, m, is greater than the number of parameters, n, we can estimate the variance, σ^2, in the data from the size of the residual:

$$\sigma^2 = \frac{\|\mathbf{Jx} - \mathbf{e}\|^2}{m - n}$$

If σ is much greater than the standard deviation of the measurement errors in the data, then it is likely that the system contains at least one incorrect match so we abandon this branch of the search tree. Otherwise, this branch continues to be explored and new matches are attempted until no more can be found.

Following each iteration of data fitting, the covariance matrix, $\mathbf{P}$, for the model parameters is given by the inverse of the matrix on the left-hand side of equation 1:

$$\mathbf{P} = \left(\mathbf{J^T J} + \mathbf{W^T W}\right)^{-1}$$

This can be computed efficiently as a by-product of the least-squares solution. Then the variance in each future predicted measurement can be computed from this covariance matrix:

$$\mathbf{S} = \mathbf{APA^T} \tag{2}$$

where each row of $\mathbf{A}$ gives the derivatives of a predicted measurement with respect to each of the model parameters. For matching model lines, we are interested in the variance perpendicular and parallel to each endpoint of each visible model edge as well as the variance in orientation of each model edge. The variance of each predicted measurement is given by the corresponding diagonal element of $\mathbf{S}$. Note that it is not necessary to compute the off-diagonal elements of $\mathbf{S}$, which otherwise would be a large and expensive matrix to compute.

Therefore, we have completed the circle, in which new matches constrain model parameters, which in turn constrain the predictions for future matches. A few correct initial matches can greatly reduce the variance of further predictions and often eliminate further search, as shown in the final examples.

Matching with minimal search

Robust matching can be achieved by searching for sets of matching image segments that are consistent with a projection of the object using a single set of parameter values. As there are usually many more matches than are needed to solve for the model parameters, the final solution is overconstrained and it is unlikely that a false set of matches will closely fit the model. However, the search process itself is computationally expensive, as it is necessary to compute updated model parameters to check each combination of matching segments. This search process is minimized

