Appears in *Proc. Workshop on New Trends in Image-Based Robot Servoing* (in association with 1997 IEEE-RSJ International Conference on Intelligent Robots and Systems) Grenoble, France, pp. 53-63.

Characterizing the Uncertainties in Point Feature Motion for Model-Based Object Tracking*

Kevin Nickels   Seth Hutchinson
knickels@uiuc.edu   seth@uiuc.edu

Dept. of Electrical and Computer Engineering and The Beckman Institute
University of Illinois at Urbana-Champaign
Urbana, IL 61801

Abstract

We describe a model-based object tracking system that updates the configuration parameters of an object model based upon information gathered from a sequence of monocular images. We formulate the tracking problem as one of parameter estimation from partially observed data, and apply the Extended Kalman Filtering (EKF) algorithm. Kinematic object models and imaging models are used to determine what point feature movement reveals about the configuration parameters of the object. This information is used by the EKF to update estimates for parameters, and for the uncertainty in the current estimates, based on observations of point features in monocular images.

The main contribution of the paper is our characterization of uncertainties in the observed data. The image plane location of a point feature is interpreted as a random vector, and the sum-of-squared-differences correlation measure is used as a spatially uncertain measure of this vector. A probability density is derived for the location vector, and a covariance matrix is estimated from this density. An observation function gives the image location of a point feature for a given object configuration. The Jacobian of this function yields a measure of motion perceptibility for each parameter with respect to this feature. This measure determines the scaling of the uncertainty in the parameter estimate, when that estimate is used to predict a feature's image plane location. These two measures of uncertainty are compared, and used in the EKF, to combine the current parameter estimate with the new observation to yield an optimal estimate for the parameter given the new observation.

1 Introduction

In this paper we describe a model-based object tracking system that updates the configuration parameters (state) of an object model based upon information gathered from a sequence of monocular images. We formulate the tracking problem as one of parameter estimation from partially observed data, and then apply the Extended Kalman Filtering (EKF) algorithm to this problem.

In computer vision systems, the uncertainty in the state estimate will depend, among other things, on the relative configuration between the camera and the object. Using the notion of motion perceptibility [15] (also referred to as resolvability [12]), we account for this by incorporating the estimated state into the characterization of uncertainty in the estimate.

Additional uncertainty is introduced by the feature extraction process. To account for this, we incorporate knowledge from the feature matching into the uncertainty in state estimates. Our system uses the SSD method to detect features in the image [1]. We use the shape of the SSD surface to yield a measure of the spatial uncertainty of the measurement.

The remainder of the paper is organized as follows. In Section 2 we provide a brief overview of the Kalman filter and the Extended Kalman Filter. In Section 3 we describe how uncertainties due to configuration and feature match quality are incorporated into the EKF formalism. In Section 4 we present some preliminary experimental results.

2 Kalman Filtering

In this section, we present a brief overview of the classical Kalman Filter, and its extension to the Extended Kalman Filter. The reader familiar with Extended Kalman Filtering may wish to refer to (29) - (35) for the notation used in this paper.

2.1 The Kalman Filter

We can describe a linear system with noise by a discrete time *state model* of the following form:

$$\begin{aligned} \mathbf{x}_{k+1} &= A_k \mathbf{x}_k + B_k \mathbf{u}_k + G_k \mathbf{w}_k \\ \mathbf{z}_k &= H_k \mathbf{x}_k + D_k \mathbf{u}_k + \mathbf{v}_k, \end{aligned} \tag{1}$$

in which $\mathbf{x}_k \in \Re^n$, $\mathbf{u}_k \in \Re^m$, $\mathbf{z}_k \in \Re^q$, $\mathbf{w}_k \in \Re^t$, and $\mathbf{v}_k \in \Re^p$. Both the process noise, $\mathbf{w}_k$, and measurement noise, $\mathbf{v}_k$, are assumed to be sequences of zero-

*Partially supported by the National Science Foundation under Grants IRI 92–16428 and by EPRI under contract RP 8030–14

mean Gaussian white noise such that $Var(\mathbf{w}_k) = Q_k$[1] and $Var(\mathbf{v}_k) = R_k$ are positive definite matrices, and $E(\mathbf{w}_k \mathbf{v}_l^T) = 0$ for all k and l. For the majority of our work in this paper, the state will be the configuration of the robot. When a motion model is to be used, the state will consist of the configuration parameters and rate of change of those parameters.

We can decompose this linear system (1) into two disjoint systems, one composed of the deterministic elements and one composed of the stochastic elements. Then (1) is the sum of these two systems. The advantage to this approach is that the deterministic system has a well-known solution, which may be combined with the solution for the stochastic system to derive the Kalman Filter [3]. The stochastic system is given by

$$\begin{aligned} \mathbf{x}_{k+1}^s &= A_k \mathbf{x}_k^s + G_k \mathbf{w}_k \\ \mathbf{z}_k^s &= H_k \mathbf{x}_k^s + \mathbf{v}_k, \end{aligned} \tag{2}$$

and the deterministic system is given by

$$\begin{aligned} \mathbf{x}_{k+1}^d &= A_k \mathbf{x}_k^d + B_k \mathbf{u}_k \\ \mathbf{z}_k^d &= H_k \mathbf{x}_k^d + D_k \mathbf{u}_k, \end{aligned} \tag{3}$$

in which $\mathbf{x}_k = \mathbf{x}_k^s + \mathbf{x}_k^d$, and $\mathbf{z}_k = \mathbf{z}_k^s + \mathbf{z}_k^d$. To simplify the notation, we will omit the superscripts unless there is a possibility of confusion between the stochastic and deterministic systems.

Our task is to estimate the value of the state, $\mathbf{x}_k$, using only the observed data. Possibly the simplest way to perform this estimation is to ignore the system dynamics, and simply estimate $\mathbf{x}_k$ from the current observation of $\mathbf{z}_k$. This approach leads to standard mean-square estimation, where the estimate $\hat{\mathbf{x}}$ is chosen to minimize the expected value of the Euclidean norm squared of the error $E[(\mathbf{x} - \hat{\mathbf{x}})^T(\mathbf{x} - \hat{\mathbf{x}})]$. The well-known solution to this problem is given by

$$\hat{\mathbf{x}}_k = (H_k^T R_k^{-1} H_k)^{-1} H_k^T R_k^{-1} \mathbf{z}_k, \tag{4}$$

where $R_k = Var(\mathbf{v}_k)$.

