# ESTIMATING THE POSE AND MOTION OF A KNOWN OBJECT FOR REAL-TIME ROBOTIC TRACKING

Olli Silvén
Department of Electrical Engineering
University of Oulu

SF-90570, Oulu
Finland

## ABSTRACT

An approach for estimating the pose and motion of a known moving object in three dimensions from a sequence of monocular images is considered. The principle is to obtain initial estimates of the pose and motion parameters and to update them by using feature location measurements made from subsequent monocular image frames. The ultimate goal is to use the obtained estimates for controlling the movements of a robot arm.

## 1. INTRODUCTION

Machine vision holds great potential for increasing the autonomity of cargo handling systems, mining equipment, robotic manipulators, and other moving machines. In these applications vision systems must be able to produce accurate real time responses to control the movements.

In our experimental set-up intended for simulating this problem area, a camera is attached to a robot arm used for vision controlled tracking of a known moving object. The problems of this system, shown in Figure 1, are similar to real applications: image acquisition, image analysis and the movements do not occur instantly but add delays to the control loop. Thus, the robot must be controlled on the basis of the computed future poses of the object.

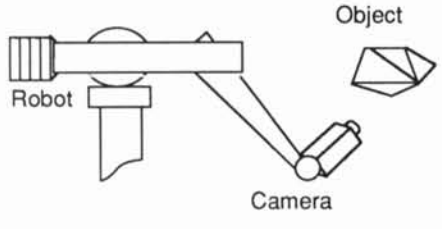


Figure 1: Experimental system

However, the future situations cannot be estimated perfectly, because the measurements made with image sensors are not infinitely accurate and the object motion may evolve unpredictably during the control delay. These uncertainties need to be taken into account in predictions and when integrating new observations into earlier knowledge.

Kalman filtering has been shown to be a powerful tool in these kinds of problems, e.g., in tracking objects from monocular image sequences [1,2,3,5,9] and robot navigation [7]. In particular, our approach has been greatly influenced by the applications and principles presented in [2] and [3].

Here, the problem is mapped onto the extended Kalman filtering framework. The strategy is to first obtain initial pose and motion estimates by using the pose estimation techniques of [6] and [8]. The estimates are updated with subsequent feature location measurements in the image plane. Simple approaches for modeling the motion and measurement uncertainties and for selecting features are considered.

## 2. BASIC SOLUTION

The basic principle is to perform pose and motion estimation simultaneously by using Kalman filtering. For this purpose the problem is expressed as system (1) and measurement (2) equations.

$$u_k = \Phi_{k,k-1} u_{k-1} + \epsilon_k \quad (1)$$

$$v_k = F_k(u_k) + \nu_k \quad (2)$$

The state vector $u_k$ consists of the object pose and motion parameters. $\Phi_{k,k-1}$ is a state transition matrix that produces the state at time step $k$ from the state at time step $k-1$. The motion modeling uncertainty at each time step is taken into account by an additive noise vector, $\epsilon_k$, whose covariance matrix is $W_k = E(\epsilon_k \epsilon_k^T)$.

The measurement process is modeled by a nonlinear function $F_k$ that produces the measurement vector $v_k$, or the image plane coordinates of the selected features, from system state $u_k$. The measurement uncertainty is modeled by an additive noise vector $\nu_k$, $V_k = E(\nu_k \nu_k^T)$. In practice, the uncertainty estimates for features are 2-by-2 covariance matrices, because the locations are (x,y) coordinate pairs. When $n$ features are used $V_k$ is a composite 2n-by-2n matrix built from the individual covariance matrices.

