Proceedings of 1995 IEEE International Conference on Robotics and Automation, Nagoya, Japan, 1995.

Vision-Based Motion Tracking of Rigid Objects Using Prediction of Uncertainties

Akio Kosaka

School of Electrical Engineering
Purdue University
West Lafayette, IN 47907-1285, U.S.A.

Goichi Nakazawa

Strategic Business Development
Olympus America Inc.
Lake Success, NY 11042-1179, U.S.A.

Abstract

A vision-based motion tracking method described in this paper estimates the 3D position and orientation of a moving object of known shape at an average speed of 2.5 seconds per image frame even in complex environments using a conventional computer power. Given a coarse estimate of the initial 3D object pose, the method first generates an expectation view from which visible model features are automatically selected. The method then extracts potentially matched image features from image regions bounded by the propagation of object motion uncertainty. The special aspect of our vision-based tracking is an optimal correspondence search for model features and image features in which we use a Kalman filter-based updating scheme to perform the precise 3D object pose estimation. Experimental results are presented to demonstrate the robustness of the method even in the presence of occlusion.[1]

1 Introduction

Motion tracking using visual feedback is an important issue in robotic applications. For real-time applications, many researchers have developed hand-eye coordination algorithms in the field of visual servoing [6], [2]. Due to the fast processing speed required for real-time operation, most of the reported approaches require simple environments without any occlusion.

Recently, model-based motion tracking using visual feedback has also been introduced. Harris [5] and Lowe [9] developed object pose estimation algorithms using 3D wireframe models of polyhedral objects. Such algorithms use an expectation view of polyhedral objects and compare it with line features extracted from the image to estimate the object pose. Harris applied his method to the tracking of airplanes [5]. In his method, control feature points are selected from the expectation view, and the distance from each control point to the closest image line is used to estimate the object pose. This algorithm, therefore, requires an expectation view to be sufficiently close to the actual view. Lowe developed a general scheme for the 3D object pose estimation in more complicated cases, suggesting the applicability of this method to the motion tracking of 3D objects [9]. The algorithm first generates an expectation view and selects the line segments as model features to be tracked. The algorithm next extracts all line segments from the whole image by applying an edge detector and performing a grouping procedure for extracted edges. Easily identified image lines corresponding to significant visible model lines are then selected to generate a partial correspondence between the model lines and the image lines and to update the object pose. This updated object pose is then used to find a further correspondence between other unmatched model lines and image lines. In his method, therefore, if the image features forming the partial correspondence are incorrectly selected, the algorithm does not guarantee to generate an appropriate estimate for the object pose.

In this paper, we will introduce a new motion tracking method using model-based reasoning and prediction of uncertainties. This method is similar to the work by Harris and Lowe in usage of an expectation view to estimate the object position and orientation. *However, we provide a more robust tracking algorithm even in a complicated environment with occlusion, as one shown in Fig. 5, taking full advantage of uncertainties involved in prediction and observation.* The fundamental idea of this method is derived from our previous work called *FINALE*, a mobile robot navigation architecture using model-based vision [7]. *FINALE* was mainly developed to guide a robot to a predetermined 2D destination by updating its predicted robot position through visual observations.

A new motion tracking method described in this

[1] This work was done while the authors were at Olympus Optical Co., Ltd., Tokyo, Japan.

paper, however, deals with an image sequence taken by a single camera on a real time basis, and precisely estimates the 3D position and orientation of a moving object of known shape, relative to the camera coordinate frame, by iteratively processing image frames in the sequence. In this framework, the camera position is not necessarily stationary. Even in the case that the camera is moving in the world coordinate frame, the method has the capability of estimating the 3D object pose relative to the camera coordinate frame. In this paper, we will first introduce this new motion tracking method and will then show some experimental results.

2 Motion Tracking Algorithm

An object to be tracked is represented in a local coordinate frame. In this object coordinate frame, the object is approximated by a polyhedral shape, and is represented by a wireframe model in a conventional way [3]. In our current representation of the object, 3D line segments constituting the wireframe are selected for model features to be tracked over image frames. A single camera is set up in the environment, and its location is not necessarily stationary relative to the environment. In order to relate the object frame to the camera frame, we specify the origin of the object coordinate frame by a 6D vector $\mathbf{p} = [p_x, p_y, p_z, \phi_x, \phi_y, \phi_z]$ where (p_x, p_y, p_z) represents the origin of the object coordinate frame relative to the camera coordinate frame, and (ϕ_x, ϕ_y, ϕ_z) the three angles of yaw (ϕ_x), pitch (ϕ_y), and roll (ϕ_z) [4]. This 6D vector $\mathbf{p}$ will be called the *object pose vector* in this paper. *The problem of motion tracking is, therefore, to estimate this object pose vector over a time period using visual observations.* In practice, we deal with this object pose vector as a random vector, and specify the statistics of this random vector by the mean vector $\bar{\mathbf{p}}$ and its covariance matrix Σ as described in the following subsections.