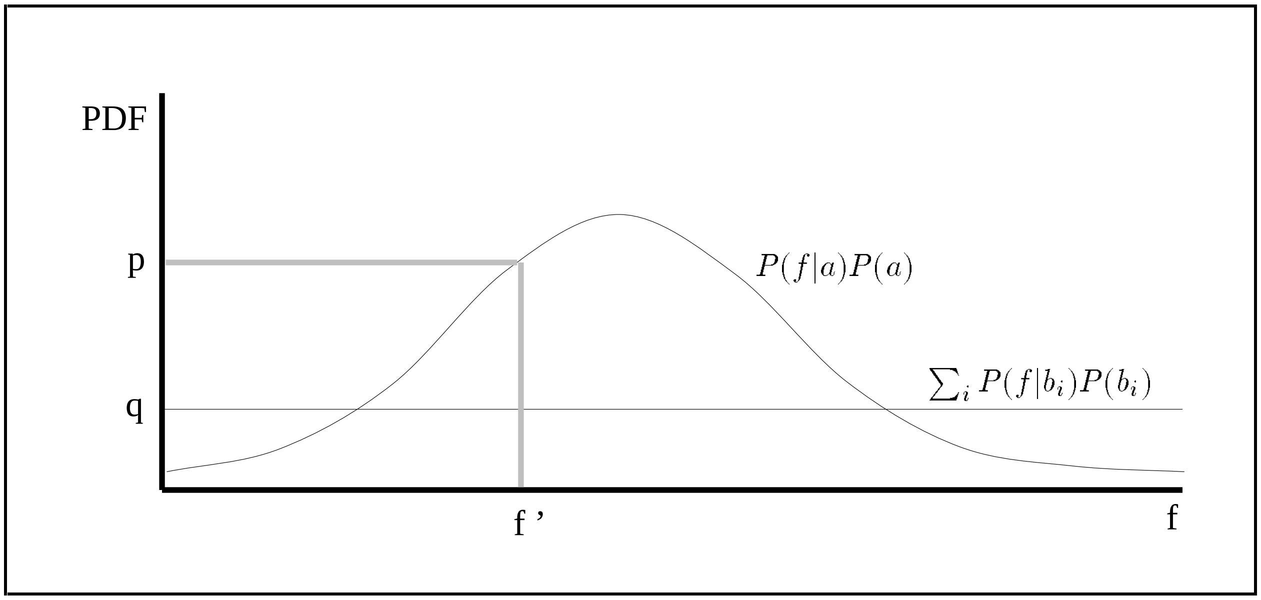

Figure 1: The probability distribution is illustrated for a measurement, f, of a predicted model feature, a. This is compared to the uniform background distribution of other features, b_i, which could give rise to false matches.

by using a best-first search, starting with those matches that are most likely to be correct and using those to constrain the expected locations of other matches. In this way, the most reliable matches in each image are used to increase the probabilities of correctly matching more ambiguous features, which in many cases can eliminate backtracking altogether. Because the final identification is overdetermined and can be reliably verified, the search can terminate once a valid set of matches have been found.

Therefore, an important aspect of minimizing search during matching is to accurately estimate the probability that each potentially matching image feature matches some corresponding model feature. These probabilities can be determined using Bayesian decision theory (Duda & Hart 1973) as a function of a vector **f** of feature measurements relating each pair of model and image features. Let a represent an image feature that arose from the projection of a corresponding model feature, and let $b_i, 1 \leq i \leq n$, represent all other (incorrectly matching) image features. In the absence of other information, the incorrectly matching image features are modeled as arising from a uniform background distribution. Then we can use Bayes rule to compute the probability that a particular feature measurement vector **f** arose from a model feature rather than the set of background features:

$$P(a|\mathbf{f}) = \frac{P(\mathbf{f}|a)P(a)}{P(\mathbf{f})}$$

$$= \frac{P(\mathbf{f}|a)P(a)}{P(\mathbf{f}|a)P(a) + \sum_i P(\mathbf{f}|b_i)P(b_i)}$$

This probability calculation is illustrated in Figure 1 for the case of a one-dimensional feature measurement, f. We assume a Gaussian probability distribution for the model feature and a uniform background density for the other features. For a particular feature measurement, f', the probability that the feature arose from the model is given by

$$P(a|f') = \frac{p}{p+q} \tag{3}$$

where p and q are the values of the probability distributions at f', as shown in Figure 1.

The particular feature measurements that are used for matching line segments are the perpendicular distance of the center of the image segment from the projected model segment and the angular difference in orientation of the segments. Therefore, it is necessary to determine the probability distributions for model and background features as a function of this two-dimensional space of measurements.

Let x be the perpendicular distance of an image segment from its correctly matching model segment and y be the angular difference in orientation. For the sake of efficiency, we assume that these measurements are independent. Therefore, the two-dimensional probability distribution for these variables is

$$p(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-0.5[(x/\sigma_x)^2 + (y/\sigma_y)^2]}$$

where σ_x and σ_y are obtained from the square roots of the corresponding diagonal elements of the matrix $\mathbf{S}$ in equation 2.

We assume that the background distribution of other (incorrectly matching) image segments is uniform with respect to location and orientation. We can calculate the density of this uniform distribution by dividing the total number of segments in the image by the area that the features can occupy in the feature measurement space. It would be possible to use a local measure of feature density around each potential match, although this is not done in the current implementation. So far, we have considered only the perpendicular distance of an image segment from a projected model segment. However, it is also necessary that the image segment overlap the model segment in the direction parallel to its length. As the image segment could be partially detected for any interval along its length, this is better modeled as a uniform interval probability distribution rather than a Gaussian distribution, which is why it is not included in the multivariate normal distribution above. The predicted model segment is extended in length to include the uncertainty in its endpoint positions, giving a total length m. An image segment of length s can have its midpoint fall anywhere within an interval of length $m - s$. If the image contains N line segments, then the background summed probability density function of line segments over all orientations and midpoint positions meeting the overlap constraint is the uniform value (independent of x and y)

$$q = \frac{N(m-s)}{\pi w h}$$

where w and h are the width and height of the image, or the area over which N was determined. The factor π in the denominator arises because this is the range of orientations for nondirected line segments.

