Real-Time Systems for Tracking Articulated Three-Dimensional Objects

David G. Lowe

Computer Science Department
University of British Columbia
Vancouver, B.C. V6T 1W5, Canada

Abstract

This paper presents an overview of a number of techniques used for tracking the motion of articulated 3-D objects in real time. With recent advances in robust methods for model-based vision and improved performance of computer systems, 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. Unlike most previous work on model-based motion tracking, this system provides for the integrated treatment of matching and measurement errors during motion tracking. 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 also be used for general model-based recognition. The current real-time implementation is based on matching straight line segments, but some preliminary experiments on matching arbitrary curves are also described.

Introduction

Model-based computer vision makes use of prior knowledge of the shape and appearance of objects to identify and locate them in an image. This is a particularly appropriate approach for many applications of computer vision to robotics, as robotic tasks are likely to involve known objects and model-based matching can be performed in a robust and efficient manner. However, robotic interaction by its nature is a dynamic process, in which information about the motion of objects must be updated in real time. This paper describes the development of methods for tracking the location of moving objects in a robust and efficient manner that would be suitable for a wide range of tasks. These methods apply not only to rigid objects, but also to those with moving articulated components. Although we do not address the general recognition problem [5] here, these same techniques can be used to speed verification during model-based recognition. In many robotics tasks, the approximate positions of objects are known, so the methods described in this paper can be directly used to lock on to the object position, give its precise location, and update this information as the object moves.

One advantage of the model-based approach is that it tracks 3-D objects from the information extracted from a standard inexpensive video camera. In many situations, it is useful to have multiple cameras to provide more accurate information, reduce the likelihood of occlusion, and increase the overall robustness of the system. All of the techniques presented in this paper can be applied to features derived from multiple cameras in the same way as those derived from a single camera. The relative positions of the cameras can be prespecified or can be treated as additional parameters to be solved for.

There are two types of errors that must be accounted for during the model-based matching 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 (particularly measurement errors), but there has been little work on methods for simultaneously accounting for both. This paper describes a number of techniques for integrated treatment of matching and measurement errors. In particular, allowance for matching errors improves the measurement accuracy for unknown model parameters by eliminating outliers, while accurate computation of variance in measurements can be used to limit the amount of search.

Matching errors occur due to ambiguities in the detec-

tion of image features that allow 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 other areas of computer vision is to perform a search, in which different combinations of potential matches are individually evaluated for consistency. 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 use of a probabilistic selection of the matches that are most likely to be correct. As reliable verification 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 usually 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. As the integrated treatment of measurement errors can be performed simultaneously without the exponential costs of the search process, there is much to be gained by exploiting the measurements to reduce the amount of search.

Previous approaches

Most previous work on model-based motion tracking has assumed that the motion 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 overdetermined systems, it is even possible to compensate for occasional incorrect matches, so these 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 [3], which tracked Sobel edges within a 5-pixel range of predicted edges. The prediction included velocity extrapolation and filtering. Verghese, Gale & Dyer [12,13] 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 [1] 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 [2] 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.* [14]. Some early work on real-time motion tracking using marked objects is described by Pinkney [10], which has been used to develop a vision system that can be used to track and grasp free-flying satellites from the Space Station.

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. This allows 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 [4, 5], 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. However, each task can benefit from both matching techniques, so we view this paper as contributing to the eventual merging of systems for recognition and tracking. Thompson and Mundy [11] 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.

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 motion 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 an object is knocked or dropped onto a hard surface), so that the smoothing of the Kalman filter would be either misleading or ineffective. 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.

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

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

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

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

This system is minimized by solving the corresponding normal equations:

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

Which multiplies out to

$$(\mathbf{J}^T\mathbf{J} + \mathbf{W}^T\mathbf{W})\,\mathbf{x} = \mathbf{J}^T\mathbf{e} + \mathbf{W}^T\mathbf{W}\mathbf{d}. \qquad (1)$$

Since $\mathbf{W}$ is a diagonal matrix, $\mathbf{W}^T\mathbf{W}$ is also diagonal but with each element on the diagonal squared. This means that stabilization can be accomplished by first forming $\mathbf{J}^T\mathbf{J}$ and then adding small numbers to the diagonal. This method almost always converges 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 method have been given in an earlier paper [8]. In the motion tracking system described in this paper, the following additional techniques are used for verifying the solution and predicting the variance of further feature measurements.