We may extend this approach to incorporate all of the observed data by exploiting our knowledge of the system dynamics [3]. We define $\overline{\mathbf{z}}_k$ as

$$\overline{\mathbf{z}}_k = \begin{bmatrix} \mathbf{z}_0 \ldots \mathbf{z}_k \end{bmatrix}^T. \tag{5}$$

By fairly straightforward manipulations of (2) we obtain

$$\overline{\mathbf{z}}_j = C_{k,j} \mathbf{x}_k + \overline{\mathbf{e}}_{k,j} \tag{6}$$

in which

$$C_{k,j} = \begin{bmatrix} H_0 \Phi_{0,k} \\ \vdots \\ H_j \Phi_{j,k} \end{bmatrix}, \qquad \overline{\mathbf{e}}_{k,j} = \begin{bmatrix} \mathbf{e}_{k,0} \\ \vdots \\ \mathbf{e}_{k,j} \end{bmatrix}. \tag{7}$$

The transition matrices $\Phi_{j,k}$ are defined by

$$\Phi_{j,k} = \begin{cases} A_{j-1} A_{j-2} \cdots A_{k+1} A_k & \text{if } j > k \\ I & \text{if } j = k \\ \Phi_{k,j}^{-1} & \text{if } j < k, \end{cases} \tag{8}$$

and

$$\mathbf{e}_{k,l} = \mathbf{v}_l - H_l \sum_{i=l+1}^{k} \Phi_{l,i} G_{i-1} \mathbf{w}_{i-1}.$$

The form given in (6) is intuitively pleasing, as it a separates the nominal system operation (expressed as $C_{k,j}\mathbf{x}_k$) from the noise that has entered the system and been "fed through" the system dynamics (expressed as $\overline{\mathbf{e}}_{k,j}$).

We may now combine (6) with the previous result (4) to obtain the optimal estimate of the state in terms of the previously observed data $\overline{\mathbf{z}}_j$. We denote this estimate by $\hat{\mathbf{x}}_{k|j}$, and

$$\hat{\mathbf{x}}_{k|j} = (C_{k,j}^T W_{k,j} C_{k,j})^{-1} C_{k,j}^T W_{k,j} \overline{\mathbf{z}}_j. \tag{9}$$

in which $W_{k,j}^{-1} = Var(\overline{\mathbf{e}}_{k,j})$.

The Kalman Filter is obtained by deriving a recursive formulation of the optimal estimate given in (9). This recursive solution gives $\hat{\mathbf{x}}_{k|k}$ in terms of the previously computed quantities $\hat{\mathbf{x}}_{k|k-1}$ and $\hat{\mathbf{x}}_{k-1|k-1}$. We begin by deriving recursive formulations for the matrices $W_{k,j}^{-1}$, and $C_{k,j}$. These are given by

$$W_{k,j}^{-1} = Var(\overline{\mathbf{e}}_{k,j}) = \begin{bmatrix} W_{k,j-1}^{-1} & 0 \\ 0 & R_j^{-1} \end{bmatrix}, \tag{10}$$

$$C_{k,j} = \begin{bmatrix} C_{k,j-1} \\ H_j \end{bmatrix} \tag{11}$$

The derivation proceeds by expanding (9) in terms of these recursive forms. After some algebraic manipulations, we obtain

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + K_k(\mathbf{z}_k - H_k \hat{\mathbf{x}}_{k|k-1}), \tag{12}$$

in which

$$K_k = (C_{k,k-1}^T W_{k,k-1} C_{k,k-1} + H_k^T R_k^{-1} H_k)^{-1} H_k^T R_k^{-1}. \tag{13}$$

[1] The variance of a vector is simply the Covariance matrix of the vector with itself ($Var(\mathbf{v}) = Cov(\mathbf{v}, \mathbf{v})$).

The quantity K_k is known as the Kalman Gain, and (12) is the classic Kalman "prediction-correction" update equation.

Equation (12) expresses $\hat{\mathbf{x}}_{k|k}$, the estimate of the state at time k given data observations $\mathbf{z}_0 \ldots \mathbf{z}_k$, recursively in terms of quantities known at time k: $\hat{\mathbf{x}}_{k|k-1}$, $\mathbf{z}_0 \ldots \mathbf{z}_{k-1}$, K_k, and $\mathbf{z}_k$. With this recursive formulation, the entire observation history is summarized in these few matrices, and the memory requirements for Kalman Filtering over an extended period of time become feasible.

In order to use the update in (12), we need an expression for $\hat{\mathbf{x}}_{k|k-1}$ as a function of $\hat{\mathbf{x}}_{k-1|k-1}$. After some manipulation of the recursive form for $W_{k,k-1}^{-1}$, we obtain

$$\hat{\mathbf{x}}_{k|k-1} = A_{k-1}\hat{\mathbf{x}}_{k-1|k-1}. \tag{14}$$

This result is not particularly surprising; it merely states that our best prediction for the next state in terms of the estimate for the present state is obtained by using the system dynamics with no noise.

The final matrix for which we require a recursive solution is K_k, the Kalman Gain Matrix. From (13), (10), and (11) we have that

$$K_k = (C_{k,k}^T W_{k,k} C_{k,k})^{-1} H_k^T R_k^{-1}.$$

Defining $P_{k,j} = (C_{k,j}^T W_{k,j} C_{k,j})^{-1}$ and applying (11) and (10), we can see that

$$P_{k,k}^{-1} = P_{k,k-1}^{-1} + H_k^T R_k^{-1} H_k. \tag{15}$$

Using some facts from linear algebra (see, e.g., [5]) about inversion of block matrices, it can be shown that

$$P_{k,k} = P_{k,k-1} - P_{k,k-1} H_k^T (H_k P_{k,k-1} H_k^T + R_k)^{-1} H_k P_{k,k-1}.$$

These substitutions can be made in (13), giving us

$$K_k = P_{k,k-1} H_k^T (H_k P_{k,k-1} H_k^T + R_k)^{-1},$$

and the simpler form for $P_{k,k}$,

$$P_{k,k} = (I - K_k H_k) P_{k,k-1}.$$

Furthermore, it can be shown that

$$P_{k,k-1} = A_{k-1} P_{k-1,k-1} A_{k-1}^T + G_{k-1} Q_{k-1} G_{k-1}^T. \tag{16}$$

The matrix $P_{k,k-1}$ is the covariance matrix of the error,

$$\begin{aligned} P_{k,k-1} &= Var(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}) \\ P_{k,k} &= Var(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}) \end{aligned} \tag{17}$$

In particular, note that $P_{0,0} = Var(\mathbf{x}_0)$.