The probability of correctness of each match depends on the relative sizes of p and q, as in equation 3. Therefore, matching segments are evaluated more highly if they have a small angular difference and perpendicular distance from the model prediction, if their lengths closely agree with the prediction, and if the predicted variance of each measurement is small. These criteria seem to capture the most relevant properties useful for correctly matching line segments.

Best-first search

The probability of correctness for each potential match between the model predictions and image features is used to guide a best-first search process. Each node of this search tree requires performing a least-squares solution for all model parameters, which is a relatively expensive operation. Therefore, it is important to select an optimal ordering for the search to minimize the possibility of backtracking, and to also select enough matches at each node to constrain the parameters in lower nodes and lead to a quick acceptance or rejection.

The top-level nodes of the search tree make use of enough matches to constrain at least the number of degrees of freedom in the current model. For the example shown in this paper, the model has 7 degrees of freedom so at least 4 line segment matches are selected (each line segment match constrains 2 degrees of freedom). This number of matches are selected from a ranked list of the best matches, and other matches are added with decreasing probability as long as the product of probabilities remains above 0.9. The stabilized least-squares solution is carried out, and the residual is checked as described earlier. If this match is rejected, then at least one of the segments in the match set must be in error. The probability of correctness for each segment match is reduced by $1/n$ in the rejected set of n matches, and the best-first search proceeds to form new search sets based on these updated probabilities. In general, the reduction in probabilities for the previous matches will cause other matches to be considered, but a further check is performed to see that no new set contains a complete rejected set from some previous node of the search tree.

In practice, backtracking is usually avoided by this conservative approach of selecting only a few of the most reliable matches and using these to constrain the locations of further matches. However, there will always be some probability of making mistaken matches that lead the parameter solution away from its true value, so the ability to backtrack adds substantially to the system's robustness.

A further method that is used to allow for sudden unexpected motion of the object is to increase the search range when the system is unable to find the object. The search range is determined by the prior variances attached to each object parameter before processing each new image. If the ranked list of potential matches for this image contains too few candidates, then the parameter ranges are doubled and a search is made for new matches. This doubling can be performed up to 2 times, leading to a large search region when necessary in order to find an object that cannot be found locally.

Real-time line detection

Until recently, the major computational bottleneck for motion tracking has been the large number of computations required for low-level image analysis. However, a number of vendors now offer inexpensive systems for performing a range of digital image-level operations at video rates. In this work, we have used a Datacube MaxVideo 20 board that performs 8x8 convolutions on 512x512 images at up to 60 frames per second (performing up to 1 billion 8-bit multiplications per second). We use an 8x8 convolution kernel that is a Laplacian of Gaussian operator with $\sigma = 1.2$, for use in Marr-Hildreth (1980) edge detection.

Currently, the output of the image processing operations must be transferred over a bus interface to the host computer (a Sun SPARCstation 2) for higher-level processing. This simple image transfer and edge linking are currently the major computational bottlenecks, but they are minimized by only transferring a region around the expected location of the object as computed from the previous image. This means that processing is somewhat slower for large, nearby objects as compared to smaller, more distant ones.

The convolved image region is scanned to detect zero-crossing locations where there is a change of sign between adjacent pixels. The approximate gradient at the zero crossing is computed by taking the difference of neighboring pixels across the zero crossing. These edge pixels are linked into lists on the basis of local 8-neighbor connectivity. At the same time, Canny (1986) hysteresis thresholding is performed using a high and low threshold on gradient. The resulting lists of connected edge points are segmented into straight line segments using a scale-invariant recursive subdivision algorithm described in an earlier paper (Lowe 1987). All of these operations can be performed very efficiently even on a serial machine; however, the reliability and accuracy of this feature detection could be improved with the availability of more computing resources.

In order to make subsequent feature matching as efficient as possible, all of the line segments are indexed into a three-dimensional array on the basis of 2-D position of the midpoint and orientation. Subsequent attempts to match features at a particular range of positions and orientations need to examine only those segments that are indexed in the small subset of the array locations that intersect these bounds.