By assuming that both $\epsilon_k$ and $\nu_k$ are sequences of Gaussian distributed, independent zero mean random vectors, the extended Kalman filtering cycle becomes

$$u_{k|k-1} = \Phi_{k,k-1} u_{k-1|k-1} \quad (3)$$

$$Q_{k|k-1} = \Phi_{k,k-1} Q_{k-1|k-1} \Phi_{k,k-1}^T + W_{k-1} \quad (4)$$

$$v_{k|k-1} = F_k(u_{k|k-1}) \quad (5)$$

$$K_k = Q_{k|k-1} J_F^T(u_{k|k-1}) (V_k + J_F(u_{k|k-1}) Q_{k|k-1} J_F^T(u_{k|k-1}))^{-1} \quad (6)$$

$$u_{k|k} = u_{k|k-1} + K_k(v_k - v_{k|k-1}) \quad (7)$$

$$Q_{k|k} = (I - K_k J_F(u_{k|k-1})) Q_{k|k-1} \quad (8)$$

Equations (3) and (4) predict the state vector, $u_{k|k-1}$, and the estimation covariance matrix, $Q_{k|k-1}$ that reflects the uncertainties of this *a priori* estimate. The new measurement is predicted (5).

The *a posteriori* state estimate, $u_{k|k}$, of the state is produced in (7) by weighting the innovation, $(v_k - v_{k|k-1})$, by Kalman gain $K_k$ (6). $J_F(u_{k|k-1})$ is the Jacobian of the measurement function $F(u_{k|k-1})$, or the matrix of first order derivatives of feature location coordinates in the camera image plane frame with respect to the state variables in the motion estimation frame. Finally, the estimation covariance matrix is updated by (8) before the cycle starts again from the state prediction.

It is useful to limit the image processing efforts to regions where the interesting features are most likely to be found. These uncertainty windows can be approximated from the covariance matrix

$$V_{k|k-1} = J_F(u_{k|k-1}) Q_{k|k-1} J_F^T(u_{k|k-1})$$

## 3. SYSTEM MODEL

The tracking errors of the system depend essentially on the match between the *a priori* model of motion and the actual object behavior, unless the system control delay can be made very short. In practice, image acquisition and the mechanical system determine the minimum possible control delay.

When the camera sensor is mounted in the hand of a robot manipulator, the key questions in forming the object motion model are

1. Whether the state vector should be represented in a moving coordinate frame or a global stationary world coordinate system (relative or absolute object motion).

2. What motion parameters should be included in the state vector.

### 3.1 Coordinate system

If the target object is not capable of significant accelerations, its motion in a fixed world coordinate frame can be modeled as constant rotation and translation. Because the measurements are done in the camera frame, this approach requires knowledge of the robot joint angles at the time of capturing each image. However, the controllers of robot manipulators do not generally support obtaining the angles of different joints simultaneously during motion.

A practical compromise is to model the motion in a relatively slow moving coordinate frame, in which the angle and translation parameters of the camera are simple and fast to determine. A suitable origin for such a coordinate system could be at the wrist joint of the robot.

### 3.2 Motion parameters

If the state vector is represented in a moving coordinate system, modeling the expected relative accelerations as system noise may result in impractically high uncertainties. The uncertainties can be reduced by including the accelerations in the state vector and modeling the possible small changes of acceleration as noise. Unfortunately, this approach is computationally costly and may result in a slower sampling rate that in turn increases the motion uncertainties.

We have considered two simple techniques that limit the drawbacks of higher dimensionality or higher noise level to the periods of significant accelerations:

1. Variable dimension state vector. If the estimation errors grow rapidly, the acceleration components are added to the state vector. The switch-back to the constant velocity model is performed when the acceleration approaches zero or the errors go below a given threshold.

2. Variable system noise. A constant velocity motion model is used, but when the estimation errors grow the system noise model components attributed to motion are inflated. When the errors decrease the noise component is returned to the normal quiescent level.