If the number of data points, 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:

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

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:

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

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 the following covariance matrix:

$$\mathbf{S} = \mathbf{A}\mathbf{P}\mathbf{A}^T$$

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 almost eliminate further search, as shown in the final examples.

Matching with minimal search

The problem of matching model features to image features would result in an exponential search space if we took the brute force approach of considering all possible matching subsets. However, in model-based vision there is a reliable verification procedure that will identify correct sets of matches once they are found. This means that it is possible to have a far better average case performance by using probabilistic measures to select those matches that are most likely to be correct and considering them first during the search process.

The probabilistic evaluation of each match is based on the assumption that image features are drawn from two sets with different probability distributions: 1) correctly matching features with parameter measurements that are close to the predicted values, and 2) independent features that have a uniform prior distribution of parameter measurements. The posterior probability that a particular feature is from the set of correct matches can be computed by comparing the probability densities at the measured parameter values for each distribution. The search proceeds in a best-first ordering according to these probabilistic estimates until a valid match is confirmed.

When computing the probability that a feature match is correct, the simplest approach would be to assume that correct matches are normally distributed about the predicted measurement locations. However, the range of motions encountered in typical applications is likely to have a much broader distribution. The variance of predictions for a new image frame must be large to allow for unexpected accelerations of motion, yet the great majority of predictions will be much more accurate than this. Therefore, better success has been obtained by using a probabilistic matching criterion that, in essence, uses the closeness of each match as an estimate of the expected range of variation. The variances of the measurement predictions still play a very important role in initially limiting the size of the region in which this second criterion is evaluated (currently the size of this search region is 2 standard deviations in each parameter dimension).

Under the assumption that incorrect matches will be uniformly distributed in terms of position, orientation and scale, we can compute the expected number of incorrect