Implementation results

All steps of edge detection, matching, convergence and verification can be done in under 0.3 seconds on the system described above. In most cases, the probabilistic matching criteria select correct matches on the first attempt and do not require any further search. In difficult cases, search is terminated after exploring 5 branches of the search tree so that the next image in the sequence can be processed without significant delay.

We have tested this system on thousands of images by holding the object model in front of the camera and slowly moving it. The system displays the edges of the object at the current calculated location superimposed on the camera image. These edges are shown in yellow when the object has been correctly verified and in red when the object cannot be matched. As the current computational resources limit processing to 3 to 5 frames per second and a limited search range, it is necessary to move the object quite slowly. However, the tracking is quite robust and continues even when up to half or more of the edges are occluded. The system performs well over a wide range of lighting conditions and with complex backgrounds containing many false edges.

Figures 2–7 show an example of this process for one image of a motion sequence. Figure 2 shows the input image from a CCD camera. The object is a file box with a hinged lid, so that there are 7 unknown parameters that must be solved for. Parts of the box are occluded by the author's hands, there are many reflections from the object's surface, and the background is cluttered. The line segments extracted from this image and the initial estimate for the position of the object computed by velocity extrapolation from the previous two images are shown in figure 3. This example is for an object that is relatively far from its predicted position. Also shown as heavy lines are the best matches to image line segments which are selected on this iteration. In the background, the light gray shading indicates the union of all regions within 2 standard deviations of the predicted model edges (there is no display of the variance in edge orientation, which is also computed). Subsequent iterations are shown in figures 4 to 6. The rapid reduction in the size of the shaded regions indicates how the reduction in variance resulting from earlier matches greatly reduces the subsequent search space. This "locking on" phenomenon is the result of the overconstrained nature of the model-based vision problem and is what leads to high reliability and efficiency. The shaded gray regions also illustrate that the variance is far from uniform for different parts of the object, meaning that simpler strategies for reducing the search range are unlikely to work as well. As can be seen from figure 7, the final computed parameters are quite accurate due to the overconstrained data and the least-squares fit. All steps of matching in these figures requires about 0.1 seconds on a Sun SPARCstation 2 (not including line segment extraction).

Conclusions and future directions

This paper describes an approach to model-based matching that provides for both reliability and efficiency by integrating the treatment of matching and measurement errors. There is a role for both general tree search and for error estimation.