Mapping these techniques onto the system equation is straightforward. For simplicity, we have assumed that each motion component is independent and the trajectory of the object in the used coordinate system is straight. Then the state transition matrix over one time increment for the constant velocity and constant acceleration motion models is

$$\Phi = \begin{pmatrix} \Phi_x & 0 & 0 & 0 & 0 & 0 \\ 0 & \Phi_y & 0 & 0 & 0 & 0 \\ 0 & 0 & \Phi_z & 0 & 0 & 0 \\ 0 & 0 & 0 & \Phi_\alpha & 0 & 0 \\ 0 & 0 & 0 & 0 & \Phi_\beta & 0 \\ 0 & 0 & 0 & 0 & 0 & \Phi_\gamma \end{pmatrix}$$

where $\Phi_x, ..., \Phi_\gamma$ are $\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$ or $\begin{pmatrix} 1 & 1 & \frac{1}{2} \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$, respectively. The corresponding system state vectors are $u_k = \begin{pmatrix} x(k) & \dot{x}(k) & y(k) & \dots & \gamma(k) & \dot{\gamma}(k) \end{pmatrix}^T$ and $u_k = \begin{pmatrix} x(k) & \dot{x}(k) & \ddot{x}(k) & \dots & \dot{\gamma}(k) & \ddot{\gamma}(k) \end{pmatrix}^T$.

The nondeterministic changes of motion have been assumed to be independent for each motion component.

## 4. START-UP

Before the Kalman filtering process can be started the initial state vector, $u_{0|0}$, and estimation covariance matrix, $Q_{0|0}$, need to be obtained.

By assuming that the velocities of the object are constant, the first pose and motion estimate can be obtained by using the method presented in [8] for the two first image frames. By denoting these with indices '-1' and '0' the state estimate becomes

$$\hat{u}_0 = \begin{pmatrix} x(0) \\ (x(0) - x(-1))/\tau \\ y(0) \\ . \\ . \\ (\gamma(0) - \gamma(-1))/\tau \end{pmatrix}$$

where $\tau$ is the sampling interval between frames.

The covariance matrix of the pose and motion estimate error can be estimated using the general expression

$$\hat{Q}_0 = E((u_0 - \hat{u}_0)(u_0 - \hat{u}_0)^T)$$

where

$$u_0 - \hat{u}_0 = \begin{pmatrix} -\eta_x(0) \\ (\eta_x(-1) - \eta_x(0))/\tau \\ -\eta_y(0) \\ . \\ . \\ (\eta_\gamma(-1) - \eta_\gamma(0))/\tau \end{pmatrix}$$

and $\eta_k = \begin{pmatrix} \eta_x(k) & . & . & . & \eta_\gamma(k) \end{pmatrix}^T$ is the pose measurement noise that is assumed Gaussian, zero mean, and independent for each frame.

Because the pose estimation algorithm [8] has solved for the correspondences between the features in the image and the object model, the pose $u_k$ can be obtained by the inverse of the measurement function

$$u_k = F_k^{-1}(v_k) = G_k(v_k)$$

By using the pose determination function $G_k(*)$ the covariances of the pose parameters, $V_u(u_k) = E(\eta_k \eta_k^T)$, can be approximated with

$$V_u(u_k) = J_G(v_k) V_v(v_k) J_G(v_k)^T \qquad (9)$$

where $J_G(v_k)$ is the Jacobian of $G_k(v_k)$ or its matrix of first partial derivatives, and $V_v(v_k)$ is the covariance matrix for feature measurements. Equation (9) is straightforward to solve numerically by computing $J_G(v_k)$ from the Jacobian of measurement function, because

$$G_k'(G_k^{-1}(x)) = \frac{1}{(G_k^{-1})'(x)}$$

or equivalently

$$J_G(v_k) = J_F^+(u_k)$$

where matrix pseudoinversion (+) is used, because $J_F(u_k)$ is not necessarily a square matrix.

By making a simplifying assumption that the standard deviation of the feature location error is 1 pixel, $V_v(v_k)$ becomes an identity matrix and the pose parameter covariance matrix can be approximated by

$$V_u(u_k) = \begin{pmatrix} V_{xx}(u_k) & . & . & V_{x\gamma}(u_k) \\ V_{xy}(u_k) & . & . & V_{y\gamma}(u_k) \\ . & . & . & . \\ . & . & . & . \\ V_{x\gamma}(u_k) & . & . & V_{\gamma\gamma}(u_k) \end{pmatrix} = (J_F^T(u_k) J_F(u_k))^+ \qquad (10)$$

where the pseudoinversion reduces to the conventional matrix inverse when $J_F^T(u_k) J_F(u_k)$ is non-singular.