Fig. 1 shows the overall framework of our motion tracking method. *Model Database* provides a 3D wireframe model of the object which is represented in the object coordinate frame. *Object Pose Prediction Module* predicts the uncertainty involved in the object motion using the previously estimated motion trajectory, and then predicts the object pose vector for the next image frame. *Model Feature Prediction Module* first generates an expectation view, using the 3D wireframe model of the object and the predicted value of the object pose vector. From this expectation view, *Model Feature Prediction Module* then automatically selects significant model features, and

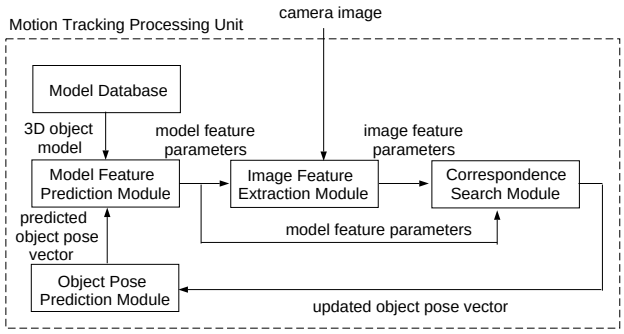


Figure 1: *Overall framework for motion tracking.*

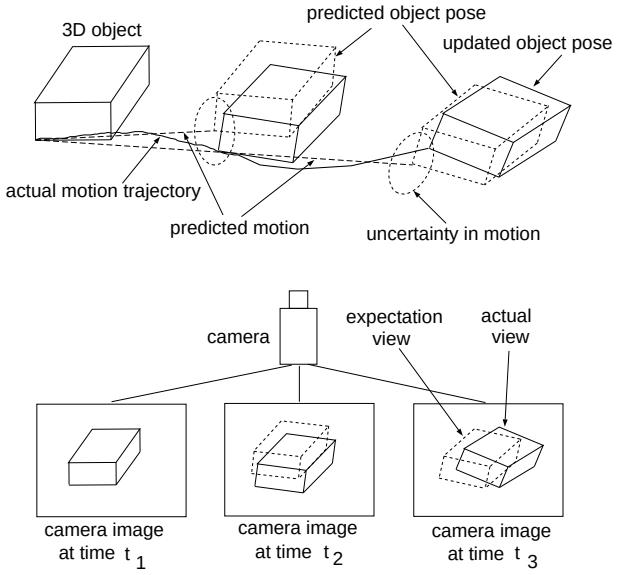


Figure 2: *Basic idea of our motion tracking method.*

plans observation strategies such as *from which regions of the image* one should extract the corresponding image features. Based on these strategies, *Image Feature Extraction Module* efficiently extracts potentially matched image features from the image. Upon the availability of model features and image features, *Correspondence Search Module* finds an optimal correspondence between the model features and the image features, when multiple image features are extracted from the restricted image regions. Using the obtained optimal correspondence, *Correspondence Search Module* also provides the updated object pose vector and sends this updated estimate to *Object Pose Prediction Module*. Using this updated estimate, *Object Pose Prediction Module* again predicts the object pose vector for the next frame. This prediction-and-update cycle is iterated in our motion tracking method.

Fig. 2 shows the basic idea of our motion tracking

method. The upper part of Fig. 2 displays the motion of a 3D object in the time sequence. The dotted lines represent the predicted pose of the object, and the solid lines the actual pose of the object. Using the previously obtained estimates of the object pose, the uncertainty involved in the object motion is predicted; this uncertainty is represented by the dotted ellipsoid. The lower part of the figure shows the images taken at times t_1, t_2, t_3. At each time, the method generates an expectation view of the object to select visible model features. The method then updates the object pose by an optimal correspondence between model features and image features extracted from the image. We will explain the details in the following subsections:

2.1 Prediction of Initial Object Pose Vector

We assume that the object pose vector for the first frame is approximately known with a certain level of uncertainty. More precisely, the uncertainty involved in the object pose vector $\mathbf{p}_1$ for the first frame is specified by the pair of prediction mean vector $\bar{\mathbf{p}}_1$ and the prediction error covariance matrix Σ_1. This uncertainty $(\bar{\mathbf{p}}_1, \Sigma_1)$ is called the *object pose uncertainty.* For example, the positional deviation for the initial frame can be as large as the half size of the object, and the rotational derivation can be as large as 10 degrees in each rotational parameter of ϕ_x, ϕ_y, ϕ_z.