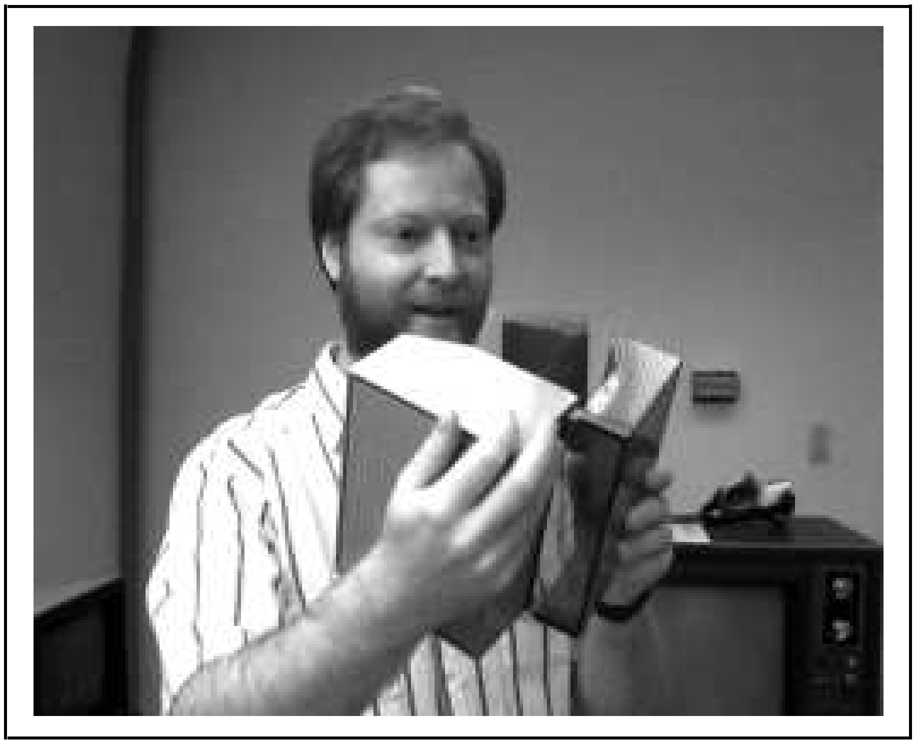

Figure 2: The original image from a motion sequence of a file box with a hinged lid.

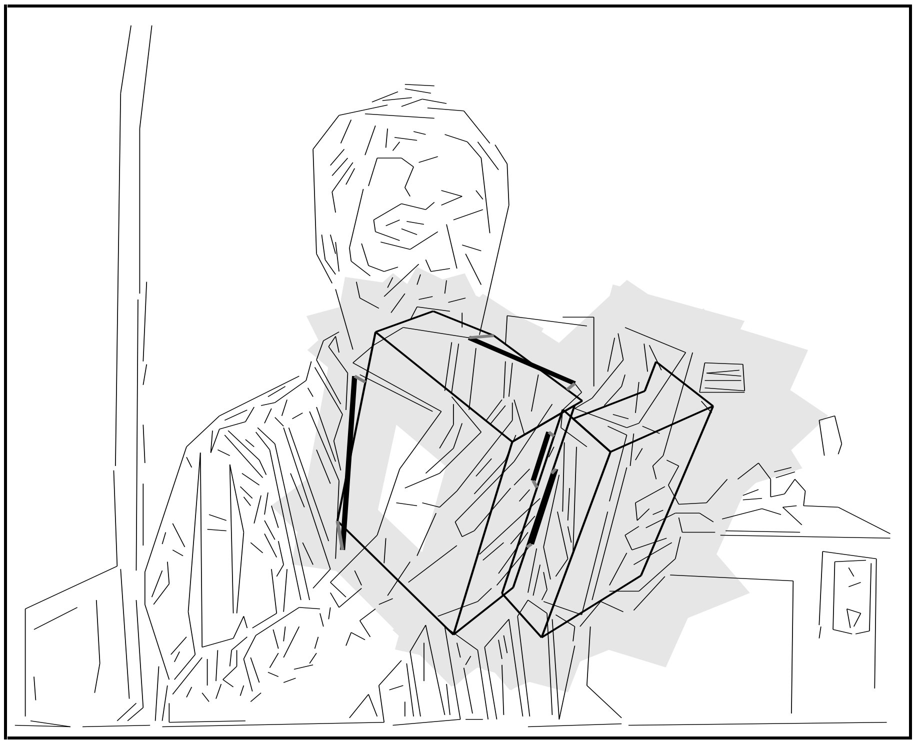

Figure 3: Line segments extracted from the image are shown with the model superimposed from its initial estimated viewpoint. The shaded area shows the union of the regions of uncertainty for feature locations. Initial matched image segments are shown with heavy lines.