Then the covariance matrix of the initial pose and motion error becomes

$$\hat{Q}_0 = \begin{pmatrix} V_{xx}(0) & . & . & V_{x\gamma}(0)/\tau \\ V_{xx}(0)/\tau & . & . & (V_{x\gamma}(0) + V_{x\gamma}(-1))/\tau^2 \\ . & . & . & . \\ . & . & . & . \\ V_{x\gamma}(0)/\tau & . & . & (V_{\gamma\gamma}(0) + V_{\gamma\gamma}(-1))/\tau^2 \end{pmatrix}$$

This is used as $Q_{0|0}$ in start-up.

## 5. SELECTION OF FEATURES

The images of real world objects often provide numerous features whose correspondences with the model can be recovered. In practice, it is useful to select only a few of the features for updating the pose and motion estimates. Here, three methods for selecting subsets of correspondences are considered.

### 5.1 Minimization of estimation covariances

The optimal subset of correspondences minimizes the uncertainty of the object pose and motion estimate. Thus, the most straightforward selection algorithm is to use equations (4), (6) and (8) for selecting the set that minimizes the *a posteriori* estimation covariance matrix $Q_{k|k}$.

In most cases, this approach is not feasible because of its combinatorial computational complexity. Consequently, it is most interesting as a reference.

### 5.2 Conditioning of measurement equations

To perform a good state estimate update, well conditioned, stable measurement equations are needed that minimize the sensitivity of the system state estimate to the errors in measurements. In [3] a Gauss-Markov estimator based technique is presented for this purpose. The selection of a correspondence set of size $n$ is based on maximizing the respective properly scaled and weighted determinant of $J_{F,n}^+$. The point set in use is updated one correspondence at a time by using a gradient search technique, so the computational cost grows linearly with the number of available points.

A method proposed in [4] produces almost identical results. The principle is to select a set that minimizes the condition of the Jacobian

$$cond(J_{F,n}) = ||J_{F,n}|| ||J_{F,n}^{-1}||$$

However, this method is expensive because of the needed matrix inversion and the combinatorial complexity of the selection task.

The computational cost can be made negligible by table look-up because the visible features at each pose can be predicted off-line by using the object model. However, the risk of failing to find all the expected features complicates this approach.

### 5.3 Random selection

The previous method tends to select the same points from frame to frame, if the object pose does not change enough to stimulate modifications. This may cause problems, if the selected sets are small, e.g., 1-2 points.

A solution is to force some random variations on the selected point sets. However, this is close to choosing the correspondences at random in the first place. In practice, it may be necessary to make the selection of certain important feature points more probable than of the others.

## 6. EXPERIMENTS

For comparing the different solutions, we have used simulated test runs in which the object approaches the camera. Table 1 shows "snapshots" of the object pose in the camera centered coordinate frame during a rather simple test run. The times $t_1-t_0 : t_2-t_1 : t_3-t_2$ spent in each part of run relate as 2:1:2.

Table 1: Object poses at the points of motion changes.

| sample time | $x$ (m) | $y$ (m) | $z$ (m) | $\alpha$ (deg) | $\beta$ (deg) | $\gamma$ (deg) |
|---|---|---|---|---|---|---|
| $t_0$ | -0.25 | -0.25 | 1.00 | 0.0 | 0.0 | 0.0 |
| $t_1$ | -0.20 | -0.20 | 0.75 | 10.0 | 10.0 | 10.0 |
| $t_2$ | -0.15 | -0.15 | 0.60 | 22.0 | 22.0 | 22.0 |
| $t_3$ | 0.00 | 0.00 | 0.25 | 60.0 | 60.0 | 60.0 |

In the beginning of the run, the $z$ distance from the camera to the object is 1m and reduces at a constant speed. Between $t_1$ and $t_2$ the motions of the object are accelerated and the run terminates at the $z$ distance of 0.25m from the camera at time $t_3$.