2.2 Prediction of Visible Model Features

Using a wireframe of the object and the initial estimate of the predicted mean values of object pose vector, our algorithm first generates an expectation view from the camera viewpoint. In Fig. 3, dashed lines show such an expectation view. Algorithms that generate such an expectation view are well known in [3]. From this expectation view, line segments whose lengths in the view are larger than a prespecified threshold are selected as visible model features so as to be easily extracted from an actual image.

A key element of this vision-based motion tracking method is the propagation of the object pose uncertainty into both the camera image frame and the feature space in which many of image features used for model matching are easily extracted. In fact, the locational uncertainty of model features in the camera image (expectation view) can also be predicted by the mean vector and the covariance matrix. Such a propagation can be realized by projecting the prediction error covariance Σ of the object pose vector $\mathbf{p}$ into both the camera image and the feature space, using

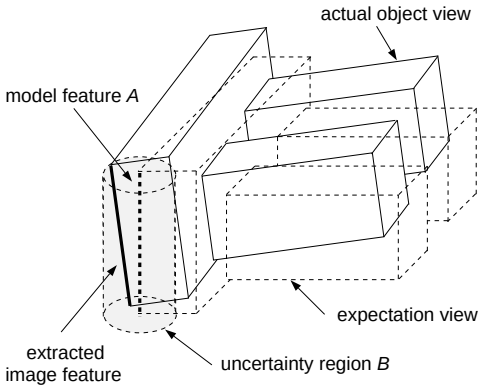


Figure 3: *Expectation view and uncertainty region.*

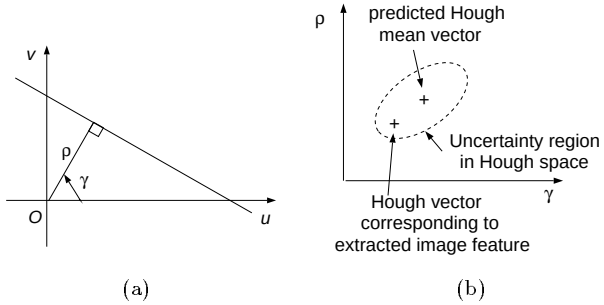


Figure 4: *(a) The Hough transform uses two parameters ρ and γ. (b) The uncertainty region in the Hough space can also be specified by the prediction mean vector and the error covariance matrix.*

the Jacobian matrices of the transformations involved [7].

The uncertainty propagated into the camera image bounds the regions where the model features should exist in the camera image. For example, applying an appropriate threshold to the covariance, we are able to define such a region. We will call this region the *uncertainty region* in the camera image. In Fig. 3, the uncertainty region for model feature A is shown as the shaded area B, which is created by taking the enclosure of the uncertainty regions of two endpoints of model feature A. Fig. 6 shows the expectation view with such uncertainty regions of visible model features in the camera image whose original image is shown in Fig. 5. Similarly, we can propagate the object pose uncertainty to a feature space such as the Hough space [7]. The Hough space, formed by the Hough transform, is specified by two parameters $\mathbf{q} = (\rho, \gamma)$ to represent a line in the camera image – the perpendicular distance ρ from the image origin to the line,

and the orientation angle γ of the normal vector to the line in the image as shown in Fig. 4 (a). The propagation of the object pose uncertainty into the Hough space also restricts the region where these Hough parameters take values. Fig. 4 (b) shows such a region of a particular model feature where the region is specified by the predicted Hough mean vector $\bar{\mathbf{q}}$ and the error covariance matrix Q.

2.3 Extraction of Image Features

For each model feature, image features are extracted from its uncertainty region in the camera image. *We, therefore, do not need to make any effort in extracting image features outside the uncertainty regions.* Since we use the Hough transform to extract image features, this extraction process is facilitated by applying the Hough transform to only the uncertainty region in the Hough space. In other words, image features are extracted from the image by restricting the regions in both the image space and the Hough space. In our method, the extracted image features are represented by the Hough parameters $\hat{\mathbf{r}} = (\hat{\rho}, \hat{\gamma})$. Fig. 7 shows extracted image features corresponding to the visible model features in Fig. 6.