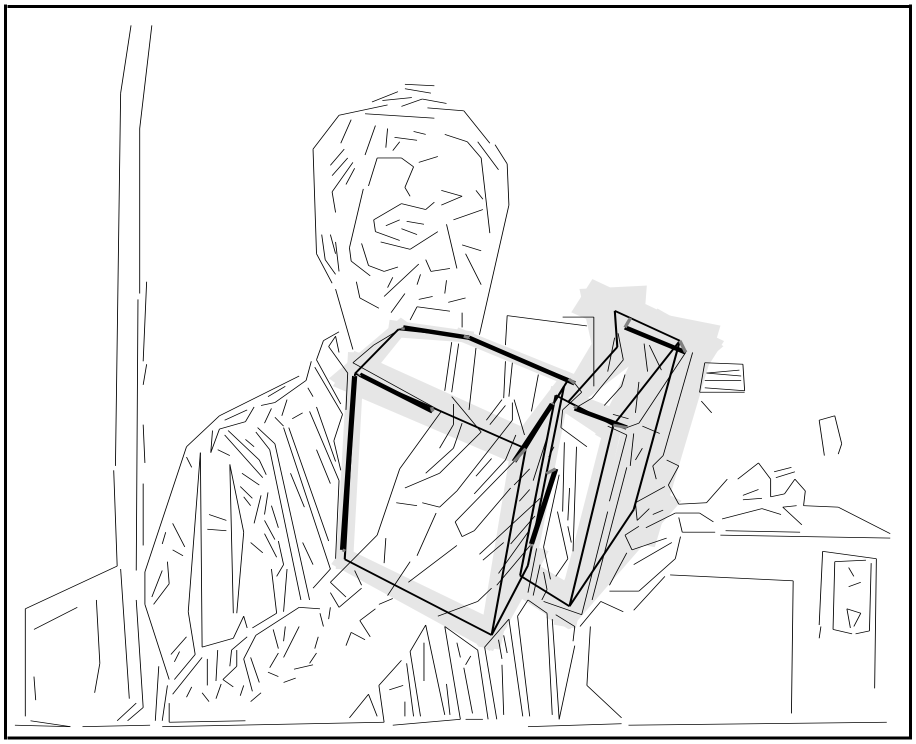

Figure 4: The position of the model is shown following the first iteration of matching and parameter determination. Further matched image segments are shown with heavy lines.

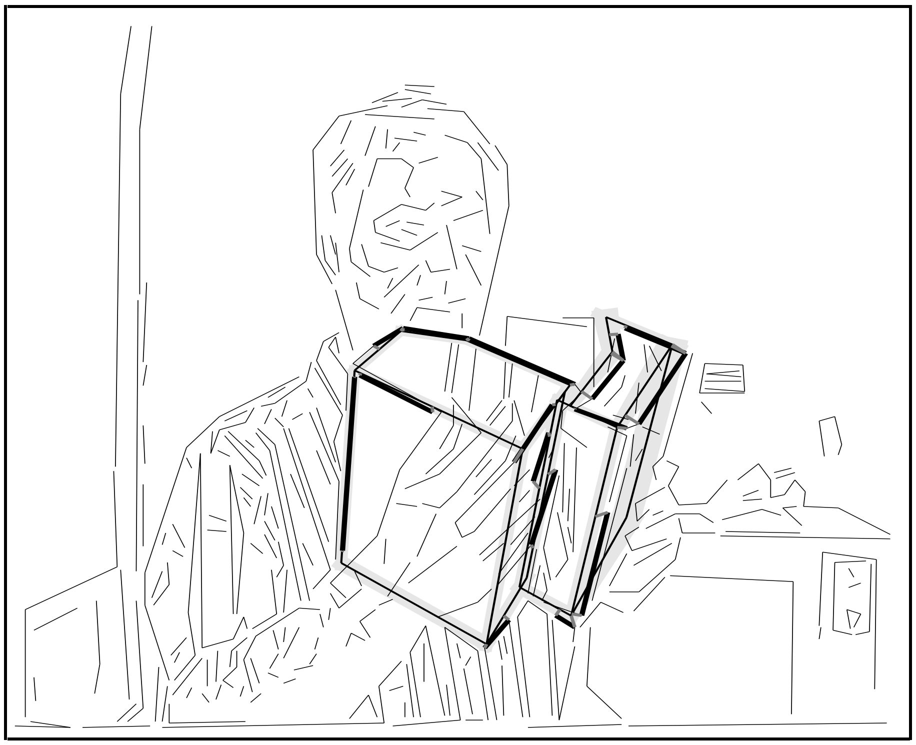

Figure 5: The model following the second iteration of parameter determination.