The real time length of the test run is assumed to be 10 seconds, so the rotational and translational accelerations during $[t_1, t_2]$ are approximately $3.5 deg/s^2$ and $12.5 mm/s^2$, respectively, and zero at all other times. The number of image frames in the test run is 75, corresponding to 133ms sampling intervals.

The test object used is an irregular polyhedron with 11 facets and 12 vertices that have been used as feature points. On average, nine feature points have been available from each image frame. The size of the test object is about $0.15 * 0.15 * 0.15 m^3$. The focal length of the camera is $10mm$.

### 6.1. Motion models

Table 2 shows the prediction errors when constant velocity, constant acceleration, variable noise and variable dimension motion models have been used. The error values correspond to the mean pixel distance on the image plane between predicted and actual projections of feature points. The feature points were selected using the minimum estimation covariance method.

The standard deviation of system noise used with the constant velocity and acceleration models was set at 5% of the maximum accelerations of the test run. This resulted in near minimum overall prediction errors.

With the variable noise model, the system noise was either 5% or 100% of the maximum accelerations. In addition, the system noise of the variable dimension model was changed from the normal 5% to 100% for three sample times after each dimensionality increase. The model changes were made when the pose estimation frame equivalent of the prediction error reached 2.5mm.

It is clear that the improvements in prediction error become progressively smaller when more feature points are added.

Table 2: Prediction errors with different motion models.

| points | constant velocity | constant acceleration | variable noise | variable dimension |
|---|---|---|---|---|
| 1 | 2.13 | 4.45 | 1.99 | 3.09 |
| 2 | 1.76 | 2.62 | 1.48 | 1.52 |
| 3 | 1.42 | 1.91 | 1.28 | 1.26 |
| 6 | 1.29 | 1.65 | 1.25 | 1.23 |

The constant acceleration model produces the largest errors, making it the least attractive approach with this test run. This is explained by the observability problem, because from 2 to 12 measured coordinate values (1 to 6 feature points) are used to estimate 18 state variables. The observability is significantly better for the 12 state variable constant velocity model as is demonstrated by the smaller errors of the other solutions that use this model most of the time.

The errors of variable noise and variable dimension models are very close to each other and slightly better than for the plain constant velocity method. The main difference is that the variable dimension method adapts faster to the velocity changes. This is demonstrated in Figure 2, where the thinnest and medium thick line represent the variable noise and variable dimension methods, respectively. The thickest line shows the error limit for changing the motion model.

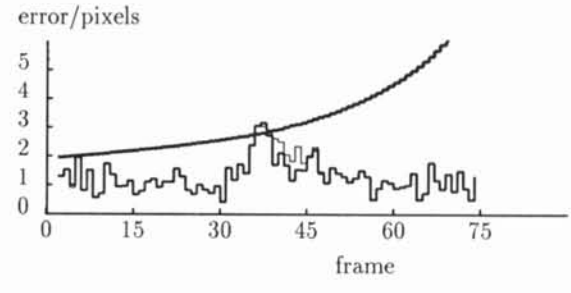


Figure 2: The prediction errors of variable noise and variable dimension methods. Three feature points are selected per frame.

The plots start at frame 2, because the first two are used for start-up. The acceleration period starts and ends at frames 29 and 44. After frame 36, where the motion models are modified, the error of the variable dimension method does not drop immediately, because two more observations are needed to determine the acceleration components in the state vector. The error of the variable noise scheme reduces more slowly.

With our test runs, higher dimensionality models did not result in any significant benefits over the variable noise method. In addition, continuous modification of the system noise level based on the prediction error, was not found to be noticeably better than the simple two level scheme.

### 6.2 Feature selection

The feature selection methods have been compared by using them with the variable system noise model based implementation. The resulting prediction errors are shown in Table 3.