The recursive solution for the deterministic system given in (3) is given by [4]

$$\left\{ \begin{array}{l} \mathbf{x}_{k|k-1}^d = B_{k-1}\mathbf{u}_{k-1} \\ \mathbf{x}_{k|k}^d = \mathbf{x}_{k|k-1}^d \\ k = 1, 2, \ldots \end{array} \right. . \tag{18}$$

Thus, after unifying notation, it is easy to superimpose (12) - (16) and (18) to arrive at the classic Kalman filtering process:

$$\left\{ \begin{array}{l} P_{0,0} = Var(\mathbf{x}_0) \\ P_{k,k-1} = A_{k-1} P_{k-1,k-1} A_{k-1}^T + G_{k-1} Q_{k-1} G_{k-1}^T \\ K_k = P_{k,k-1} H_k^T (H_k P_{k,k-1} H_k^T + R_k)^{-1} \\ P_{k,k} = (I - K_k H_k) P_{k,k-1} \\ \hat{\mathbf{x}}_{0|0} = E(\mathbf{x}_0) \\ \hat{\mathbf{x}}_{k|k-1} = A_{k-1}\hat{\mathbf{x}}_{k-1|k-1} + B_{k-1}\mathbf{u}_{k-1} \\ \hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + K_k(\mathbf{z}_k - D_k\mathbf{u}_k - H_k\hat{\mathbf{x}}_{k|k-1}) \\ k = 1, 2, \ldots . \end{array} \right. \tag{19}$$

2.2 The Extended Kalman Filter

The problem of tracking an articulated object, such as a robot, is not a linear problem. Therefore the derivation above does not exactly capture the dynamics of the problem. Update equations for the exact nonlinear system dynamics can be derived (see [8] for an example), but they have been shown to be intractable. The Extended Kalman Filter (EKF) provides an approximation to the nonlinear system dynamics by linearizing them about the current state estimate.

Consider a nonlinear model of the form

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k) + G_k'(\mathbf{x}_k)\mathbf{w}_k \tag{20}$$

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{v}_k, \tag{21}$$

where $\mathbf{f}_k$ and $\mathbf{h}_k$ are vector valued functions with ranges of dimension n and q respectively, and G_k' is a matrix valued function of dimension $n \times p$. We make the usual assumptions with respect to the correlation of the noise and initial conditions:

$$\begin{aligned} E(\mathbf{w}_k\mathbf{w}_l^T) &= Q_k\delta_{kl} \\ E(\mathbf{v}_k\mathbf{v}_l^T) &= R_k\delta_{kl} \\ E(\mathbf{w}_k\mathbf{v}_l^T) = E(\mathbf{w}_k\mathbf{x}_0^T) &= E(\mathbf{v}_k\mathbf{x}_0^T) = 0. \end{aligned}$$

Consider the linear Taylor series approximation of $\mathbf{f}_k$ at $\hat{\mathbf{x}}_k$ and of $\mathbf{h}_k$ at $\hat{\mathbf{x}}_{k|k-1}$.

$$\begin{aligned} \mathbf{f}_k(\mathbf{x}_k) &\approx \mathbf{f}_k(\hat{\mathbf{x}}_k) + \left(\frac{\partial \mathbf{f}_k}{\partial \mathbf{x}_k}(\hat{\mathbf{x}}_k)\right)(\mathbf{x}_k - \hat{\mathbf{x}}_k) \\ \mathbf{h}_k(\mathbf{x}_k) &\approx \mathbf{h}_k(\hat{\mathbf{x}}_k) + \left(\frac{\partial \mathbf{h}_k}{\partial \mathbf{x}_k}(\hat{\mathbf{x}}_k)\right)(\mathbf{x}_k - \hat{\mathbf{x}}_k), \end{aligned} \tag{22}$$

where $\frac{\partial \mathbf{f}_k}{\partial \mathbf{x}_k}(\hat{\mathbf{x}}_k)$ and $\frac{\partial \mathbf{h}_k}{\partial \mathbf{x}_k}(\hat{\mathbf{x}}_k)$ are the Jacobians of $\mathbf{f}$ and $\mathbf{h}$, respectively, evaluated at $\hat{\mathbf{x}}_k$. That is,

$$\left[\frac{\partial \mathbf{h}}{\partial \mathbf{x}_k}(\mathbf{x}_k^*)\right] = \begin{bmatrix} \frac{\partial h_1}{\partial x_k^1}(\mathbf{x}_k^*) & \cdots & \frac{\partial h_1}{\partial x_k^n}(\mathbf{x}_k^*) \\ \vdots & \ddots & \vdots \\ \frac{\partial h_m}{\partial x_k^1}(\mathbf{x}_k^*) & \cdots & \frac{\partial h_m}{\partial x_k^n}(\mathbf{x}_k^*) \end{bmatrix} \tag{23}$$

and

$$\left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}_k}(\mathbf{x}_k^*)\right] = \begin{bmatrix} \frac{\partial f_1}{\partial x_k^1}(\mathbf{x}_k^*) & \cdots & \frac{\partial f_1}{\partial x_k^n}(\mathbf{x}_k^*) \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_k^1}(\mathbf{x}_k^*) & \cdots & \frac{\partial f_m}{\partial x_k^n}(\mathbf{x}_k^*) \end{bmatrix}.$$

This leads to the approximate linear state space description

$$\begin{aligned} \mathbf{x}_{k+1} &= A_k \mathbf{x}_k + \mathbf{u}_k + G_k \mathbf{w}_k \\ \mathbf{y}_k &= H_k \mathbf{x}_k + \mathbf{v}_k \end{aligned} \tag{24}$$

in which

$$A_k \approx \frac{\partial \mathbf{f}_k}{\partial \mathbf{x}_k}(\hat{\mathbf{x}}_k)$$

$$H_k \approx \frac{\partial \mathbf{h}_k}{\partial \mathbf{x}_k}(\hat{\mathbf{x}}_k) \tag{25}$$

$$\mathbf{u}_k \approx \mathbf{f}_k(\mathbf{x}_k) - A_k \hat{\mathbf{x}}_k \tag{26}$$

$$G_k \approx G_k'(\hat{\mathbf{x}}_k)$$

$$\mathbf{y}_k \approx \mathbf{z}_k - \mathbf{h}_k(\hat{\mathbf{x}}_{k|k-1}) + H_k \hat{\mathbf{x}}_{k|k-1}. \tag{27}$$

We choose the initial estimate $\hat{\mathbf{x}}_0 = E(\mathbf{x}_0)$ and predicted position $\hat{\mathbf{x}}_{1|0} = \mathbf{f}_0(\hat{\mathbf{x}}_0)$ to be consistent with this linear model.

Using these approximations and the results from Section 2.1, we arrive at the EKF update equation:

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + K_k(\mathbf{z}_k - \mathbf{h}_k(\hat{\mathbf{x}}_{k|k-1})) \tag{28}$$

Combining (28) with the previously derived standard Kalman Filter, we arrive at the following set of equations, generally collectively called the *Extended Kalman Filter*:

$$P_{0,0} = Var(\mathbf{x}_0) \tag{29}$$

$$\hat{\mathbf{x}}_0 = E(\mathbf{x}_0) \tag{30}$$

$$\begin{aligned} P_{k,k-1} = &\left[\frac{\partial \mathbf{f}_{k-1}}{\partial \mathbf{x}_{k-1}}(\hat{\mathbf{x}}_{k-1})\right] P_{k-1,k-1} \left[\frac{\partial \mathbf{f}_{k-1}}{\partial \mathbf{x}_{k-1}}(\hat{\mathbf{x}}_{k-1})\right]^T \\ &+ G_{k-1}(\hat{\mathbf{x}}_{k-1}) Q_{k-1} G_{k-1}^T(\hat{\mathbf{x}}_{k-1}) \end{aligned} \tag{31}$$