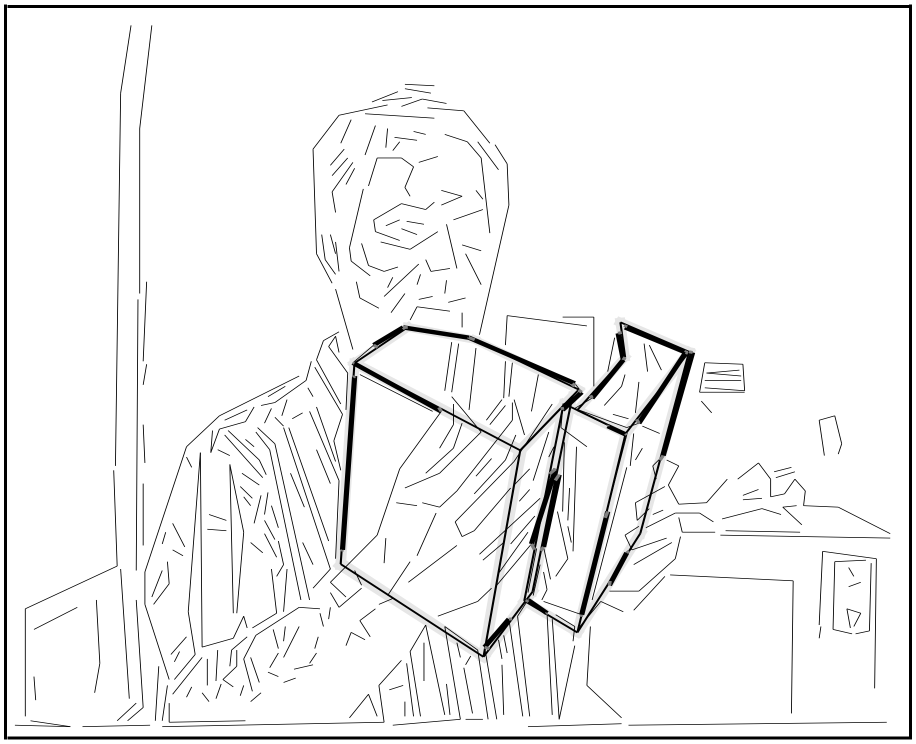

Figure 6: The model is shown following the third and final iteration of parameter determination. The region of uncertainty around each model edge is now very small.

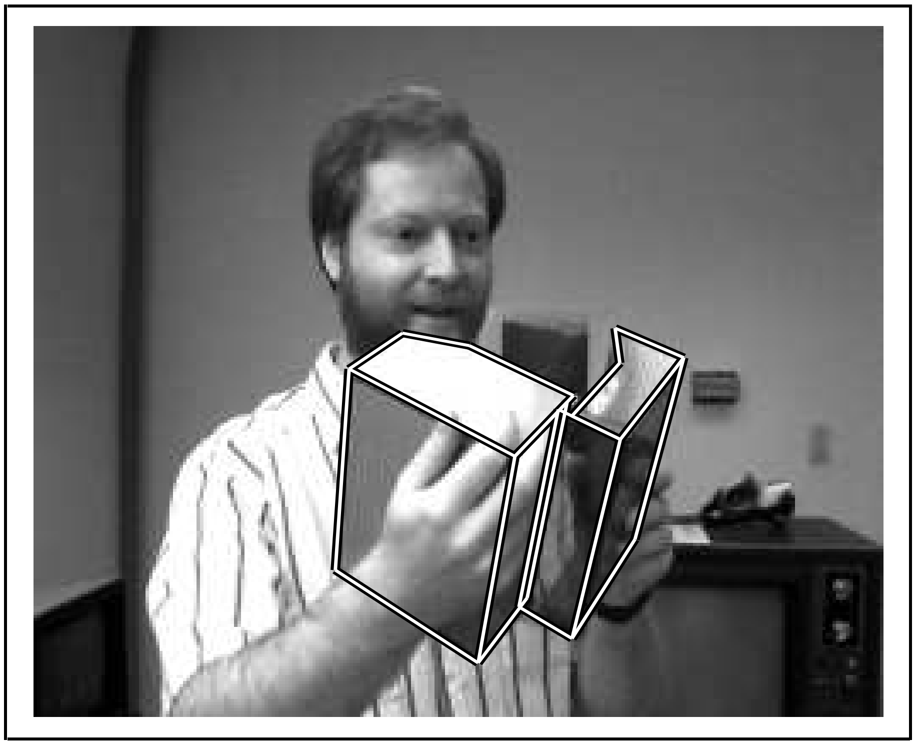

Figure 7: The model is superimposed on the original image from its final viewpoint.