2.4 Updating of Object Pose Vector Through Optimal Correspondence Search

Due to visual occlusions and background conditions, the line extraction algorithm may extract more than one image feature in the uncertainty region. In order to precisely estimate the object pose vector $\mathbf{p}$, we need to find a correct correspondence between model features and image features. Our algorithm performs an optimal correspondence search using the parametric reasoning for model and image feature relations, originally introduced by *FINALE* [7]. This algorithm, performing a constrained search, is verified to be robust even in the presence of occlusion and with a large displacement of expectation and actual views [7].

A key idea of this constrained search is that a single match between a model feature and an image feature constrains the object pose uncertainty, updating the statistics of that uncertainty. In practice, we use the Kalman filtering to realize this updating. Given a single match of model feature $\mathbf{s}$ and image feature $\mathbf{r}$, object pose uncertainty represented by the pair of the mean vector and the error covariance matrix $(\bar{\mathbf{p}}, \Sigma)$ is updated to the pair of the new mean vector and the new covariance matrix $(\bar{\mathbf{p}}^{(new)}, \Sigma^{(new)})$. *Since this update reduces the object pose uncertainty, it also reduces the uncertainty regions in both image and feature spaces for other model features, and therefore allows subsequent model features to be easily matched to image features in the subsequent search process.* In addition, the new estimate of the object pose vector is indeed obtained as the by-product of the Kalman filter-based updating.

In the actual steps of the search process, the algorithm looks for an optimal correspondence in a branch-and-bound depth-first search mode [7]. We introduce optimal criteria based on the match probability of model features and image features to evaluate each correspondence. The detail of this optimal correspondence search will be shown in the next section.

Fig. 8 shows the result of the optimal correspondence search. In this figure, model features are projected back onto the camera image using the updated object pose vector. Therefore, Fig. 8 demonstrates the accuracy of our pose estimation process.

2.5 Prediction of Object Pose Vector for Next Frame

Once the object pose vector is estimated through the optimal correspondence search, the object pose vector for the next image frame is predicted using a linear prediction scheme. Let t_k denote the time index to the next frame to be predicted and t_{k-1}, t_{k-2} be the time indices to the previous two frames whose object pose vectors $\mathbf{p}_{k-1}$ and $\mathbf{p}_{k-2}$ have been already estimated. Then the object pose vector $\mathbf{p}_k$ at time t_k can be predicted by

$$\mathbf{p}_k = \mathbf{p}_{k-1} + \frac{\mathbf{p}_{k-1} - \mathbf{p}_{k-2}}{t_{k-1} - t_{k-2}}(t_k - t_{k-1}) + \epsilon_k \qquad (1)$$

where ϵ_k represents the error component for the prediction, and its mean vector and covariance matrix are assumed to be known from the a priori knowledge of the object motion. The prediction error component ϵ_k affects the object pose uncertainty for the next frame t_k, and again the object pose uncertainty of the next frame is represented by the prediction mean vector $\bar{\mathbf{p}}_k$ and the prediction error covariance matrix Σ_k, which can be directly computed from the statistics of ϵ_k.

In our current implementation of the next frame prediction, we use the simple linear prediction scheme expressed in Eq. (1). Of course, we will be able to use a more sophisticated scheme. But as will be shown in the experimental results, our method of pose estimation is robust enough to apply the above linear prediction scheme.

3 Optimal Correspondence Search

As described in the previous section, we wish to deal with a correspondence search using some optimality criteria. This section will briefly present how we define optimality criteria based on the probability of match. Let us assume that each model feature $\mathbf{s}_i$ $(i = 1, 2, ..., n)$ has a candidate pool $\mathbf{B}_i$ of image features $\mathbf{B}_i = \{r_i{}^1, r_i{}^2, ..., r_i{}^m, nil\}$ where $r_i{}^j (j = 1, 2, ..., m)$ represents an image feature extracted from the image, and *nil* symbol denotes the case that no image feature is matched to model feature $\mathbf{s}_i$. Such a case happens especially when occlusion is involved. Our correspondence search problem is then finding how to select a single candidate $\mathbf{r}_i$ from $\mathbf{B}_i$ for each model feature $\mathbf{s}_i$. We define the optimality criteria according to [7]. More specifically, an optimal correspondence is a match of $(\mathbf{s}_1, \mathbf{r}_1)$, $(\mathbf{s}_2, \mathbf{r}_2)$, ... , $(\mathbf{s}_n, \mathbf{r}_n)$ such that

(1) the number of *nils* in $\{\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_n\}$ is minimum,

(2) an objective function D becomes optimal.

The first condition states that we like as many model features as possible to be matched to image features, of course without violating parametric constraints among these features. The second condition expresses how well such parametric constraints should be satisfied in terms of objective function D. We now explain how we will select this objective function D.