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{f}_{k-1}(\hat{\mathbf{x}}_{k-1}) \tag{32}$$

$$K_k = P_{k,k-1} \left[\frac{\partial \mathbf{h}_k}{\partial \mathbf{x}_k}(\hat{\mathbf{x}}_{k|k-1})\right]^T \times \tag{33}$$

$$\left[\left[\frac{\partial \mathbf{h}_k}{\partial \mathbf{x}_k}(\hat{\mathbf{x}}_{k|k-1})\right] P_{k,k-1} \left[\frac{\partial \mathbf{h}_k}{\partial \mathbf{x}_k}(\hat{\mathbf{x}}_{k|k-1})\right]^T + R_k\right]^{-1}$$

$$P_{k,k} = \left[I - K_k \left[\frac{\partial \mathbf{h}_k}{\partial \mathbf{x}_k}(\hat{\mathbf{x}}_{k|k-1})\right]\right] P_{k,k-1} \tag{34}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + K_k(\mathbf{z}_k - \mathbf{h}_k(\hat{\mathbf{x}}_{k|k-1})) \tag{35}$$

2.3 Robustness/Limitations of Kalman Filtering

The Extended Kalman Filter (EKF) is a very robust and flexible algorithm. This fact is evidenced by the wide use of the EKF in spite of the fact that many of the assumptions involved in the derivation of the filter are violated in most common situations. The noise covariance matrices Q and R are often known only to within an order of magnitude. The process and observation noise processes, being real, cannot be white. In spite of all these violations of the theoretical assumptions, the EKF gives reasonable results in a wide array of applications, including many applications very similar to ours.

Least squares techniques in general, and Kalman Filtering in particular, have some well-known limitations when applied to parameter estimation. The assumptions made by these techniques exclude gross errors (outliers), systematic errors and correlations in observations, and assume exact models [19]. Each of these assumptions is often violated to a certain extent in any given observation; these violations may prevent least-squares techniques from yielding acceptable results. Estimators have been developed to reduce or eliminate problems stemming from these types of errors [14], but the flexibility and overall robustness of the EKF, particularly when combined with a simple outlier reduction test (see, for example, [19]), lead us to use the EKF in favor of the more numerically robust methods [14].

3 Incorporating Uncertainty

In this section we describe how we exploit the EKF formalism to account for uncertainties due to relative camera object configuration and due to feature measurement. Throughout, we assume that the shape and appearance of the object being tracked are known (not an uncommon assumption [10]), which allows us to generate *templates* for the predicted appearance of features of interest in a given configuration.

3.1 Configuration Dependent Uncertainties

The predicted location of object features in the image as well as the ability of the computer vision system to perceive object motion are dependent on the configuration of the object. Thus, the weight that we give to an observation should also change as a function of the estimated configuration of the object.

At a given configuration of the object, we can compute the sensitivity of each object parameter to changes in the image plane location of each feature. This leads to a relationship that is similar what Sharma and Hutchinson term *motion perceptibility* [15]. The motion perceptibility ellipsoid for a feature is the ellipsoid traced out by the image of that point as the object parameters change by a unit magnitude. The scalar measure of motion perceptibility, defined to be the volume of this ellipse, relates the magnitude of the rate of change in an object's configuration parameters to the magnitude of the rate of change in the image of that object.

We relate the relationship described above to the update of the state variables. The observation in the image plane of highly perceptible motion provides accurate estimates of that motion in configuration space. Conversely, the observation of imperceptible motion provides inaccurate estimates of that motion. This idea is used by Sutanto and Sharma [18] to evaluate motion plans for visual servoing applications. Nelson and Khosla [12] have proposed resolvability, a similar measure, for the purpose of sensor placement analysis.

The confidence in a feature location prediction should depend on the motion perceptibility of the feature at the estimated configuration, and the uncertainty in that estimated configuration. It can be seen from (22) that the variance in a prediction is

$$\begin{aligned} W_k &= Var\left[\mathbf{h}_k(\mathbf{x}_k) - \mathbf{h}_k(\hat{\mathbf{x}}_k)\right] \\ &\approx Var\left[\left(\frac{\partial \mathbf{h}_k}{\partial \mathbf{x}_k}(\hat{\mathbf{x}}_k)\right)(\mathbf{x}_k - \hat{\mathbf{x}}_k)\right] \\ &= \left[\frac{\partial \mathbf{h}_k}{\partial \mathbf{x}_k}(\hat{\mathbf{x}}_k)\right] Var(\mathbf{x}_k - \hat{\mathbf{x}}_k) \left[\frac{\partial \mathbf{h}_k}{\partial \mathbf{x}_k}(\hat{\mathbf{x}}_k)\right]^T \\ &= H_k P_k H_k^T, \end{aligned}$$

where H_k is as given in (25), and P_k is as given in (17).

Note that this weighting is based solely on the estimated configuration and not on R_k, the quality of the feature extraction. The uncertainty involved in feature extraction is discussed in the next section. Both components relate to the final gain matrix K_k, as can be seen in the definition of that matrix given in (33).

3.2 Measurement Error

Feature detection can be accomplished by comparing an image region against a template for the feature, and using some metric, usually the Sum of Squared Differences (SSD) between the two regions, to rate the similarity. A good review of this technique is given in [2]. The main problem with this approach is that the simple template is a 2D entity, and the image patch may not undergo transformations that the template can model. A slightly more complex algorithm that also works in certain situations is to use an image patch from the previous image, taken from the area around the last computed position of the feature in that image, for the template. The main difficulty of this approach is feature skew, where the template slowly stops tracking the feature of interest and creeps onto another feature. The next step in sophistication is to have a template that can model more general transformations than the simple template. For example, there have been templates which can model a general affine transformation [6]. An even more sophisticated template might have a 3D registered texture as part of the template, and use the predicted position of the object, along with computer graphics techniques, to render the relevant portion of the scene, complete with sophisticated texture warping, and estimate the appearance of a feature in the image for use as a template to match against [9], [10]. In our work, we presently use the basic SSD approach.