This approach has been implemented in a functioning system that can robustly track objects at the rate of 3 to 5 frames per second. With moderate increases in computer speeds—as are already available with low-cost parallel systems of microprocessors—such a system could be used to track objects at 30 or 60 frames per second and provide real-time visual input for robots. Tracking multiple objects would require at most a linear increase in computer speeds. With yet faster processing, it would be possible to track flexible objects with large numbers of internal parameters.

An important future direction for this work is to incorporate the capability for general object recognition (Lowe 1987). Recognition would make use of all of the components described in this paper, but would need in addition an indexing system from image feature groupings to potential object matches. The addition of feature grouping techniques would also be very useful for the motion tracking problem, as higher-level groupings are far less likely to be incorrectly matched than isolated line segments. The result would be the integration of recognition and tracking, which are simply different ends of a continuum representing the degree of prior knowledge regarding the locations of objects in an image.

References

Bray, A.J. 1990. "Tracking objects using image disparities," *Image and Vision Computing,* 8(1): 4–9.

Brooks, R.A. 1981. "Symbolic reasoning among 3-D models and 2-D images," *Artificial Intelligence,* 17: 285-348.

Canny, J. 1986. "A computational approach to edge detection," *IEEE Trans. on Pattern Analysis and Machine Intelligence,* 8(6): 679–698.

Dickmanns, E., and Graefe, V. 1988. "Dynamic monocular machine vision," *Machine Vision and Applications,* 1: 223–240.

Duda, R.O. and Hart, P.E. 1973. *Pattern Classification and Scene Analysis.* Wiley: New York.

Gennery, D. 1982. "Tracking known three-dimensional objects," *Proc. of National Conference on Artificial Intelligence,* Pittsburgh, 13–17.

Gennery, D. 1981. "A feature-based scene matcher," *Proc. of Inter. Joint Conference on Artificial Intelligence,* Vancouver, Canada, 667–673.

Grimson, E., and Lozano-Pérez, T. 1987. "Localizing overlapping parts by searching the interpretation tree," *IEEE Trans. on Pattern Analysis and Machine Intelligence,* 9: 469–482.

Lowe, D.G. 1985. *Perceptual Organization and Visual Recognition.* Kluwer Academic Publishers: Boston, MA.

Lowe, D.G. 1987. "Three-dimensional object recognition from single two-dimensional images," *Artificial Intelligence,* 31(3): 355-395.

Lowe, D.G. 1991. "Fitting parameterized three-dimensional models to images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(5): 441–450.

Marr, D., and Hildreth, E. 1980. "Theory of edge detection," *Proc. Royal Society of London, B,* 207: 187-217.

Stephens, R.S. 1990. "Real-time 3D object tracking," *Image and Vision Computing,* 8(1): 91–96.

Thompson, D.W., and Mundy, J.L. 1988. "Model-based motion analysis: Motion from motion," *Robotics Research: The Fourth International Symposium,* R. Bolles and B. Roth, eds., MIT Press: Cambridge, Mass., 299–309.

Verghese, G. and Dyer, C.R. 1988. "Real-time model-based tracking of three-dimensional objects," Univ. of Wisconsin, Computer Sciences, Technical Report 806.

Verghese, G., Gale, K., and Dyer, C.R. 1990. "Real-time, parallel motion tracking of three-dimensional objects from spatiotemporal image sequences," in *Parallel Algorithms for Machine Intelligence and Vision,* Kumar *et al.,* Eds., Springer-Verlag: New York.

Wu, J.J., Rink, R.E., Caelli, T.M., and Gourishankar, V.G. 1989. "Recovery of the 3-D location and motion of a rigid object through camera image (an extended Kalman filter approach)," *International Journal of Computer Vision,* 2(4): 373–394.