# Estimating Uncertainty in SSD-Based Feature Tracking

Kevin Nickels [a,*], Seth Hutchinson [b]

[a] *Dept. of Engineering Science, Trinity University, San Antonio TX 78212, USA*

[b] *Dept. of Electrical and Computer Engineering and The Beckman Institute, University of Illinois at Urbana-Champaign, Urbana IL 61801, USA*

**Abstract**

SSD-based feature trackers have enjoyed growing popularity in recent years, particularly in the field of visual servo control of robotic manipulators. These trackers use sum-of-squared-differences correlation measures to locate target features in sequences of images. The results can then be used to estimate the motion of objects in the scene, to infer the 3D structure of the scene, or to control robot motions.

The reliability of the information provided by these trackers can be degraded by a variety of factors, including changes in illumination, poor image contrast, occlusion of features, or unmodeled changes in objects. This has led other researchers to develop confidence measures that are used to either accept or reject individual features that are located by the tracker. In this paper, we derive quantitative measures for the spatial uncertainty of the results provided by SSD-based feature trackers. Unlike previous confidence measures that have been used only to accept or reject hypotheses, our new measure allows the uncertainty associated with a feature to be used to weight its influence on the overall tracking process. Specifically, we scale the SSD correlation surface, fit a Gaussian distribution to this surface, and use this distribution to estimate values for a covariance matrix. We illustrate the efficacy of these measures by showing the performance of an example object tracking system with and without the measures.

*Key words:* Feature Tracking, Sum of Squared Differences, Uncertainty Estimation

* Corresponding author.

*Email addresses:* knickels@trinity.edu (Kevin Nickels), seth@uiuc.edu (Seth Hutchinson).

*Article published in Image and Vision Computing 20 (2002) 47–68*

## 1 Introduction

A growing number of applications in robotics and computer vision rely on real-time tracking of features in image sequences. These applications include estimating the motion of autonomous mobile vehicles, inferring the 3D structure of an environment from 2D image data, and controlling robotic manipulators. The reliability of feature trackers can be degraded by many factors, e.g., changes in illumination or poor image contrast. Nevertheless, many feature trackers treat all features in a uniform manner, without considering in any way the uncertainty associated with the feature tracking process. In this paper, we present a formalism by which the uncertainty associated with individual features can be assessed and subsequently used in the tracking process. This allows features to be accorded a level of confidence commensurate with the uncertainty in their measurement, which enables applications to weight features appropriately, depending on the associated confidence level and on the application's needs.

In this paper, we consider only the case of feature trackers that use the sum-of-squared-differences (SSD) correlation method [1] [2] [3]. In SSD-based feature tracking, a *feature template* is compared to portions of an image to locate that feature. A similarity metric is used to rate the similarity of the template and the image patch. The image region found to be the most similar to the template is typically taken to be the location of the feature. In our work, we consider features that are defined as given greyscale patterns in an image.

We are specifically concerned here with the *uncertainty* in the computation of the (2D) location of features in an image. These feature uncertainties can then be used by higher level application software. For example, in related work, we have used the methods presented in this paper to drive an uncertainty estimation process for an object tracking system that also takes into account kinematic and imaging uncertainties in tracking complex articulated objects [4].

There have been several other attempts to incorporate feature uncertainty into the tracking process. In the work described by Gennery [5] and Lowe [6] object tracking is performed by tracking salient features on the image of an object, and this information is used to compute the object motion that gave rise to the observed feature motion. Gennery tracks estimated error in the position, orientation, velocity, and angular velocity of the rigid object to aid in prediction of object motion. He uses naturally occurring edges on the polyhedral object as features. Lowe distinguishes between two different types of errors, matching errors and measurement errors, and attempts to utilize separate mechanisms to deal with each. Matching errors, mismatches between feature points on the object model and feature points in the image, are dealt with by removing outliers. Measurement errors are dealt with by using the variance in the object location, orientation, and configuration to compute the expected variance in measurements. In [3], estimates from regions of high

confidence are used to improve estimates in regions of low confidence. To facilitate this, a metric for the confidence in a motion estimate is developed. Finally, feature tracking confidence measures have been used in visual servo control to increase the robustness of the control [2]. In each of these cases, the uncertainty characterizations are used only to reject or accept features. In contrast, the approach presented here refers to the certainty of the location of the feature rather than the certainty in the measurement of that feature. It gives a quantitative evaluation of uncertainty that can be used to weight feature measurements in proportion to their reliability in various directions in the image.

In our research, we assume that the shape and appearance of the object being tracked are known. This is a particular version of the model-based tracking problem, which is of current interest in both the robotics and computer vision communities (see, e.g. [7], [8], [9]). By exploiting the information contained in shape and appearance models, we are able to generate templates for the predicted appearance of features of interest in a given configuration. An alternative to this model-based approach is to use features gathered on-line [2], which may help to ensure the quality of the features tracked, since feature templates exactly match previous feature appearance. However, this method neglects any *a priori* information about the geometric relationship of the individual features to the object.

The remainder of the paper is organized as follows. We begin with a review of the SSD-based feature tracking method. Following this, we describe our goals with respect to characterizing the spatial discrimination of features. We illustrate how these goals allow a system to account for both occlusion of features and suboptimal feature performance during object tracking. Then we present a Gaussian approximation and describe how sufficient statistics can be used to characterize this approximation. Finally, we present some feature tracking results taken from our implemented tracking system, illustrating the information gained from these error estimates and show a case study illustrating the degradation in our system when uncertainty estimation information is ignored.

## 2 Correlation and feature templates

The following sections discuss the three areas that are most crucial to the performance of correlation based tracking: the content of the feature templates, the definition and use of a specific similarity metric for tracking, and the definition of confidence measures on the tracking results.

### *2.1 Feature template generation*

The content of the template is an important choice in feature tracking. If the template faithfully reproduces the actual appearance of the feature in the image, tracking will work well. However, if a template is oversimplified or does not match the appearance of a feature in the image due to unmodeled effects, feature tracking will almost certainly perform poorly.

A template could be generated from a canonical view of the feature, and template matching done in a search window centered about the predicted position of the image. Brunelli and Poggio give a good review of this technique in the context of facial feature tracking [10]. The main problem with this straightforward approach is that the simple template is a 2D entity, and the image patch may undergo transformations that the template cannot model, such as rotation, shear, and changes in illumination [11].

A 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. Hager [12] uses this approach for visual servoing. Hager and Belhumeur [11] have also used previous tracking information to warp this image patch before use as a feature template to account for illumination and geometric changes, which increases the flexibility of this approach. The main difficulty with this approach is *feature drift*. Feature drift occurs when the computed position of the feature is slightly incorrect. This causes the next template to be slightly offset from the true feature location, and the next computed position of the feature to be slightly more incorrect. Slowly, the computed position migrates off the feature of interest, and the template contains image structure divergent from the feature of interest.