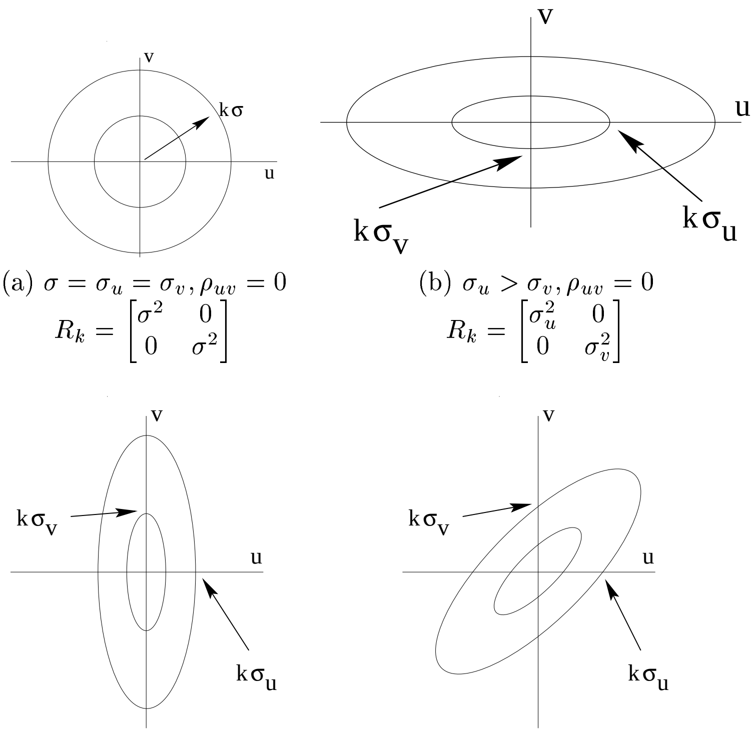
One advantage to SSD based feature detection is that in addition to an extracted feature location, some knowledge of the "goodness of fit" of that feature to the surrounding image is returned, in the form of an SSD surface [1]. The local shape of the SSD surface can tell in which directions the template is a good match with the local image content, and in which directions the template differs greatly from the local image content [13].

If we interpret the feature location as a two dimensional random vector, the surface returned by a SSD measurement can be normalized and treated as a probability measure on the geometric position of the feature. Since the matching properties of the SSD measurement are only valid within some region of the feature location in the image, this interpretation and normalization needs to be a local one. Away from the feature location, other features interfere with the matching process, and the match score no longer reflects the spatial uncertainty for the given feature.

The mode, or most probable value, of the random vector is the peak of the SSD surface. We take this as the feature location measurement, z_k. Assuming a symmetric unimodal random vector (for the purposes of the EKF, the random vector is assumed to be Gaussian), contours of equal probability surround the mode.

It has been noted [20] [11] that the covariance matrix of a random vector determines the shape and orientation of these contours (in the noiseless case, they are ellipsoids) of constant probability. Kosaka and Kak [7] provide an in-depth discussion on the equivalence between a covariance matrix and the related error ellipsoids. Figure 1 show contours of constant probability for several different situations, and the respective covariance matrices. The orientation of the semi-major axes of the error ellipsoids determine the eigenvectors of the covariance matrix. The lengths of the semi-major axes determine the eigenvalues of the covariance matrix [20].

The contours of constant probability can be estimated from the SSD surface. From this, the equations for the contours of constant likelihood in the (u, v)



(a) $\sigma = \sigma_u = \sigma_v, \rho_{uv} = 0$

$$R_k = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}$$

(b) $\sigma_u > \sigma_v, \rho_{uv} = 0$

$$R_k = \begin{bmatrix} \sigma_u^2 & 0 \\ 0 & \sigma_v^2 \end{bmatrix}$$

(c) $\sigma_u < \sigma_v, \rho_{uv} = 0$

$$R_k = \begin{bmatrix} \sigma_u^2 & 0 \\ 0 & \sigma_v^2 \end{bmatrix}$$

(d) $\sigma = \sigma_u = \sigma_v, \rho_{uv} \neq 0$

$$R_k = \begin{bmatrix} \sigma^2 & \rho_{uv}\sigma^2 \\ \rho_{uv}\sigma^2 & \sigma^2 \end{bmatrix}$$

Figure 1: Contours of constant probability

plane can be estimated,

$$\frac{(u-u_0)^2}{\sigma_u^2} + \frac{(v-v_0)^2}{\sigma_v^2} - \frac{2\rho_{uv}(u-u_0)(v-v_0)}{\sigma_u\sigma_v} = k$$

for any k, where u_0 and v_0 are the location of the mode in the (u, v) plane. The knowledge of the variance of u (σ_u^2), the variance of v (σ_v^2), and the correlation coefficient between u and v (ρ_{uv}) yields the desired covariance matrix,

$$R_k = \begin{bmatrix} \sigma_u^2 & \rho_{uv}\sigma_u\sigma_v \\ \rho_{uv}\sigma_u\sigma_v & \sigma_v^2 \end{bmatrix}. \tag{36}$$

In the following, we will describe the error in terms of the geometric interpretation given above (the shape and size of the contours of equal probability), as well as describing the resultant covariance matrix R_k.

To date, researchers have used the local shape of the SSD surface only as a confidence measure, and those researchers have only looked at the curvature along the image plane axes [1] or the image plane axes and 45° lines in the image plane [13]. More importantly, the curvature estimates are used only as a computation of the certainty of the match between the template and the local image structure, not as an uncertainty measurement on the location of the match.

Retaining only uncertainty measurements along image plane axes is equivalent to recording σ_u and σ_v. Thus, the implicit assumption made by these researchers is that the equations for the contours of equal probability are of the form shown in Figure 1(a)-(c). By observing ρ_{uv} as well, we propose to retain information about the orientation of the ellipsoids of constant probability, as well as their intersection with the u and v axes. Therefore, we gain the ability to maintain information about directions of good spatial discrimination even if that direction of that discrimination doesn't happen to be aligned with the (u, v) image plane axes.

3.3 Combination of Measurement Error with Configuration Dependant Uncertainties

In this section, we combine R_k given in (36) with W_k, derived in Section 3.1, to arrive at the final weighting matrix K_k, which determines the amount of influence observations have at each step when updating the state variables.

By incorporating the measurement of uncertainty R_k as a part of the feature observation at each step, the system gains the ability to give more weight to observations with low spatial uncertainty than those observations with high spatial uncertainty. Note that since this weighting matrix is computed at every step, feature observations with time-varying uncertainty are weighted appropriately. Confidence measures on feature measurements [13] [16] are typically used for feature selection before tracking progresses.

This use of time-varying measurement uncertainty allows robustness to external occlusion of features. When a feature is occluded, it is assumed that the template for that feature will not match the local image structure well. In this case, the peak will not be sharp, and any measurements returned under this condition will have high uncertainty associated with them. Note that since the features are not tracked as individual tokens, but are used to update a state model, a false match with an occluding surface will not distract the tracker.

This characterization of measurement uncertainty also allows the use of a single feature extraction method for different feature types. Often, special consideration must be made when tracking features of differing types [6]. By retaining information about the directional performance of the feature extraction, we propose to allow the system to use the same feature extraction method for point and line features, for example.