Before *Correspondence Search Module* finds an optimal correspondence between model features and image features, we are given the object pose uncertainty $(\bar{\mathbf{p}}, \Sigma)$ of object pose vector $\mathbf{p}$. So we consider the probability that model feature $\mathbf{s}$ is matched to image feature $\mathbf{r}$ given the object pose vector $\mathbf{p}$:

$$prob(\mathbf{s} \rightarrow \mathbf{r} \mid \mathbf{p}). \tag{2}$$

As we described in the previous section, we are able to propagate the object pose uncertainty into the Hough space. For each model feature $\mathbf{s}$, we represent the propagated Hough space uncertainty also by the prediction mean vector $\bar{\mathbf{q}}$ and the prediction error covariance matrix Q. Given the knowledge of the prediction of the object pose vector $\mathbf{p}$, it is natural to say that if an extracted image feature denoted by its measured vector $\mathbf{r}$ is closer to the predicted Hough mean vector $\bar{\mathbf{q}}$, model feature $\mathbf{s}$ is more likely to be matched to image feature $\mathbf{r}$. In practice, we approximate the probability of match in Eq. (2) by

$$prob(\mathbf{s} \rightarrow \mathbf{r} \mid \mathbf{p}) = \frac{1}{2\pi\sqrt{|Q|}} exp\left[-\frac{1}{2}(\mathbf{r} - \bar{\mathbf{q}})^T Q^{-1} (\mathbf{r} - \bar{\mathbf{q}})\right].$$

Now we consider the probability of match for two pairs of model features and image features: $(\mathbf{s}_1, \mathbf{r}_1)$ and $(\mathbf{s}_2, \mathbf{r}_2)$. This probability can be written using the conditional probability as

$$prob(\mathbf{s}_1 \rightarrow \mathbf{r}_1, \mathbf{s}_2 \rightarrow \mathbf{r}_2 \mid \mathbf{p}) = prob(\mathbf{s}_1 \rightarrow \mathbf{r}_1 \mid \mathbf{p}) \times prob(\mathbf{s}_2 \rightarrow \mathbf{r}_2 \mid \mathbf{s}_1 \rightarrow \mathbf{r}_1, \mathbf{p}).$$

Let us consider the condition part like $(\mathbf{s} \rightarrow \mathbf{r}, \mathbf{p})$ of this probability. When we are given the object pose vector $\mathbf{p}$ and the match of model feature $\mathbf{s}$ and image feature $\mathbf{r}$, a parametric constraint among these three vectors can be expressed by the following form:

$$\mathbf{f}(\mathbf{s}, \mathbf{r}, \mathbf{p}) = \mathbf{0} \tag{3}$$

where $\mathbf{f}$ is a two-dimensional function since the measurement $\mathbf{r}$ is a two-dimensional vector, and its actual form is shown in [7]. Given such a parametric constraint equation, we are able to revise the statistics of object pose vector $\mathbf{p}$. Such a revision can be realized by the extended Kalman filter developed by Ayache and Faugeras [1]. In reality, the measurement of the Hough parameters $\mathbf{r}$ is contaminated by noise, and we represent the measurement by the measured Hough parameters $\hat{\mathbf{r}}$ and the expected error covariance matrix R. Then the parametric constraint of Eq. (3) revises the statistics of the object pose vector to

$$\begin{aligned} \bar{\mathbf{p}}^{(1)} &= \bar{\mathbf{p}} - K\,\mathbf{f}(\mathbf{s}, \hat{\mathbf{r}}, \bar{\mathbf{p}}) \\ \Sigma^{(1)} &= (I - KM)\,\Sigma \end{aligned}$$

where K, M and W are computed by

$$\begin{aligned} K &= \Sigma\, M^T\, (W + M\Sigma M^T) \\ M &= \frac{\partial \mathbf{f}(\mathbf{s}, \hat{\mathbf{r}}, \bar{\mathbf{p}})}{\partial \mathbf{p}} \\ W &= \left(\frac{\partial \mathbf{f}(\mathbf{s}, \hat{\mathbf{r}}, \bar{\mathbf{p}})}{\partial \mathbf{q}}\right) R \left(\frac{\partial \mathbf{f}(\mathbf{s}, \hat{\mathbf{r}}, \bar{\mathbf{p}})}{\partial \mathbf{q}}\right)^T. \end{aligned}$$