If object and scene modeling are part of the tracking framework, it is possible to create templates soley from this information, neglecting image content from previous images. Lopez et al. [9] have a 3D registered texture of a face as part of their object model. Computer graphics (CG) techniques are used to texture-map the texture onto a wire-frame model of the face to estimate the appearance of a feature in the image. This image patch is then used as a template in the feature tracking portion of the system.

Our work uses 3D models for complex articulated objects, also in a CG based framework (specifically, OpenGL [13]), to generate feature templates. Imaging and object models are used to produce a CG image of the scene in the estimated configuration. Since the CG scene is completely known the 2D image locations of salient features on the CG object are known. Portions of this image are then used as feature templates to compare against the input image. More information on our method for feature template generation is given in [14].

### *2.2 The SSD similarity metric*

In correlation-based tracking, a similarity metric is used to compare the feature template described above to areas of the image to locate the feature in the image.

The standard sum-of-squared-differences (SSD) metric for grayscale images is defined as:

$$SSD(u,v) = \sum_{m,n \in N} [T(m,n) - I(u+m, v+n)]^2, \quad (1)$$

where $T$ is the template image and $I$ is the input image. The location $(u,v)$ represents some location in the input image whose content is being compared to the content of the template. Papanikolopoulos [2] uses the SSD measure to generate tracking results that are then used for robotic visual servoing experiments. Anandan [3] and Singh and Allen [15] use this SSD metric for the computation of image flow. Alternative dissimilarity measures can be found in [16], [17], [18] and [19].

### *2.3 Windowing*

Often, the SSD measure is not computed for the entire input image, but only for some *search window* in the input image. Primarily for computational reasons, this restriction also serves as a focus of attention for the feature tracking algorithm. Singh and Allen [20] [15] define a fixed size square search window surrounding the previous location of the feature. Kosaka and Kak [21] consider at length the shape and location of the search window. They model the scene and compute a spatial probability density function for the location of each feature, then search the image area corresponding to 85% of the probability mass.

Our work uses a constant-velocity model for an articulated object to predict 3D positions for relevant points on the object. Imaging models are then used to project these locations to points on the image plane. A fixed size rectangular search window centered at these locations is established in the input image. See [22] for more details on the models used in this work.

## 3 Confidence measures and spatial uncertainty

It has been noted [3] that popular similarity measures often lead to some unreliable matches, particularly in image regions with little textural information. For this reason, it is often helpful to compute a *confidence* on the match found, as well as a location. This confidence measure typically gives information regarding the reliability of the match score. This scalar score often is used to estimate the reliability of

the feature, e.g., for use in later tracking operations or to propagate image flow information from one portion of an image to another [15]. Below, we will describe a matrix-valued measure that contains information both about the overall confidence in a feature measurement and information about how accurate the measurement is in all image directions.

Anandan [3] used the SSD matching scores of a template with a $5 \times 5$ image region to develop a match confidence measure based on the variation of the SSD values over the set of candidate matches. Anandan argued that if the variation of the SSD measure along a particular line in the search area surrounding the best match is small, then the component of the displacement along the direction of that line cannot be uniquely determined. Conversely, if there is significant variation along a given line in the search area, the displacement along this line is more likely correct.

Singh and Allen define a *response distribution* based on the SSD metric (1) as

$$\mathscr{R}D_c(u,v) = exp(-kSSD(u,v)), \tag{2}$$

where $k$ is used as a normalization factor. The normalization factor $k$ was chosen in [15] so that the maximum response was 0.95. Singh and Allen then argue that each point in the search area is a candidate for the "true match." However, a point with a small response is less likely to be the true match than a point with a high response. Thus, the response distribution could be interpreted as a probability distribution on the true match location – the response at a point depicting the likelihood of the corresponding match being the true match. This interpretation of the response distribution allows the use of estimation-theoretic techniques.

Under the assumption of additive zero mean independent errors, a covariance matrix $\mathbf{P}_m$ is associated with each location estimate. This matrix is constructed from information about the *shape* of (2) as $u$ and $v$ (the horizontal and vertical pixel locations) change.

$$\mathbf{P}_m = \begin{bmatrix} \dfrac{\sum\limits_{u,v\in N} \mathscr{R}D(u,v)(u-u_m)^2}{\sum\limits_{u,v\in N} \mathscr{R}D(u,v)} & \dfrac{\sum\limits_{u,v\in N} \mathscr{R}D(u,v)(u-u_m)(v-v_m)}{\sum\limits_{u,v\in N} \mathscr{R}D(u,v)} \\ \dfrac{\sum\limits_{u,v\in N} \mathscr{R}D(u,v)(u-u_m)(v-v_m)}{\sum\limits_{u,v\in N} \mathscr{R}D(u,v)} & \dfrac{\sum\limits_{u,v\in N} \mathscr{R}D(u,v)(v-v_m)^2}{\sum\limits_{u,v\in N} \mathscr{R}D(u,v)} \end{bmatrix} \tag{3}$$

where $u_m$ and $v_m$ are the estimated locations, in the $u$ and $v$ directions, of the feature and $N$ is the neighborhood of the pixel. Expressions for the normalized variance in the horizontal and vertical directions appear along the diagonal of (3), and an expression for the normalized covariance appears in both off-diagonal elements of $\mathbf{P}_m$. See [15] for more in-depth discussion of this formulation. The reciprocals of the eigenvalues of the covariance matrix are used as confidence measures associated

with the estimate, along the directions given by the corresponding eigenvectors. To our knowledge, Singh and Allen are the first researchers to treat the location of the best match as a random vector and to use the (normalized) SSD surface (SSDS) to compute the spatial certainty of the estimate of this vector. These confidence measures are then used in the propagation of high confidence measurements for local image flow to regions with lower confidence measurements, such as those caused by large homogeneous regions.

As the SSD measure is used to compare the template to areas of the image near the area generating the minimum SSD score, some measure of the *spatial discrimination* power of the template can be generated [3]. Spatial discrimination is defined as the ability to detect feature motion along a given direction in the image. This concept is quite similar to the confidence measures discussed above that estimate the reliability of the location estimate. However, we interpret the confidences as spatial uncertainties in the returned location. Papanikolopoulos explains the concept of spatial discrimination in detail, giving examples of corner features, edge features, and homogeneous features, along with their respective autocorrelation SSDSs [2]. We illustrate the effect on several features in Section 5.

While conclusions about the efficacy of a given template for feature localization can be drawn from the fully computed SSDS, it is both computationally expensive and memory intensive to maintain the complete surface for this purpose. In the next section, we derive an approximation for $\mathscr{R}D$ that is more practical to maintain and propagate.