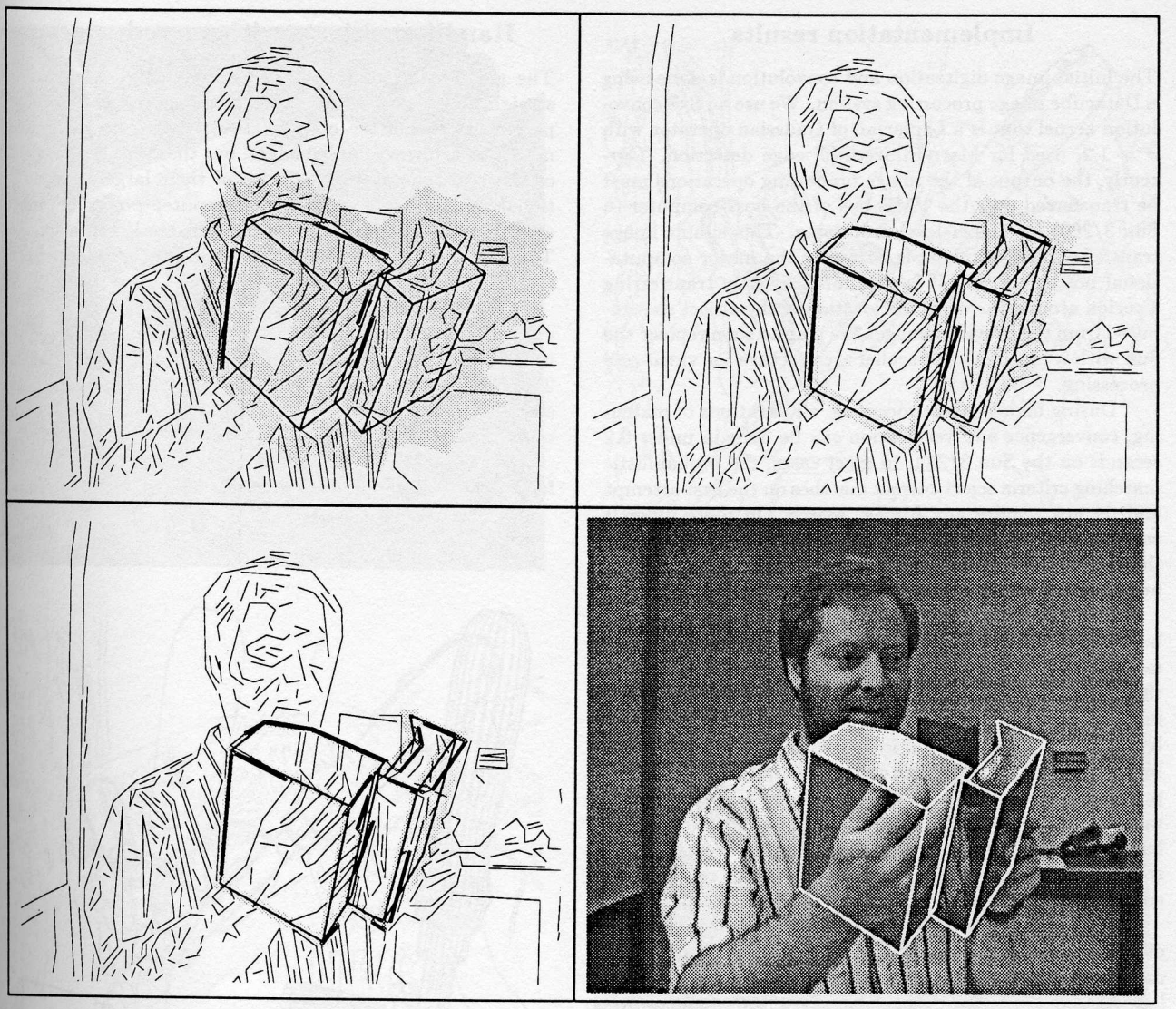

Figure 1(a–d): These images show a number of steps in the iterative matching of a model to a single frame of a motion sequence. Detected image edges are shown with thin lines, predicted model edges are thicker, and matched image edges are shown with the thickest lines. The shaded area in the background of each figure shows the union of all image locations within 2 standard deviations of predicted model edges, illustrating the rapid reduction in variance as the matching proceeds.

matches, E, that will be within the measured agreement in each parameter. This is the same problem as was addressed in an earlier paper [6] for the purpose of feature matching during model based recognition, resulting in the following expression:

$$E = \frac{4D\theta s(p-m)}{\pi m^2}$$

where θ is the orientation difference, p is the length of the predicted segment plus error bounds, m is the length of the matched segment, and s is the perpendicular distance from the center of the matched segment to the predicted segment. For small values of E, the expected number is approximately equal to the prior probability of occurrence of a false match within the given range of measurements.

These probability measures for each potential match are used to order the search by growing the tree of possible interpretations in a best-first ordering. The other limitation on this search is that the depth of the tree prior to parameter solving should be sufficient to constrain the number of free parameters (e.g., a minimum of 4 line matches is required for a 7 degree of freedom model). This results in a substantial immediate reduction in the variances of further predictions, thereby limiting the time that is devoted to checking this branch of the search tree.

Implementation results

The initial image digitzation and convolution is done using a Datacube image processing system. We use an 8x8 convolution kernel that is a Laplacian of Gaussian operator with $\sigma = 1.2$, used for Marr-Hildreth [9] edge detection. Currently, the output of the image processing operations must be transferred over the VME bus of the host computer (a Sun 3/260) for higher-level processing. This simple image transfer and edge linking is currently the major computational bottleneck, but it is minimized by only transferring a region around the expected location of the object as computed from the previous image. We plan to soon replace the Sun with a much faster parallel architecture for video-rate processing.

During higher-level processing, all iterations of matching, convergence and verification can be done in under 0.2 seconds on the Sun 3/260. In most cases, the probabilistic matching criteria select correct matches on the first attempt and do not require any further search. In more 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 outputs 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 1 to 2 frames per second and a limited search range, it is necessary to move the object quite slowly. As long as the object is moved slowly, the tracking is quite robust and continues even when up to half or more of the edges are occluded.

Figure 1 shows an example of this process for one image of a motion sequence. Figure 1(a) shows the input line segments extracted from an image of a file box with a hinged lid. Also shown in thicker lines is the initial estimate for the position of the object computed by velocity extrapolation from the previous image. The thickest lines are the best matches to image line segments which were selected on this iteration. In the background, the light grey 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 (b) and (c). 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 grey 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. The final position of the object is shown superimposed on the original image in figure 1(d).

Handling objects with curved edges

The real-time system described above makes use only of straight line segments extracted from the image. We have performed a number of experiments with modeling and matching arbitrary curved edges, but these are not yet part of the real-time system because of their larger computational requirements. As more computer power becomes available, for example with a parallel implementation on Transputers, we expect these methods to be adopted for real-time use.

There are many techniques for modeling objects with curved surfaces, but the most efficient method for low-level manipulation is a local polyhedral approximation. Figure 2 shows an example of such a model, and illustrates the computation of curved occluding boundaries for matching to derived image edges.