It has been shown [11] that the optimal combination of two measurements z_1 and z_2 of a Gaussian random variable with standard deviations σ_1 and σ_2 respectively, is to weight each according the the relative variance of the other. That is,

$$\hat{z} = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2} z_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} z_2.$$

We now consider combining the feature location prediction $h_k(\hat{x}_k)$, with the new feature location observation z_k, to arrive at an optimal updated estimate for the state, $\hat{x}_k$. The variance of the prediction W_k, has been shown to be $H_k P_k H_k$, where H_k is given by (25). The variance of the measurement has been directly estimated to be R_k. Therefore, we have that

$$\begin{aligned}
h_k(\hat{x}_k) &= H_k P_k H_k^T (H_k P_k H_k^T + R_k)^{-1} z_k + \\
&\qquad R_k (H_k P_k H_k^T + R_k)^{-1} h_k(\hat{x}_{k|k-1}) \\
h_k(\hat{x}_{k|k-1}) + H_k(\hat{x}_k - \hat{x}_{k|k-1}) &= \\
&\qquad H_k K_k z_k + (1 - H_k K_k) h_k(\hat{x}_{k|k-1}) \\
H_k \hat{x}_k &= H_k \hat{x}_{k|k-1} + H_k K_k z_k - H_k K_k h_k(\hat{x}_{k|k-1}) \\
\hat{x}_k &= \hat{x}_{k|k-1} + K_k (z_k - h_k(\hat{x}_{k|k-1})),
\end{aligned}$$

where K_k is given by (33). Therefore, our state estimate update can be computed by the Extended Kalman Filter given in (29) - (35). We propose to achieve the desired system operation, therefor, through careful selection of the system functions, and the weighting equations for the EKF need not be modified for our purpose.

4 Preliminary Results

In this section, we will present some preliminary results for the case of a one link arm being observed by a 1D sensor. In order to concentrate on the effect that the object kinematics have on the tracking algorithm, no motion model is used in the first two cases. A constant velocity motion model is used for the object parameters in the third case.

Finally, we present an illustration of the current method for extracting a feature measurement z_k and related uncertainty R_k from SSD correlation surfaces.

In the first example, illustrated in Figure 2, there is a planar one link arm that can rotate about one end, which is fixed at the origin. Thus, configuration space is one dimensional. We assume that the robot is embedded in a two dimensional workspace, and there is an orthogonal projection onto the one dimensional sensor plane (to the right of the robot, sufficiently far away that there is no interaction between the arm and the sensor plane, at least a distance a_1).

Using the notation outlined in Section 2.2, the (scalar) filter for this problem is as follows. This system has one parameter, so the state vector has one entry,

$$x_k = \theta. \tag{37}$$

There is no motion model incorporated into this system. The prediction for the location of the arm at the next time instant is the best estimate for the location of the arm at the current time. This is done for simplicity of the example. Systems with motion models will be explored below. In this situation, the prediction equations are

$$\begin{aligned}
\mathbf{f}_k(x_k) &= x_k \\
\frac{\partial f_k}{\partial x_k} &= 1.
\end{aligned} \tag{38}$$

Using orthogonal projection, the observation equations are

$$\begin{aligned}
h_k(x_k) &= a_1 cos(x_k) \\
\frac{\partial h_k}{\partial x_k} &= -a_1 sin(x_k)
\end{aligned} \tag{39}$$

We assume for simplicity that the process noise does not vary with the state,

$$G_k'(x_k) \equiv 1.$$

Under these assumptions, the filter equations are as follows. Recall that Q_k is the covariance matrix for

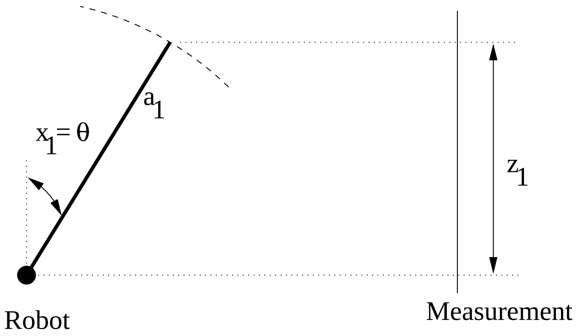


Figure 2: 1 Degree of Freedom Example.

the system noise, and R_k is the covariance matrix for the observation noise. In this case, both Q_k and R_k are scalar.

$$P_{k,k-1} = P_{k-1,k-1} + Q_k \tag{40}$$

$$\hat{x}_{k|k-1} = \hat{x}_{k-1|k-1} \tag{41}$$

$$K_k = P_{k,k-1} \frac{-a_1 sin(\hat{x}_{k-1})}{a_1^2 sin^2(\hat{x}_{k-1}) P_{k,k-1} + R_k} \tag{42}$$

$$P_{k,k} = P_{k,k-1} \left(1 + K_k a_1 sin(\hat{x}_{k-1})\right) \tag{43}$$

To illustrate the usefulness of an object model for tracking, it is helpful to observe the response of the filter components to changes in the configuration of the robot. When $\theta = -\pi/2$, the arm is pointing directly at the sensor plane, and changes in x are easily observable from changes in z. This effect can be seen in Figure 3(b), where the ratio $P_{k,k}/P_{k,k-1}$ (intuitively, the reduction in uncertainty due to a single measurement) is small near $\theta = -\pi/2$. As $\theta \to 0$, changes in x do not cause any changes in z. This fact, gleaned from our knowledge of the kinematics of the robot and the geometry of the sensor with respect to the robot, is reflected in the measurement update portions of the filter (33) - (35), and is illustrated by a lack of decrease in the uncertainty during the update (Figure 3(b)) near $\theta = 0$. That is, the measurement doesn't help locate the target very well, as is to be expected near this configuration. In this region, the feature measurement is weighted heavily with respect to the a priori state estimate, as can be seen from Figure 3(a). In this figure, a value of 0 denotes complete dependence on the feature measurement when computing the next state estimate. Conversely, a value of 1 denotes complete disregard for the feature measurement when computing the state estimate update.

Another dimension in Figure 3 that is interesting to observe is that of R_k, the variance of the measurement noise. In most configurations, increasing the measurement noise variance increases the multiplicative filter variance coefficient, $P_{k,k}/P_{k,k-1}$ (equivalently, reduces the amount of uncertainty reduction), as would be expected. In Figure 3(a), we can see that low measurement noise causes the measurements to become weighted rather heavily with respect to the a priori state estimate, as one would expect.

Note that the analysis above is for a single observation. Multiple observations, say at times k_1 and k_2, would have different uncertainties associated with them, R_{k_1} and R_{k_2}. This allows the framework to weight different observations of the same feature differently, as is appropriate for their respective R_k.

To summarize the changes that occur within the filter over time as the configuration of the robot changes, we look at Figure 4. We see that as the robot nears the singular positions at $-\pi$ and 0, the variance (uncertainty) rises and the weighting favors the previous state estimate over the observations. As the robot passes into the observable region (informally, the region not near a singularity), the gain lowers again, lending more credence to the observations in that region. Note that it takes a few observations in the non-singular region to bring the variance down to steady-state.