## 4 A practical approximation for $\mathscr{R}D$

In order to maintain and use relevant information about the shape of the response distribution, we introduce a mathematical approximation to the distribution given in (2). By suppressing the off peak response of the feature tracking result, this response distribution function converts the SSDS into an approximately Gaussian distribution that contains the feature tracking information we wish to maintain.

### *4.1 Uncertain feature measurements*

The measurement vector $\mathbf{z}_k$ is interpreted as an uncertain location in the $(u,v)$ plane, and modeled as a 2D Gaussian random vector. It is illustrative to analyze the behavior of the density function for this vector with respect to the spatial certainty of the feature tracking result as $\mathbf{R}_k$, the covariance matrix for the vector, changes. For example, if $\mathbf{R}_k = \sigma^2 I$, where $\sigma^2$ is the variance of the vector, the location is equally certain in each direction. The ellipses of equal probability on the density surface

are circles. If $\sigma_u \neq \sigma_v$, where $\sigma_u^2$ and $\sigma_v^2$ are the variances in the $u$ and $v$ directions, the location is more certain in one direction (given by the minor axis of the ellipses of equal probability) than in the other direction (given by the major axis). As the length of the major axis approaches infinity, complete uncertainty on the location along this dimension is asserted. It is well known that the mean and covariance are sufficient statistics for a Gaussian random variable. Therefore, if this Gaussian density surface is sufficient to model the tracking behavior, it is no surprise that the mean and covariance suffice to maintain this information. In the next section we explain how we estimate these quantities from $\mathscr{R}D$.

### *4.2 Parameter estimation from the SSDS*

This section describes a process for analyzing the SSDS to arrive at estimates for the mean and variance of a Gaussian random vector. The density function of this vector acts as an approximation to the response distribution $\mathscr{R}D$ (see (2)) for the purpose of tracking features.

Our work develops a different normalization procedure for $\mathscr{R}D$ that is useful for the evaluation of isolated feature measurements from template images. This procedure differs from that described above in several ways. First, Singh and Allen were comparing quite similar image patches, leading to a very peaked response distribution. Our work compares CG generated template images to actual captured images, leading to a much less peaked distribution. Second, we emphasize the importance of suppressing the off peak response, so that only the image areas with significant agreement to the template image affect the uncertainty.

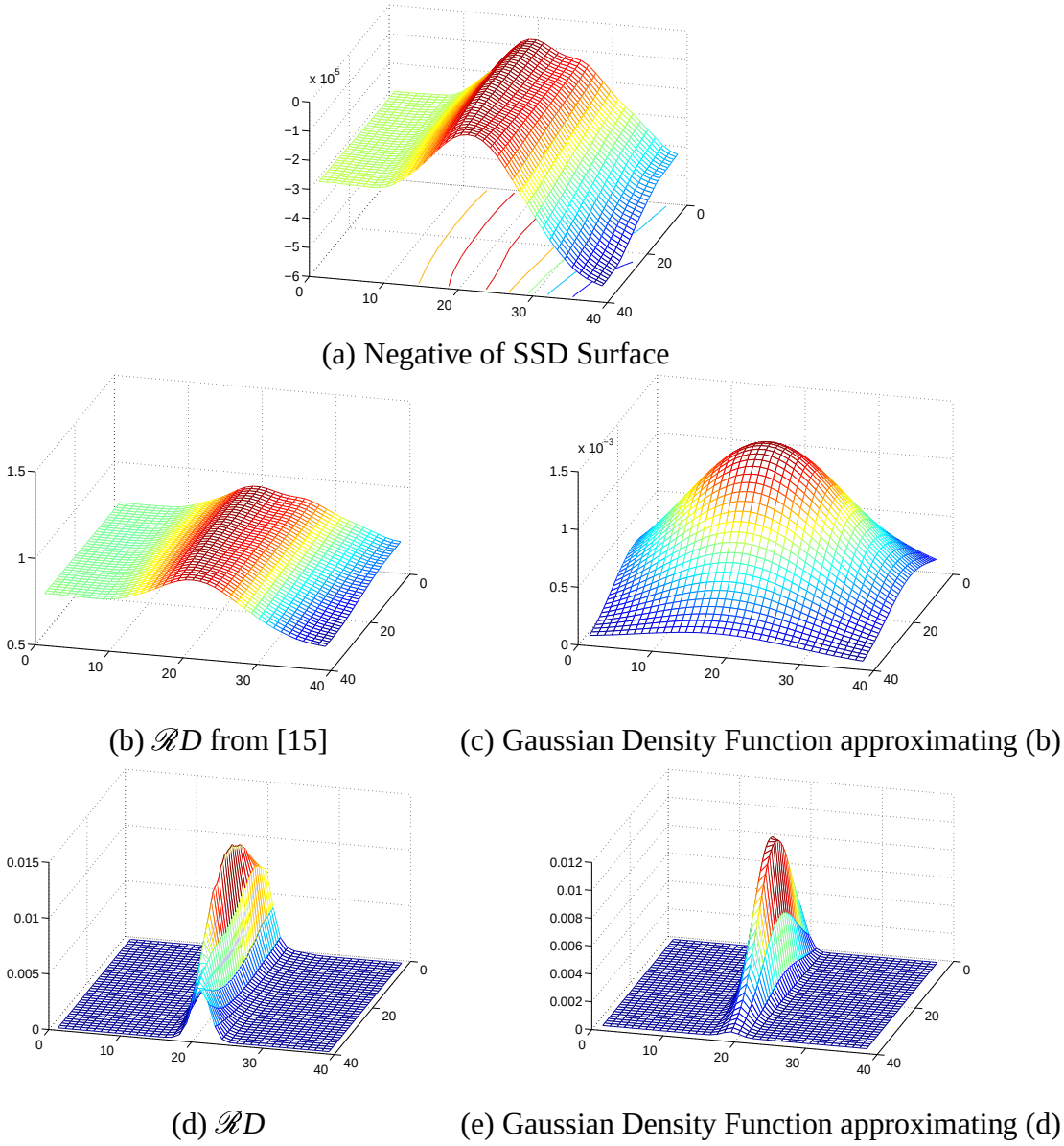
Our computation of the normalization factor $k$ in (2) differs from that of Singh and Allen [15]. We chose $k$ such that

$$\sum_{u,v \in N} \mathscr{R}D(u,v) \approx 1. \tag{4}$$

As can be seen in Figure 1, this has the effect of suppressing the off peak response of the feature detector, when compared with Singh and Allen's normalization. We believe this to be a more appropriate normalization for our situation.