Figure 3 shows the matching and parameter solving for a hand-drill model with curved surfaces. The matching

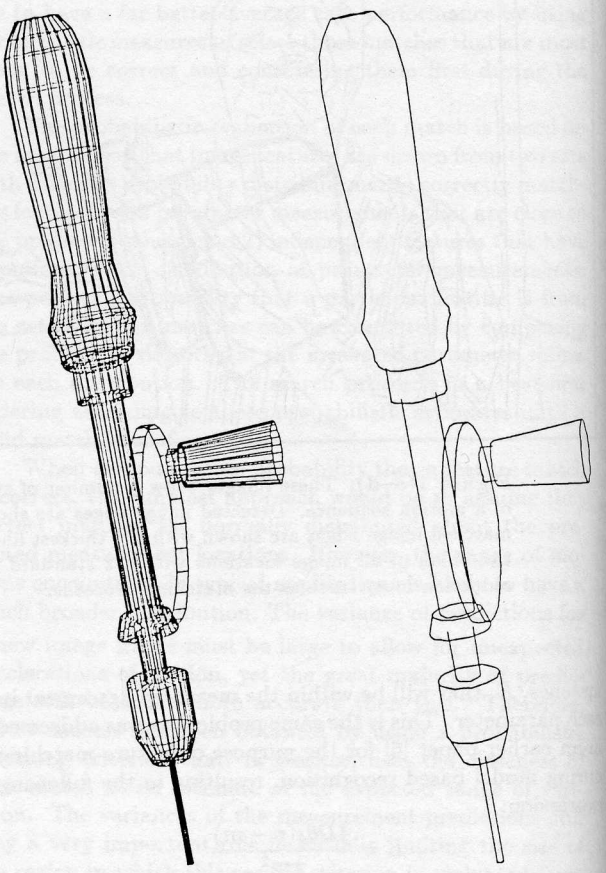

Figure 2: An example of a model with curved surfaces and an internal parameter specifying rotation of the handle. The underlying approximating patches are shown on the left, and the generated contours for matching are shown on the right.