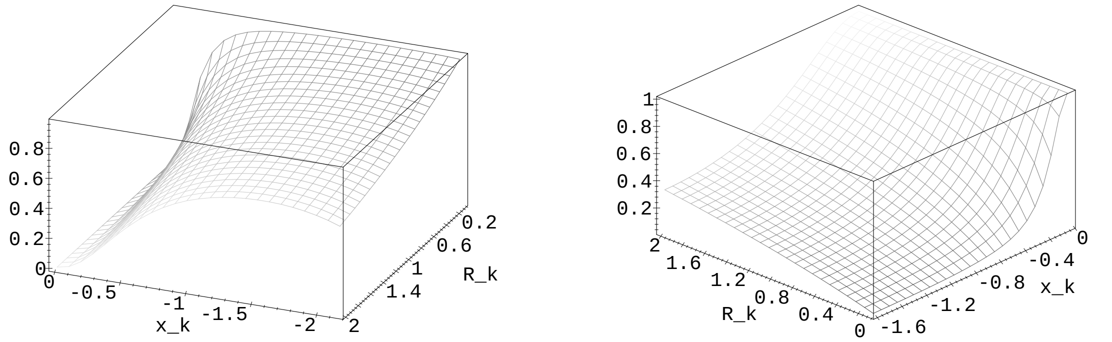
Although the above example is simple and easy to understand, some simplifications need to be dropped in order to more closely approximate our experimental setup. The end effector offset is nonzero in the x, y, and z directions in the robots coordinate system (in the above example, the y and z offsets are 0,) and the camera is sufficiently close to the arm to warrant the use of a perspective projection model.

Under these assumptions, the motion model does not change, but the observation equations become

$$h_k(x_k) = \frac{f(zsin(x_k) - xcos(x_k))}{d_c - zcos(x_k) - xsin(x_k)} \tag{44}$$

$$\frac{\partial h_k}{\partial x_k} = \frac{f(zcos(x_k) + xsin(x_k))}{d_c - zcos(x_k) - xsin(x_k)} - \frac{f(zsin(x_k) - xcos(x_k))^2}{(d_c - zcos(x_k) - xsin(x_k))^2} \tag{45}$$

where f is the focal length of the camera, and d_c is the



(a) A priori estimate weighting coefficient

(b) $P_{k,k}/P_{k,k-1}$

Figure 3: Changes in Filter Components with Configuration Changes.

distance of the camera from the workspace origin.

The qualitative behavior of this filter, shown in Figure 5 is not appreciably different from that presented above. The most noticeable difference is the shift, due to the perspective projection model and link offsets, of the observation singularity from exactly zero to somewhat before zero.

We now present the changes to the framework presented above required to utilize a motion model. In this example, we estimate the value and first time derivative of θ. Hereafter, superscripts will be used to denote vector or matrix indices, and subscripts will continue to be used to denote time indices.

$$\mathbf{x}_k = \begin{bmatrix} x_k^1 \\ x_k^2 \end{bmatrix} = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix}.$$

With a motion model, the estimate at time k (38) is no longer the previous estimate. Now, we can adjust by the estimated velocity (multiplied by the elapsed time Δ), to give a better extrapolated estimate

$$\mathbf{f}_k(\mathbf{x}_k) = \begin{bmatrix} x_k^1 + \Delta x_k^2 \\ x_k^2 \end{bmatrix}$$
$$\frac{\partial \mathbf{f}_k}{\partial \mathbf{x}_k} = \begin{bmatrix} 1 & \Delta \\ 0 & 1 \end{bmatrix}.$$

In the observation function, we preserve most of (39), adding a row to note that we cannot observe the velocity.

$$\mathbf{h}_k(\mathbf{x}_k) = \begin{bmatrix} a_1 cos(x_k) \\ 0 \end{bmatrix}$$
$$\frac{\partial \mathbf{h}_k}{\partial \mathbf{x}_k}(\mathbf{x}_k) = \begin{bmatrix} -a_1 sin(x_k) \\ 0 \end{bmatrix}$$

We again assume for simplicity that the process noise does not vary with the state,

$$G'_k(\mathbf{x}_k) \equiv I.$$

Under these assumptions, the filter equations are as follows

$$P_{k,k-1} = \begin{bmatrix} 1 & \Delta \\ 0 & 1 \end{bmatrix} P_{k-1,k-1} \begin{bmatrix} 1 & 0 \\ \Delta & 1 \end{bmatrix} + Q_k \tag{46}$$

$$= \begin{bmatrix} P^{1,1}_{k-1,k-1} + 2\Delta P^{1,2}_{k-1,k-1} + \Delta^2 P^{2,2}_{k-1,k-1} & P^{1,2}_{k-1,k-1} + \Delta P^{2,2}_{k-1,k-1} \\ P^{1,2}_{k-1,k-1} + \Delta P^{2,2}_{k-1,k-1} & P^{2,2}_{k-1,k-1} \end{bmatrix} + Q_k \tag{47}$$

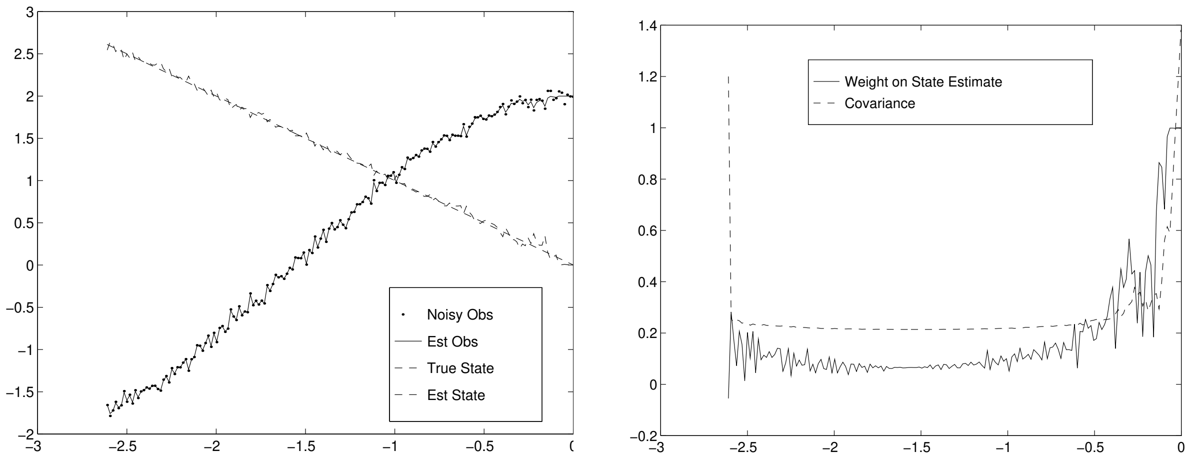
$$\hat{\mathbf{x}}_{k|k-1} = \begin{bmatrix} \hat{x}^1_{k-1|k-1} + \hat{x}^2_{k-1|k-1}\Delta \\ \hat{x}^2_{k-1|k-1} \end{bmatrix} \tag{48}$$

$$K_k = \frac{a_1 cos(\hat{\mathbf{x}}_{k-1})}{a_1^2 cos^2(\hat{\mathbf{x}}_{k-1}) P^{1,1}_{k,k-1} + R_k} \begin{bmatrix} P^{1,1}_{k,k-1} \\ P^{1,2}_{k,k-1} \end{bmatrix} \tag{49}$$

$$P_{k,k} = \begin{bmatrix} 1 - a_1 cos(K_k^1) & 0 \\ -a_1 cos(K_k^2) & 1 \end{bmatrix} P_{k,k-1} \tag{50}$$

Note that the above equations are essentially matrix versions of (40) - (43), with additional terms involving increased uncertainty, and increased predictive power, due to elapsed time and the motion model. Also of note is the gain matrix K_k in (49), which updates the velocity estimate based upon a fraction of the prediction correction.