As described in Section 4.2, we compute one covariance matrix and one location for each feature, and use this information in a model-based object tracking framework. We do not reject *any* tracking information but instead weight each measurement on the basis of this covariance matrix, using as much information as possible from the feature tracking.

The mode, or most probable value, of a random vector is located at the peak of the density function. We take the location of the minimum of the SSDS as our value




(a) Negative of SSD Surface

(b) $\mathscr{R}D$ from [15]

(c) Gaussian Density Function approximating (b)

(d) $\mathscr{R}D$

(e) Gaussian Density Function approximating (d)

Fig. 1. Approximation of an example response distribution by a density function. Pixels are along the *x* and *y* axes, and SSD Score or probability mass is along the *z* axis.

for the mode of the vector,

$$\mathbf{z}_k = argmin_{u,v} SSD(u,v). \tag{5}$$

The variance of $u$ ($\sigma_u^2$), the variance of $v$ ($\sigma_v^2$), and the covariance between $u$ and $v$ ($\rho_{uv}\sigma_u\sigma_v$) can be estimated directly from the response distribution using Equations (2) and (3), yielding the desired covariance matrix,

$$\mathbf{R}_k = \begin{bmatrix} \widehat{\sigma_u^2} & \rho_{uv}\widehat{\sigma_u}\sigma_v \\ \rho_{uv}\widehat{\sigma_u}\sigma_v & \widehat{\sigma_v^2} \end{bmatrix}, \tag{6}$$

which, as described above, contains complete information about the *orientation* and *shape* of the error ellipsoids.

By computing the covariance as well as the variances, we 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.

Of course, as we are only maintaining the mean and variance of the random vector, and not the complete SSDS, this is only an approximation to the complete information about local image structure given by the SSD. However, it does give an indication of both the absolute quality of the match and, in cases where edge features [1] exist, the direction of the edge.

## 5 Results

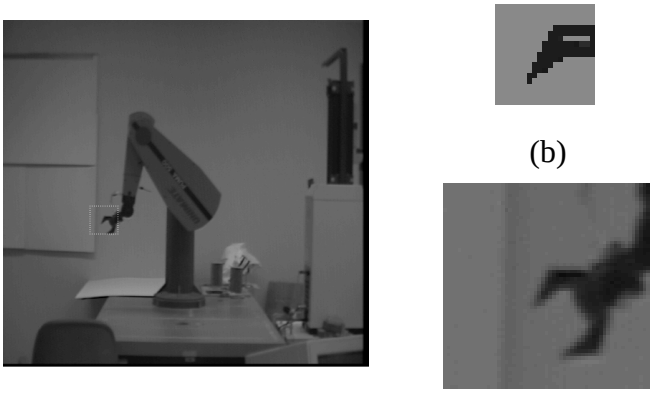
In this section, we review the results of feature tracking with measurement uncertainty estimation. We examine some illustrative examples graphically and quantitatively, and describe the use of this information. Finally, we present a brief example from our related work in object tracking that uses this uncertainty information to aid in tracking an object.

### *5.1 Gripper feature*

The feature considered in this section is a portion of the end-effector of a robot. Figures 2 and 3 show the inputs used for these searches. Two situations will be considered: a reasonably standard input image with the gripper visible as expected and an input image in which the gripper is completely occluded by an external object. The tracking results and computed uncertainties will be shown in both cases.

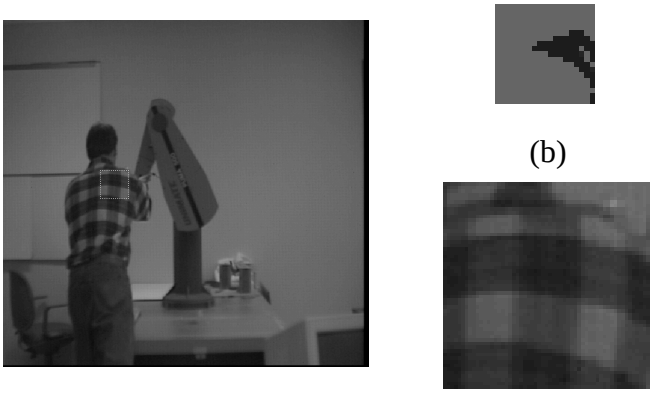
Figures 4 and 5 show the tracking results for these cases. In each figure, the feature location is shown by a cross in (a), the negative SSDS is shown in (b), and the computed probability density function of this location (see Section 4 for details on this computation) is shown in (c). In Figure 4, both the SSDS and GRV density surfaces indicate equal accuracy of the tracking result in all directions.

[1] Our references to *edge features* can refer to any features with low discrimination power in one direction in the current configuration of the object, even if the 3D structure which gave rise to the feature has good texture in multiple directions. See Section 5.3 for more explaination of this case.



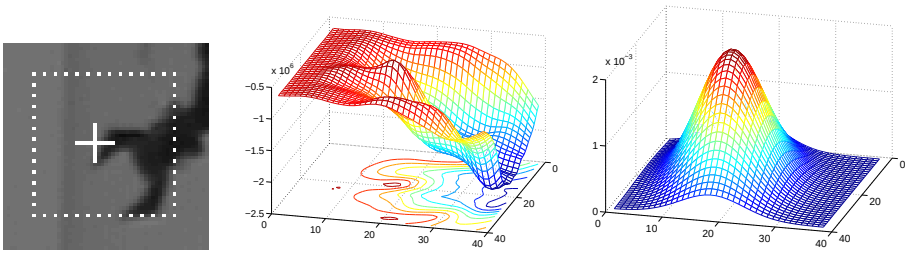
(a) (b) (c)

Fig. 2. Inputs to tracking for gripper feature (unoccluded) (a) Full Input Image with search region marked (b) Template (c) Actual area searched



(a) (b) (c)

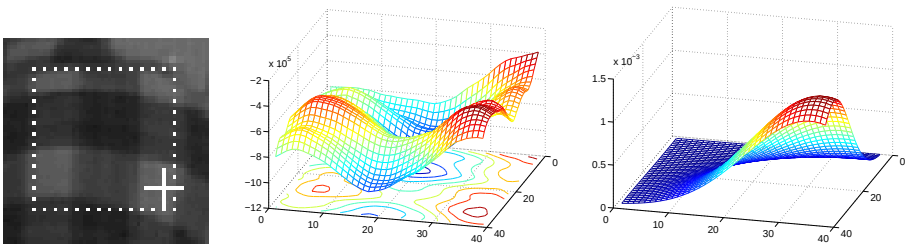
Fig. 3. Inputs to tracking for gripper feature (occluded) (a) Full Input Image with search region marked (b) Template (c) Actual area searched



(a) (b) (c)

Fig. 4. Tracking results for unoccluded gripper feature. (a) Results Image with a dotted box showing the area searched and a cross illustrating the feature location determined (b) Negative SSDS (c) GRV density

In Figure 5, we present an illustration of the usefulness of the on-line estimation of template efficacy. By calling this estimation “on-line”, we do not mean to imply that this method is real-time. With no hardware accelleration for the SSD computations




(a) (b) (c)

Fig. 5. Tracking results for occluded gripper feature. (a) Results Image with a dotted box showing the area searched and a cross illustrating the feature location determined (b) Negative SSDS (c) GRV density

or the response distribution normalization, this algorithm is actually quite slow. On a Sparc 10, each feature takes approximately 15 seconds to compute. Instead, we mean to point out that since the templates used and input images acquired both change as a function of time and configuration, the efficacy measures used must be computed within the tracking loop.