The first column presents the errors that correspond to the minimum uncertainty criterion, $min(||Q_{k|k-1}||)$. These are the best achieved in our experiments. The condition of the Jacobian of the measurement function, $cond(J_F)$, gives slightly poorer results. In addition, when only one feature point was selected, it resulted in losing track of the object.

Random selection performs surprisingly well with our test object. When compared to the minimum uncertainty method, the penalty is almost negligible with one feature point selections.

Table 3: Prediction errors with different feature point selection methods.

| points | $\min(\|\|Q_{k\|k-1}\|\|)$ | $\text{cond}(J_F)$ | random |
|---|---|---|---|
| 1 | 1.99 | - | 2.05 |
| 2 | 1.48 | 1.89 | 1.75 |
| 3 | 1.28 | 1.43 | 1.59 |
| 6 | 1.25 | 1.27 | 1.48 |

## 7. SUMMARY

Our goal has been to select methods for estimating the relative pose and motion of a known object from sequences of image frames captured by a camera mounted on a robot arm. For this purpose, Kalman filtering is used to incorporate new measurements into existing estimates. However, careful modeling of object behavior and measurement uncertainties is needed.

The object motion is modeled relative to a moving coordinate frame, whose location cannot be accurately measured, resulting in increased uncertainties from unpredicted object behavior. These can be reduced by using higher frame rates, and in our experiments this has been the most rewarding direction of development.

Using fewer features cuts the computational delays, both at image analysis, and Kalman filtering. Most of the computations needed for selection can be performed off-line, so the cost of this task is insignificant. With the test object, even selecting the feature points at random works well.

The constant velocity motion model, with a simple two level variable noise scheme, is a computationally attractive solution and has produced good results in our experiments. When only a few features are used, the prediction errors with the same frame rate are smaller than with a more complex variable dimension method. The differences would be larger if the actual computational delays were taken into account.

In practice, it is useful to look for an optimum between the number of features, the quality of motion model, and the sampling rate that results in the lowest prediction errors.

## ACKNOWLEDGMENTS

The support of the the Computer Vision Laboratory of the University of Maryland, Defense Advanced Research Projects Agency and the Finnish Academy of Sciences is gratefully acknowledged.

## References

1. Broida, T. J., and Chellappa, R., *Estimation of Object Motion Parameters from Noisy Images*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-8, No. 1, pp. 90-99, 1986.

2. Dickmanns, E. D., and Graefe, V., *Dynamic Monocular Machine Vision*, Machine Vision and Applications, Vol. 1, pp. 223-240, 1988.

3. Dickmanns, E. D., and Graefe, V., *Applications of Dynamic Monocular Machine Vision*, Machine Vision and Applications, Vol. 1, pp. 241-261, 1988.

4. Feddema, J. T., Lee, C. S. G., and Mitchell, O. R., *Automatic Selection of Image Features for Visual Servoing of a Robot Manipulator*, IEEE International Conference on Robotics and Automation 1989, Vol. 2, pp. 832-837, 1989.

5. Gennery, D., *Tracking Known Three-Dimensional Objects*, Proc. 1st Nat. Conf. of American Association of Artificial Intelligence AAAI-82, pp. 13-17, 1982.

6. Linnainmaa, S., Harwood, D., and Davis, L. S., *Pose determination of a three dimensional object using triangle pairs*, CAR-TR-143, Center for Automation Research, University of Maryland, p. 47, 1985.

7. Matthies, L., and Shafer, S. A., *Error Modeling in Stereo Navigation*, IEEE Journal of Robotics and Automation, Vol. RA-3, pp. 239-248, 1987.

8. Pehkonen, K., Harwood, D., and Davis, L. S., *Parallel Calculation of 3-D Pose of a Known Object in a Single View*, CAR-TR-502, Center for Automation Research, University of Maryland, p. 17, 1990.

9. Wu, J. J., Rink, R. E., Caelli, T. M., and Gourishankar V. G., *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, 3, 373-394, 1989.