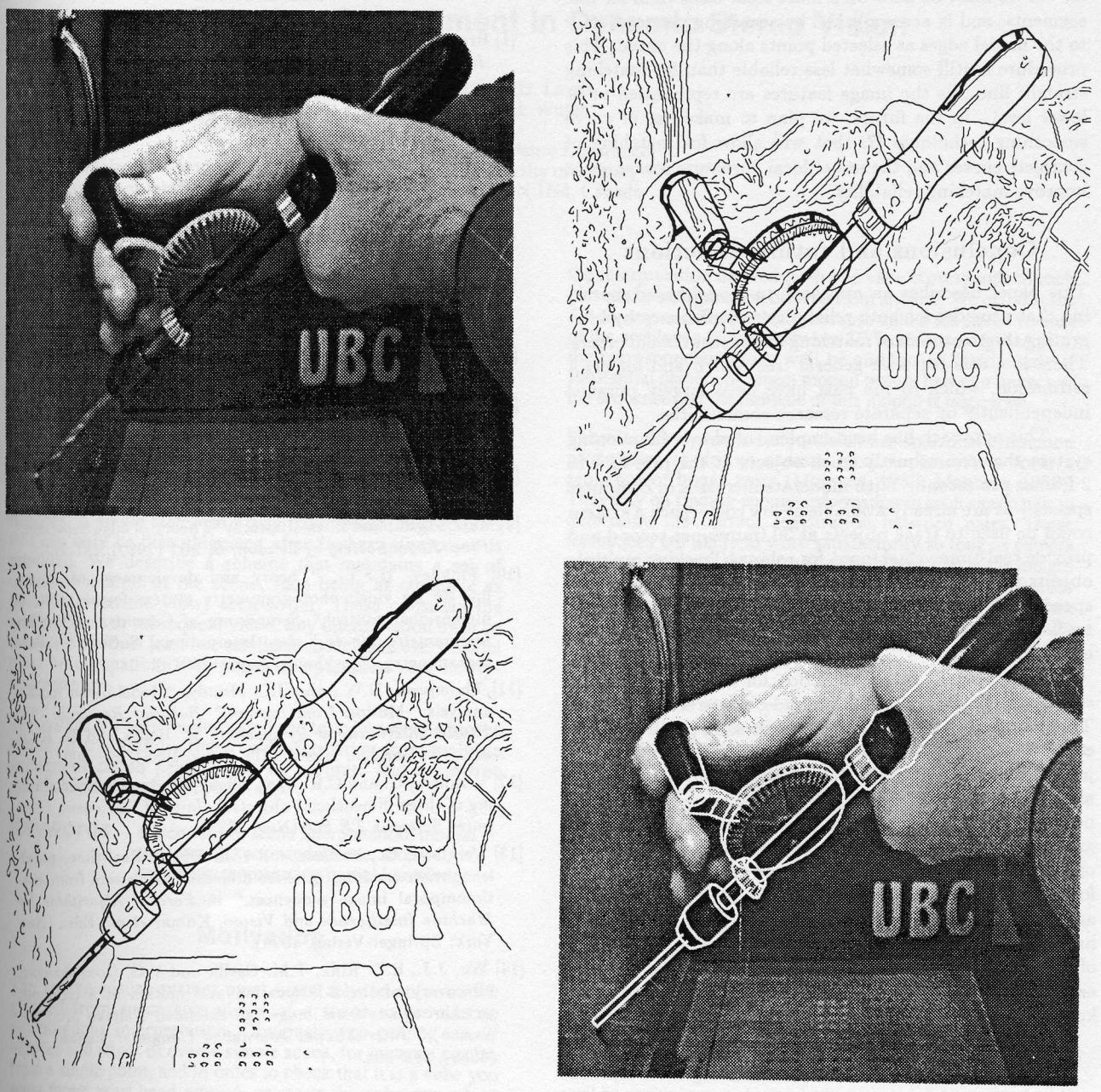

Figure 3: (a) An image from a motion sequence of a person using a hand drill. (b) Superimposed on edges extracted from the image are the model from its previous estimated viewpoint, nearby matching edges, and perpendicular errors to be minimized. (c) The new model position and handle rotation after one iteration of model fitting. New matches to image edges are shown with heavy lines. (d) After the second iteration of convergence, the model is shown superimposed on the original image.

for curves must be done on a more local basis than for line segments, and is accomplished by searching perpendicular to the model edges at selected points along the curve. This procedure is still somewhat less reliable than for matching straight lines, as the image features are represented at at lower level. In the future, we plan to make use of curve smoothing techniques [7] that will allow for matching of complete curves and the use of local curvature measures to improve matching reliability.

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, which in the past have been largely pursued independently by separate research communities.

This approach has been implemented in a functioning system that can robustly track objects at the rate of 1 to 2 frames per second. With moderate increases in computer speeds—as are already available at low cost—such a system could be used to track objects at 30 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 [5]. This would be used to initialize the tracking process, which could then proceed efficiently while also acquiring new objects in the field of view. 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 development 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 clear trend is towards the integration of recognition and tracking, which would be simply different ends of a continuum representing the degree of prior knowledge regarding the locations of objects in an image.

References

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

[2] Dickmanns, E., and V. Graefe, "Dynamic monocular machine vision," *Machine Vision and Applications,* **1** (1988), 223–240.

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

[4] Lowe, David G., *Perceptual Organization and Visual Recognition* (Boston, Mass: Kluwer Academic Publishers, 1985).

[5] Lowe, David G., "Three-dimensional object recognition from single two-dimensional images," *Artificial Intelligence,* **31,** 3 (March 1987), 355-395.

[6] Lowe, David G., "The viewpoint consistency constraint," *International Journal of Computer Vision,* **1,** 1 (1987), 57-72.

[7] Lowe, David G., "Organization of smooth image curves at multiple scales," *International Journal of Computer Vision* **3,** 2 (June 1989), 119–130.

[8] Lowe, David G., "Fitting parameterized 3-D models to images," to appear in *IEEE PAMI* (1991).

[9] Marr, David, and Ellen Hildreth, "Theory of edge detection," *Proc. Royal Society of London, B,* **207** (1980), 187-217.

[10] Pinkney, H.F.L., "Theory and development of an on-line 30 Hz video photogrammetry system for real-time 3-dimensional control," *Symposium of Commission V, Photogrammetry for Industry,* International Society for Photogrammetry (Stockholm, August, 1978).

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

[12] Verghese, G. and C.R. Dyer, "Real-time model-based tracking of three-dimensional objects," *Univ. of Wisconsin, Computer Sciences TR 806* (Nov. 1988).

[13] Verghese, G., K. Gale and C.R. Dyer, "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., (New York: Springer-Verlag, 1990).

[14] Wu, J.J., R.E. Rink, T.M. Caelli, and V.G. Gourishankar, "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 (1989), 373–394.