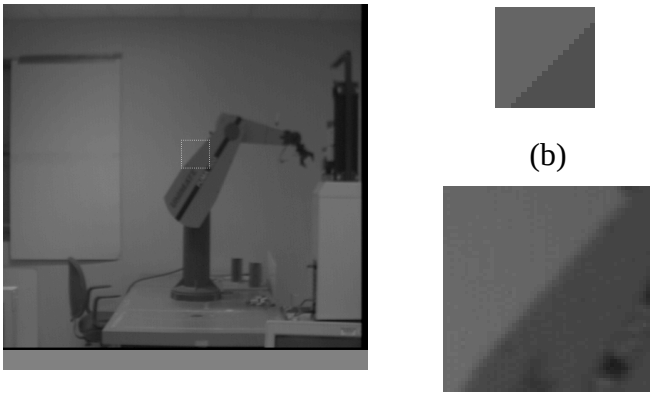
The feature shown in Figure 5 has the same template as in the previous case. However, a person has stepped between the camera and the feature, occluding the feature.

A 2D measurement, represented by the cross in Figure 4(a) and Figure 5(a), and a $2 \times 2$ covariance measurement are the output of the feature tracking, and are used directly in the EKF framework described in [14].

### *5.2 Edge feature*

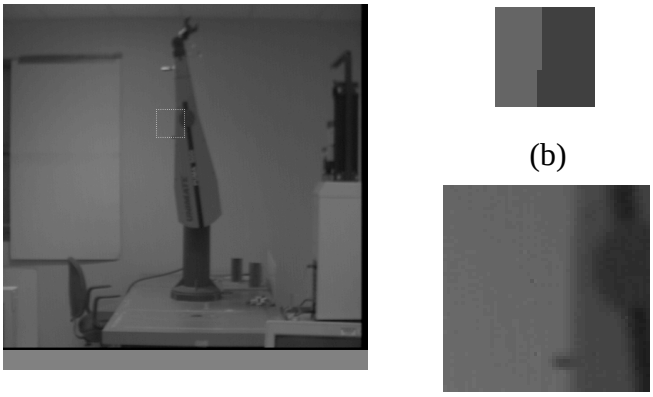
This case illustrates the usefulness of the measurement uncertainty estimation for tracking features with poor spatial discrimination in one direction. An edge feature can be tracked well only in the direction orthogonal to the edge. This feature arises from a point on the edge of the robotic arm. Thus, the orientation of the edge in the feature depends on the configuration of the robot. As the configuration of the robot changes, the direction of the edge projected onto the image plane will change. Figures 6 and 7 show the inputs used for this search.

In Figure 8, the edge is in a diagonal orientation. The negative of the SSDS shown in (b) has a ridge along this direction, indicating good match scores along the ridge. The location of the absolute maximimum of the SSDS (i.e. the returned feature location), is shown by the cross in (a). After normalization, the density function shown in (c) exhibits the same ridge, while suppressing the off peak match scores on both sides of the ridge. Similarly, the edge in Figure 9 is in a vertical orientation, so the ridges in (b) and (c) are in the vertical direction, and the returned feature location shown by the cross in (a) is known to be accurate only in the horizontal



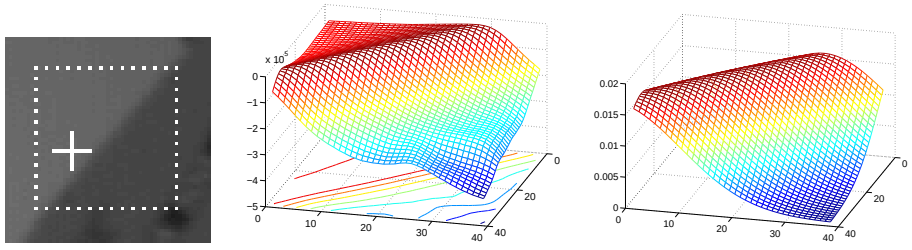
(a) (b) (c)

Fig. 6. Inputs to tracking for edge feature (diagonal configuration) (a) Full Input Image with search region marked (b) Template (c) Actual area searched



(a) (b) (c)

Fig. 7. Inputs to tracking for edge feature (vertical configuration) (a) Full Input Image with search region marked (b) Template (c) Actual area searched

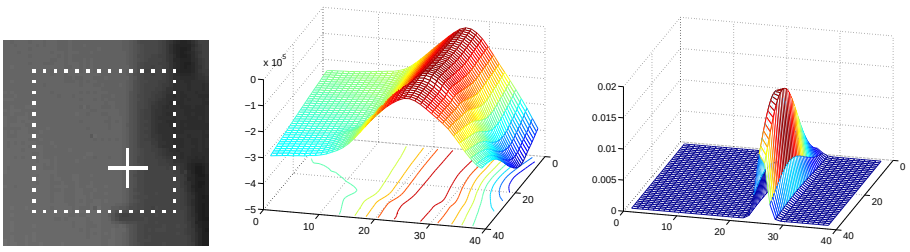


(a) (b) (c)

Fig. 8. Tracking results for diagonal edge feature. (a) Results Image with a dotted box showing the area searched and a cross illustrating the feature location determined (b) Negative SSDS (c) GRV density

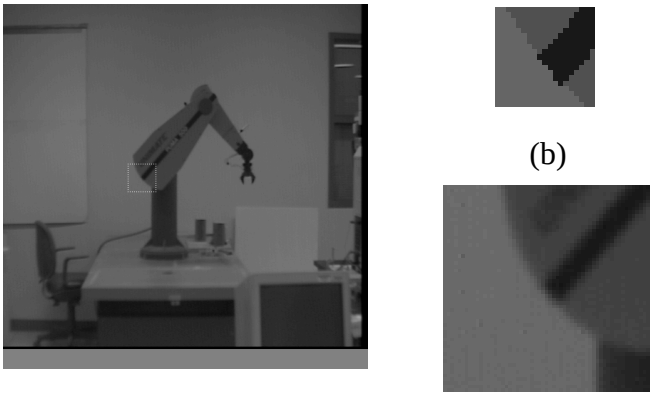
direction only.