In our experimental setup, we observe the position of the end effector of the robot with a CCD camera. The search area is short and wide, yielding an essentially one dimensional measurement. The two dimensional



(a) Observed (Noisy) and Predicted (Filter) Observations and State

(b) Filter Variance and Gain

Figure 4: Changes in Filter Components with Configuration Changes.

templates used to locate the end effector in the image are constructed from images of the end effector taken at known joint angles. An example of the correlation surface generated by this search is given in Figure 6.

The variance associated with the measurement is computed by converting the SSD surface via an exponential function as in [17], interpreting this response distribution as a probability density on the "true" location of the feature, and numerically computing the variance from this density.

5 Conclusions

We have described how to characterize the uncertainty in data observations in terms of both the perceptibility of feature motion and the quality of feature extraction. We have incorporated these characterizations into an EKF formalism, and presented preliminary results for the simple case of 1D tracking. We are currently extending our results to a variety of more complicated problems, addressing such issues as occlusion, higher dimensional systems, and tracking more complicated (possibly non-rigid) objects.

References

[1] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2:283–310, 1989.

[2] R. Brunelli and T. Poggio. Face recognition: Features versus templates. *IEEE Transactions on pattern alanlysis and machine intelligence*, pages 1042–1052, October 1993.

[3] C. K. Chui and G. Chen. *Kalman Filtering with Real-Time Applications.* Springer Series in Information Sciences. Springer-Verlag, Berlin Heidelberg, 1987.

[4] C. K. Chui and G. Chen. *Linear Systems and Optimal Control.* Springer-Verlag, New York, NY, 1989.

[5] J. L. Goldberg. *Matrix Theory with Applications.* International Series in Pure and Applied Mathematics. McGraw-Hill, 1991.

[6] G. Hager. Real-time feature tracking and projective invariance as a basis for hand-eye coordination. *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 533–539, 1994.

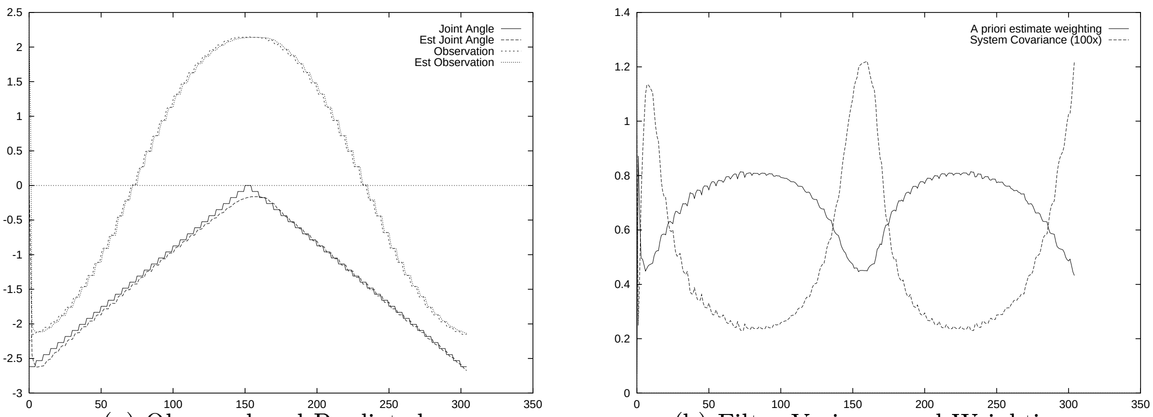
[7] A. Kosaka and A. C. Kak. Fast vision-guided robot navigation using model-based reasoning and prediction of uncertainties. *Computer Vision and Image Understanding*, pages 271–329, November 1992.

[8] F. L. Lewis. *Optimal Estimation with an introduction to stochastic control theory.* John Wiley & Sons, New York, NY, 1986.

[9] H. Li, P. Roivainen, and Forcheimer. 3-D motion estimation in model-based facial image coding. *IEEE Trans. Pattern Analysis and Machine Intelligence*, pages 545–555, June 1993.

[10] R. Lopez, A. Colmenarez, and T. S. Huang. Vision-based head and facial feature tracking. In *Advanced Displays and Interactive Displays Federated Laboratory Consortium, Annual Symposium.* Advanced Displays and Interactive Displays Federated Laboratory Consortium, January 1997.

[11] Peter S. Maybeck. *Stochastic Models, Estimation, and Control*, volume 141-1 of *Mathematics in Science and Engineering.* Academic Press, Inc, 1979.



(a) Observed and Predicted Observations and State

(b) Filter Variance and Weighting of State Estimate

Figure 5: Changes in Filter Components with Configuration Changes.

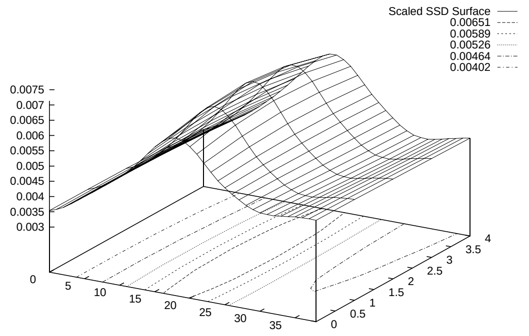


Figure 6: SSD Surfaces corresponding to end effector template.

[12] B. J. Nelson and P. K. Khosla. The resolvability ellipsoid for visual servoing. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 829–832, 1994.

[13] N. P. Papanikolopoulos. Selection of features and evaluation of visual measurements during robotic visual servoing tasks. *Journal of intelligent and robotic systems*, pages 279–304, July 1995.

[14] W. J. Rey. *Introduction to Robust and Quasi-Robust Statistical Methods*. Springer, Berlin, Heidelberg, 1983.

[15] Rajeev Sharma and Seth Hutchinson. Motion perceptibility and its application to active vision-based servo control. *IEEE Int. Conf. Robotics Automation*, 1997.

[16] J. Shi and C. Tomisito. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, June 1994.

[17] A. Singh and P. Allen. Image flow computation: An estimation-theoretic framework and a unified perspective. *Computer Vision and Image Understanding*, pages 152–177, September 1992.

[18] Herry Sutanto and Rajeev Sharma. Global performance measure of image features for visual servo control. *Journal of Robotic Systems*, 13(4):243–256, 1996.

[19] Z. Zhang and O. Faugeras. *3D Dynamic Scene Analysis*. Springer Series in Information Sciences. Springer-Verlag, Berlin Heidelberg, 1992.

[20] Ali Zolghadri. An algorithm for real-time failure detection in kalman filters. *IEEE Trans. on Automatic Control*, 41(10), October 1996.