The important thing here is that the object pose uncertainty $(\bar{\mathbf{p}}, \Sigma)$ is updated to $(\bar{\mathbf{p}}^{(1)}, \Sigma^{(1)})$ through the constraint expressed in Eq. (3). Therefore, using the updated uncertainty $\mathbf{p}^{(1)} = (\bar{\mathbf{p}}^{(1)}, \Sigma^{(1)})$, the probability of match for the two pairs can be interpreted as

$$prob(\mathbf{s}_1 \rightarrow \mathbf{r}_1, \mathbf{s}_2 \rightarrow \mathbf{r}_2 \mid \mathbf{p}) = prob(\mathbf{s}_1 \rightarrow \mathbf{r}_1 \mid \mathbf{p}) \times prob(\mathbf{s}_2 \rightarrow \mathbf{r}_2 \mid \mathbf{p}^{(1)}). \tag{4}$$

Now we consider a more general case such that the probability of match among n pairs of model features and image features, namely $(\mathbf{s}_1, \mathbf{r}_1)$, $(\mathbf{s}_2, \mathbf{r}_2)$, ..., $(\mathbf{s}_n, \mathbf{r}_n)$. By iteratively using Eq. (4), we obtain the probability as

$$\begin{aligned} D &= prob(\mathbf{s}_1 \rightarrow \mathbf{r}_1, \mathbf{s}_2 \rightarrow \mathbf{r}_2, ..., \mathbf{s}_n \rightarrow \mathbf{r}_n \mid \mathbf{p}) \\ &= prob(\mathbf{s}_1 \rightarrow \mathbf{r}_1 \mid \mathbf{p}) \times prob(\mathbf{s}_2 \rightarrow \mathbf{r}_2 \mid \mathbf{p}^{(1)}) \times ... \\ &\qquad \times prob(\mathbf{s}_n \rightarrow \mathbf{r}_n \mid \mathbf{p}^{(n-1)}). \end{aligned}$$

We use this probability as the objective function for the optimal correspondence search.

When the search procedure finds an optimal correspondence, the new estimate of object pose vector is automatically obtained as the by-product of the optimal search through the Kalman filter-based updating.

4 Experimental Results

We implemented this motion tracking method in a SUN Workstation SPARC 10. We tested the method with a rigid body object whose shape was similar to a manipulator hand as shown in Fig. 5. We set up a complicated background environment, and a human generated various types of object motion. We took six series of image frames in various situations to investigate the stability and the robustness of our method.

Due to the lack of mobile camera facilities, we have not moved the camera adaptively to the object motion in this experiment. We used a standard monochrome TV camera and a digital video recorder to capture 30 image frames in one second, and transferred them to the SUN Workstation as 8-bit monochrome images of 512×480 pixels. Prior to the experiment, the camera was calibrated as a pinhole camera model using the algorithm developed by Lopez-Abadia and Kak [8].

All processes in our method were done off-line in the SUN Workstation based on the images of 512×480 pixels. No special processor or hardware was used in this experiment. In the current stage, we estimated the initial pose of the object by manually setting the object at an approximately predetermined location and orientation. Every other frame (15 frames in one second) was processed, and each frame required approximately 2.5 sec in processing. In most cases, our method successfully tracked the object motion. Although we have not carefully examined, the positional error in estimation was less than 1 cm, with respect to the object size of 20 cm. The rotational error in estimation was less than 1 $degree$.

Figs. 5 through 10 show the experimental result for the case where an obstacle caused a significant visual occlusion. Fig. 5 shows one of the 30 original image frames, and Fig. 6 displays the expectation view with propagated uncertainty regions. In Fig. 6, the dotted ellipses represent the uncertainty regions for point features. For the extraction of line features, the uncertainty regions for line features were created by combining the uncertainty regions of two endpoint features. Then the algorithm used the Hough transform to extract line features within the bounded regions. The extracted image features are shown in Fig. 7. Using the optimal correspondence search procedures described before, we found an optimal correspondence. By using this optimal correspondence, the object pose vector was updated. Fig. 8 shows the result of optimal correspondence and estimation process. In the figure, the white lines indicate the model features that are reprojected onto the original image using the updated object pose vector. This figure, therefore, displays the accuracy of the pose estimation.

Fig. 9 shows four image frames processed just prior to the image frame shown in Fig. 8. Again the expectation view is reprojected onto the original image using the updated object pose vector. Although the obstacle causes a significant occlusion, the algorithm successfully tracks the object motion. Figs. 10 (a) and (b) show the pose estimation process over the entire sequence of image frames. Fig. 10 (a) shows the positional change of the z coordinate (p_z) over the image frames. (Note that the z axis is perpendicular to the camera image plane.) Dashed lines show the predicted object z position, and solid lines the updated z position. Fig. 10 (b) displays the change in roll angle ϕ_z. Dashed lines show the predicted roll angle, and solid lines the updated angle. As shown in Fig. 10 (b), initially the rotational deviation in roll ϕ_z was as large as 10 degrees, but such a large deviation is canceled out through this vision-based motion tracking. Since we simply use the linear prediction scheme shown in Eq. (1), the prediction is not necessarily smooth. However, the updated trajectory shows the stability of our method. Despite a large change in both the orientation and the location of the object and a visual clutter caused by an obstacle, these experiments verify the robustness of our method.