By maintaining this information, the system can exploit the feature tracking in-



(a) (b) (c)

Fig. 9. Tracking results for vertical edge feature. (a) Results Image with a dotted box showing the area searched and a cross illustrating the feature location determined (b) Negative SSDS (c) GRV density



(a) (b) (c)

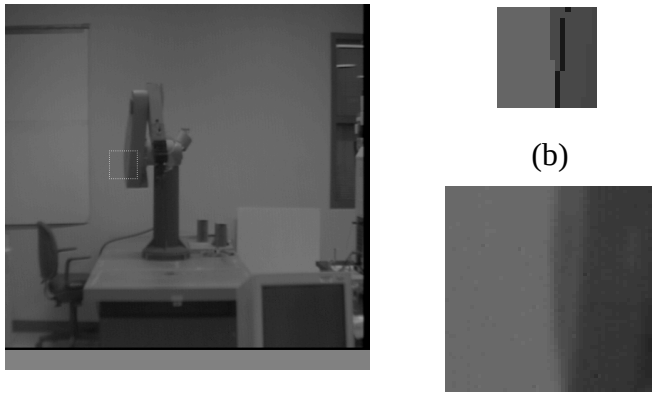
Fig. 10. Inputs to tracking for point feature (nondegenerate) (a) Full Input Image with search region marked (b) Template (c) Actual area searched

formation to its fullest extent. The result is neither endowed with inappropriate confidence due to the good accuracy in the direction orthogonal to the edge nor unduly devalued due to the poor accuracy in the direction along the edge.

### *5.3 Degenerate point feature*

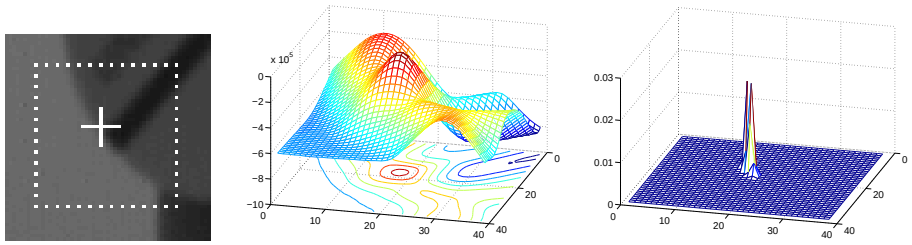
In this section, we illustrate another aspect of the usefulness of on-line estimation of template efficacy. Since our object-tracking system is intended to work under widely varying configurations of the object, the appearance of features may change significantly during tracking. A single feature acceptance or rejectance decision will not suffice in this case.

Figures 10 and 11 show the inputs used for this search. Figures 12 and 13 show tracking results for a feature that undergoes such a change in appearance, a point of intersection of a black line on the edge of the robotic arm with the rear edge of the arm.



(a) (b) (c)

Fig. 11. Inputs to tracking for point feature (degenerate — acting as a line feature) (a) Full Input Image with search region marked (b) Template (c) Actual area searched



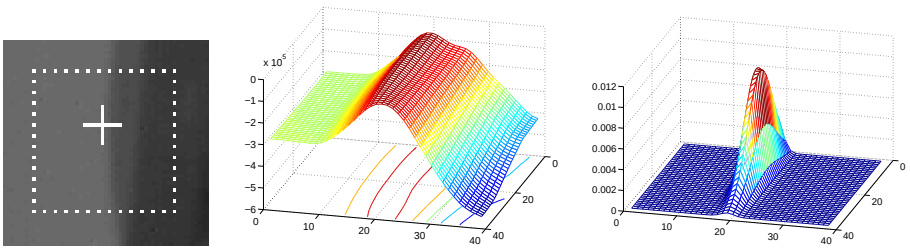
(a) (b) (c)

Fig. 12. Tracking results for nondegenerate point feature. (a) Results Image with a dotted box showing the area searched and a cross illustrating the feature location determined (b) Negative SSDS (c) GRV density

This feature is a point of high texture in both directions when the arm is approximately parallel to the image plane, as shown in Figure 10(a), and acts like a point feature. This feature location, shown in Figure 12(a) by the cross, can be found with high accuracy in all directions as shown in the Negative SSDS shown in (b) and the final density shown in (c).

However, this feature acts like an edge feature in other configurations, such as the configuration shown in Figure 11, where the arm is pointing roughly toward the camera. In this configuration, the feature appears as a vertical edge feature, as can be seen in the search image shown in Figure 11(c). The location of the feature in this case (shown by the cross in Figure 13(a)) can be found with high accuracy in only the horizontal direction. This fact can be seen in the Negative SSDS (b) and the final density (c).

Again, the maintenance of the covariance matrix instead of a single confidence measure makes this suboptimal tracking result not only tolerable, but useful.



(a) (b) (c)

Fig. 13. Tracking results for degenerate point feature. (a) Results Image with a dotted box showing the area searched and a cross illustrating the feature location determined (b) Negative SSDS (c) GRV density

### *5.4 Object tracking using uncertainty*

In each of the previous results, we have shown how the results from tracking a single feature can be analyzed to arrive at an efficacy measure for the location of that feature. In this section, we briefly describe how this type of information from several distinct features can be used to track the movements of an articulated object. We will also demonstrate a case in which the ignorance of this uncertainty information leads to losing track of the object. As we are using the absolute minimum of the SSDS for the feature measurement in both experiments, the *accuracy* of the feature tracking information is not changed. The independant variable in the two experiments is instead the associated *uncertainty information*. Kalman Filter divergence due to incorrect uncertainty information is a well-known phenomenon in the radar tracking literature [23] [24], and has received some attention in the computer vision and robotics community as well [25].

In this example, we are tracking a two degree of freedom planar arm with unknown link lengths. Thus, the system has four total degrees of freedom. We utilize an extended Kalman filter to track this system. For an in-depth look at this object tracking system, see [4]. We present a brief look at the system, from the point of view of utilizing feature tracking information, below. The system used in this example is identical except for the use of measurement uncertainty as described below.

The measurement function is the composition of the arm Jacobian and the image Jacobian (i.e. the mapping from joint angles to image location of features). Using this framework, the measurement equations are implicitly inverted [26] and feature measurements from the images are used to update the configuration of the arm.

We define the state vector for this extended Kalman filter to be

$$\mathbf{x}_k = \begin{bmatrix} q_0 & q_1 & \dot{q}_0 & \dot{q}_1 & a_0 & a_1 \end{bmatrix}^T, \tag{7}$$

where $q_i$ is the angle of the $i$th joint, and $a_i$ is the length of the $i$th link. We assume constant velocity motion in the joints and fixed unknown link lengths. Measurements for this system are the $(u,v)$ locations in the image plane of salient predefined features on the arm,

$$\mathbf{z}_k = \begin{bmatrix} u_1 & v_1 & \cdots & u_F & v_F \end{bmatrix}^T \tag{8}$$

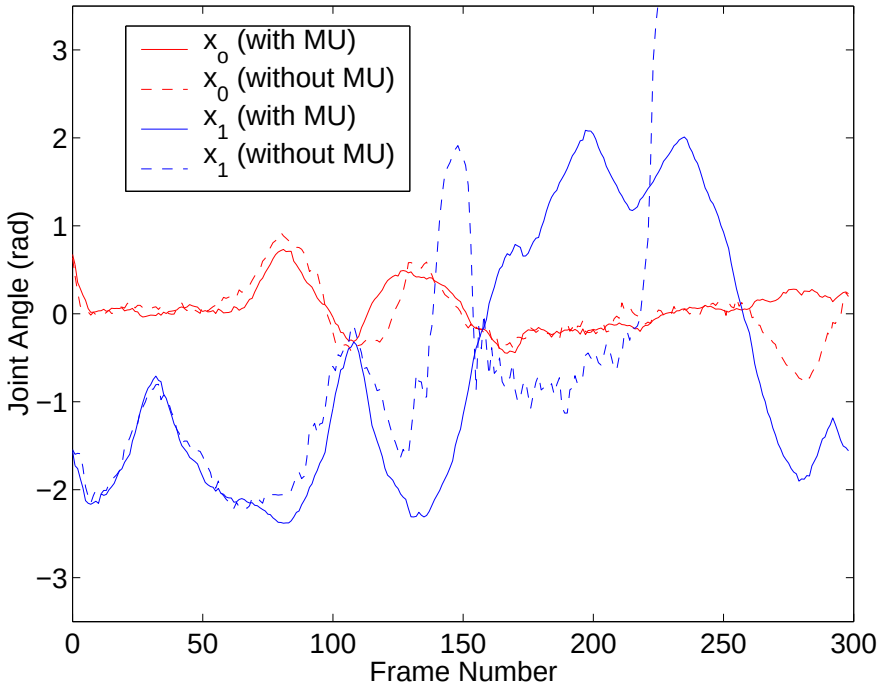
where there are $F$ features being tracked. Each feature measurement has an uncertainty matrix $R_f$ associated with it. These measurements are combined into a system measurement uncertainty matrix, as shown below.

$$\mathbf{R}_k^f = \begin{bmatrix} \sigma_{u_f}^2 & \rho_{u_f v_f}\sigma_{u_f}\sigma_{v_f} \\ \rho_{u_f v_f}\sigma_{u_f}\sigma_{v_f} & \sigma_{v_f}^2 \end{bmatrix}$$

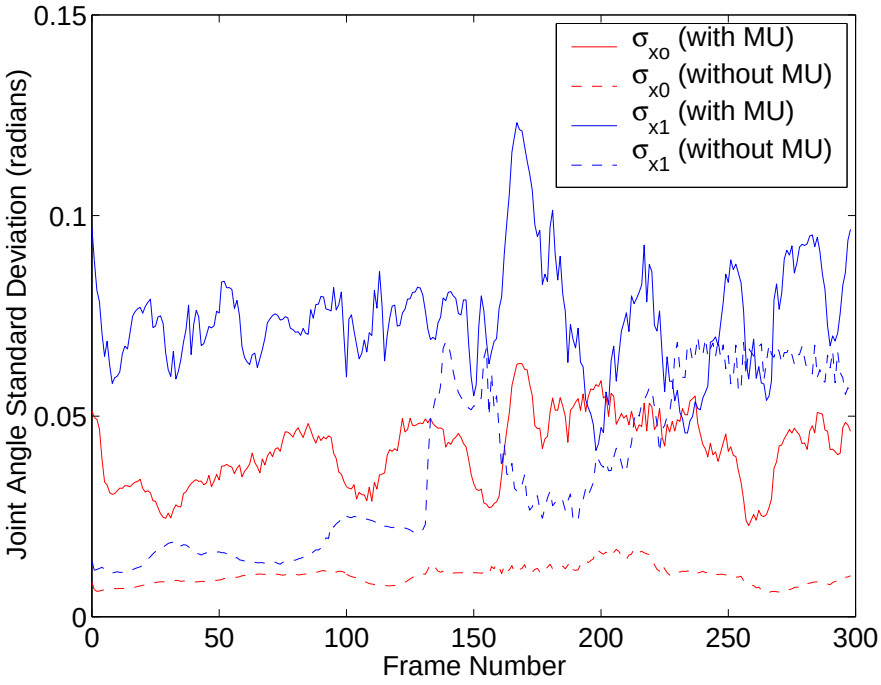
$$\mathbf{R}_k = \begin{bmatrix} \mathbf{R}_k^1(1,1) & \mathbf{R}_k^1(1,2) & \cdots & 0 & 0 \\ \mathbf{R}_k^1(2,1) & \mathbf{R}_k^1(2,2) & & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & & \mathbf{R}_k^F(1,1) & \mathbf{R}_k^F(1,2) \\ 0 & 0 & \cdots & \mathbf{R}_k^F(2,1) & \mathbf{R}_k^F(2,2) \end{bmatrix}$$

Figure 14 shows the results from two object tracking runs. Both runs utilized the same sequence of input images. In the first case, the uncertainty estimates described in this paper are used. In the second, an experimentally determined constant experimental variance was used. The *object tracking* portion of the algorithm was indentical in both cases. The computation of the feature tracking uncertainty was the only change. The solid lines marked $x_0$ (with MU) and $x_1$ (with MU) in Figure 14 represent the angle estimates for joints 0 and 1 respectively, with measurement uncertainty information. The dashed lines marked $x_0$ (without MU) and $x_1$ (without MU) represent the angle estimates for joints 0 and 1 respectively, replacing the actual measurement uncertainty information with the identity matrix multiplied by an experimentally determined scalar variance. This variance was chosen to optimize performance in the system, and worked well in many situations. However, in the case shown the initial link lengths were incorrect by a factor of approximately 2, and the latter system breaks down. Since these tracking results are from an uninstrumented human arm, ground truth is not available. Qualatitively, the estimates given by the former system track the object while the estimates given by the latter system diverge from the actual angles.

By ignoring the augmented feature tracking measurements, the system is not able to identify correctly when the system is doing well and when it has lost track. This fact is more evident in Figure 15, where the estimates for the link lengths are shown. Again, the solid lines indicate the system using measurement uncertainty and the dashed lines indicate the use of an experimentally determined scalar variance.