The average processing time for each frame requires approximately 2.5 sec on the SUN Workstation. The processing time in each module shown in Fig. 1 is: (1) Model Feature Prediction 0.10 sec, (2) Image Feature Extraction 2.0 sec, (3) Correspondence Search 0.30 sec, and (4) Object Pose Prediction 0.05 sec. The Image Feature Extraction is the most time-consuming process, since it requires image file manipulations on the SUN Workstation. This processing will be, of course, greatly reduced by using more hardware-oriented image processing tools.

Note that estimating the 3D pose of an object from a single image is sometimes suffering from an ill-posed problem. Since the output of our estimation consists of a pair of the mean vector and the error covariance matrix, the uncertainty caused from singularity in a single-frame estimation is automatically propagated to subsequent frames so that such uncertainty may be reduced during the processing in the subsequent frames.

5 Conclusions

This paper presented a robust vision-based motion tracking algorithm using the Kalman filter-based prediction/update scheme. Estimating the 3D position and the orientation of a moving object of known shape, the algorithm successfully tracked the object motion even when partial occlusions were involved. Currently, we are planning to quantitatively evaluate the robustness of our algorithm. Our future research subjects include the speed-up toward a real-time operation as well as the automatic coarse estimation of the initial pose of the object.

References

[1] N. Ayache and O. D. Faugeras, "Maintaining representations of the environment of a mobile robot," *IEEE Transactions on Robotics and Automation*, Vol. 5, No. 6, pp. 804-819, 1989.

[2] J. T. Feddema and C. S. G. Lee, "Adaptive image feature prediction and control for visual tracking with a hand-eye coordinated camera," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 20, No. 5, pp. 1172-1183, 1990.

[3] J. Forley, et al, *Computer Graphics*, 2nd edition, Addison-Wesley, Reading, Massachusetts, 1990.

[4] K. S. Fu, R. C. Gonzalez, and C. S. G. Lee, *Robotics - Control, Sensing, Vision, and Intelligence*, McGraw-Hill, New York, 1987.

[5] C. Harris, "Tracking with rigid models," *Active Vision*, edited by A. Blake and A. Yuille, MIT Press, Cambridge, Massachusetts, 1992.

[6] K. Hashimoto (ed.), *Visual Servoing*, World Scientific, Singapore, 1993.

[7] A. Kosaka and A. C. Kak, "Fast vision-guided mobile robot navigation using model-based reasoning and prediction of uncertainties," *Computer Vision, Graphics, and Image Processing – Image Understanding*, Vol. 56, No. 3, pp.271-329, 1992.

[8] C. Lopez-Abadia and A. C. Kak, *Vision-Guided Mobile Robot Navigation*, TR-EE 89-34, Electrical Engineering, Purdue University, Indiana, 1989.

[9] D. G. Lowe, "Robust model-based motion tracking through the integration of search and estimation," *International Journal of Computer Vision*, Vol. 8, No. 2, pp.113-122, 1992.

Figure 5: *Shown here is an original image frame used for the motion tracking. An obstacle significantly occludes a part of the object. The complicated background also causes a difficulty in object recognition using conventional methods. This image was taken at time index 10 in Fig. 10.*

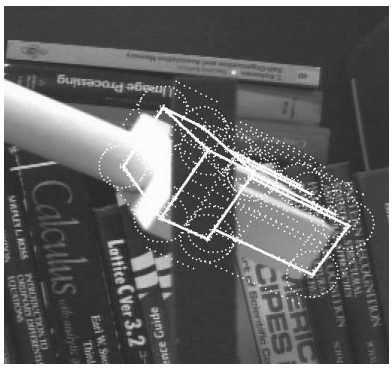


Figure 6: *An expectation view is superimposed onto the original image shown in Fig. 5. Bold white lines represent model features used for motion tracking. The dotted ellipses and lines represent the uncertainty regions for model features.*

Figure 7: *Extracted image features are superimposed onto the original image shown in Fig. 5.*

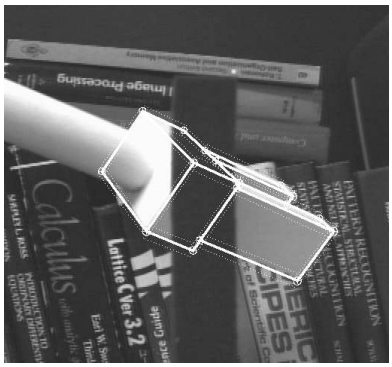
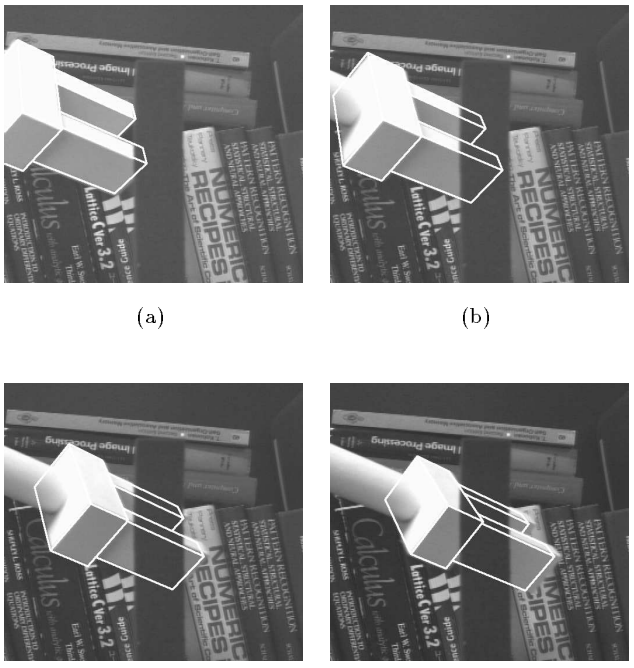


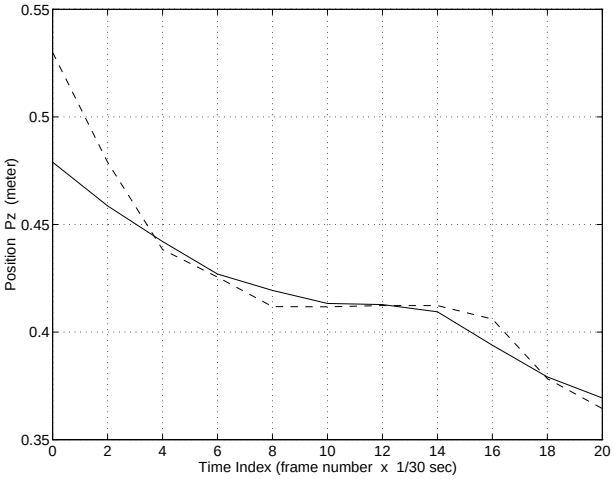
Figure 8: *The model wireframe view is reprojected onto the original image using the newly estimated object pose vector. This figure, therefore, demonstrates the accuracy of the pose estimation.*



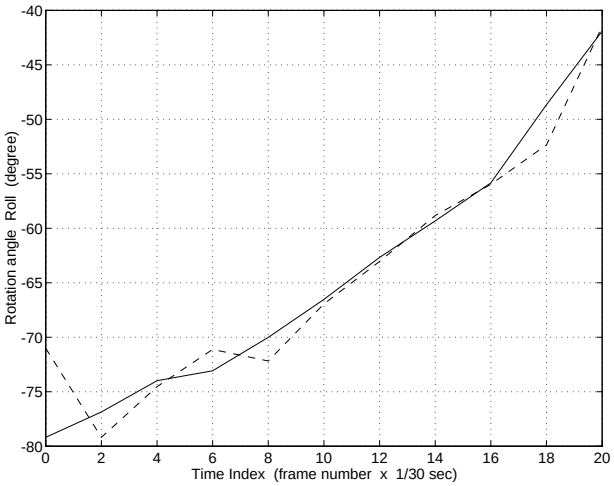
(a) (b)

(c) (d)

Figure 9: *The object pose vector is estimated over the image frames before the time of the image frame shown in Fig. 8. These frames were taken at time indices 2, 4, 6, and 8 shown in Fig. 10. Again the wireframe view is reprojected onto the original image using the updated object pose vector.*



(a)



(b)

Figure 10: *(a) The predicted and updated estimates of object position p_z over the image frames are shown. (b) The predicted and updated estimates of object orientation ϕ_z (roll angle) over the image frames are shown. In both figures, the dashed lines indicate the prediction, and the solid lines the update. Note that the time index shows the frame number of the image taken every 1/30 sec.*