Joint Angle Estimates

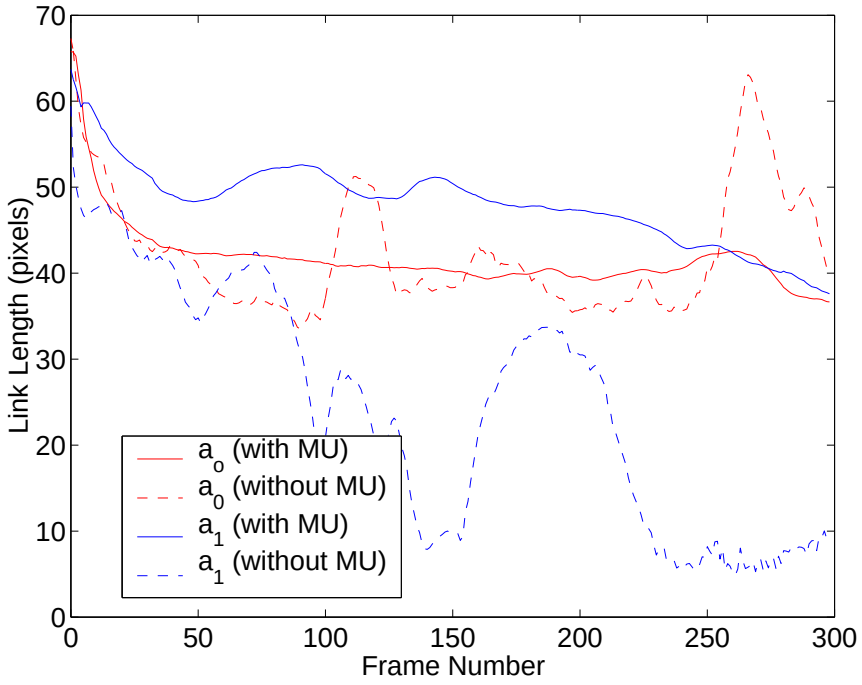



Joint Uncertainty Estimates

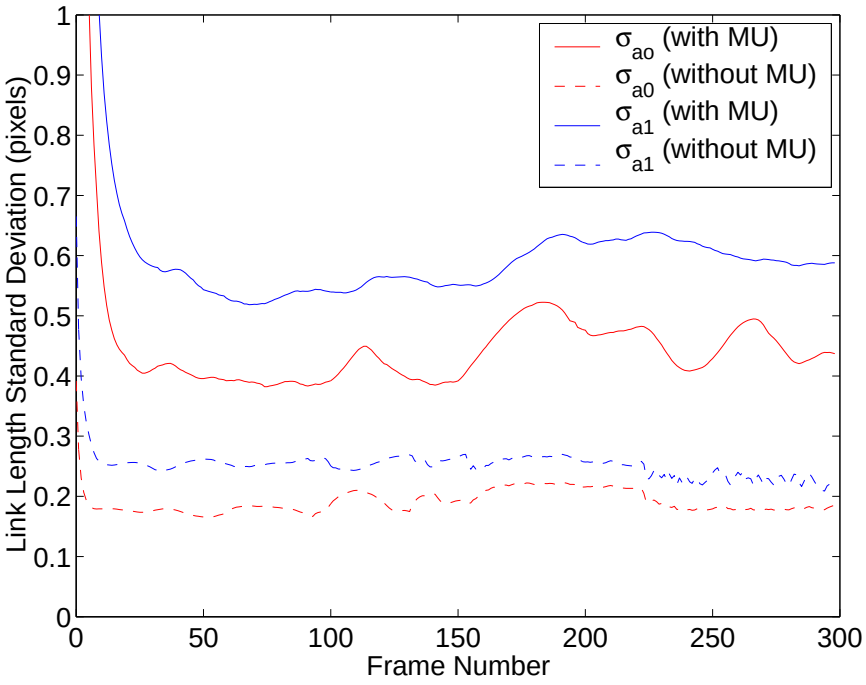
Fig. 14. Object Tracking Results and Measurement Uncertainty - Joint Estimates

## 6 Conclusions

The method presented uses the SSDS, a common intermediate result in correlation-based feature tracking, to compute quantitative estimates for the spatial accuracy




Link Length Estimates




Link Length Uncertainty Estimates

Fig. 15. Object Tracking Results and Measurement Uncertainty - Link Length Estimates

of the feature tracking result. This estimate consists of a covariance matrix for a Gaussian random vector. Analysis of this matrix yields information about the directions (if any) in which the template is discriminating the feature from the image background, and provides a quantitative measure of confidence in each direction.

Several examples were given illustrating the intuitive nature of this quantitative uncertainty estimate. This estimate is particularly useful in long sequences, when a given feature can perform well at some times, and poorly at other times. An example of one use of this information was also presented, in the context of articulated object tracking. More examples of the use of this matrix in model-based tracking of complex articulated objects can be found in [4], along with a more thorough explanation of the Kalman filter based system.

The feature tracking results, combined with the measurement uncertainty matrix, yield a composite measure that is useful when analyzing the tracking results. Analysis of the feature tracking results can detect templates that do not discriminate effectively in any direction. By associating spatial confidence measures with feature tracking results, those results can be more fully exploited: the fact that some directions may have high confidence does not lead us to accept the entire measurement, and the fact that some directions may have low confidence does not lead us to disregard useful data.

## References


[1] J. Shi, C. Tomisito, Good features to track, in: Proceedings IEEE Computer Society Conference on Computer Vision Pattern Recognition, 1994, pp. 593–600.

[2] N. P. Papanikolopoulos, Selection of features and evaluation of visual measurements during robotic visual servoing tasks, Journal of Intelligent and Robotic Systems 13 (3) (1995) 279–304.

[3] P. Anandan, A computational framework and an algorithm for the measurement of visual motion, International Journal of Computer Vision 2 (3) (1989) 283–310.

[4] K. Nickels, S. Hutchinson, Model-based tracking of complex articulated objects, IEEE Transactions on Robotics and Automation 17 (1) (2001) 28–36.

[5] D. B. Gennery, Visual tracking of known three-dimensional objects, International Journal of Computer Vision 7 (3) (1992) 243–270.

[6] D. Lowe, Robust model-based motion tracking through the integration of search and estimation, International Journal of Computer Vision 8 (2) (1992) 113–122.

[7] S. Lee, B. You, G. Hager, Model-based 3-D object tracking using projective invariance, in: Proceedings IEEE International Conference Robotics and Automation, Detroit, MI, 1999, pp. 1589–1594.

[8] J. E. Lloyd, J. S. Beis, D. K. Pai, D. G. Lowe, Model-based telerobotics with vision, in: Proceedings IEEE International Conference Robotics and Automation, Vol. 2, Albuquerque, NM, 1997, pp. 1297–1304.

[9] R. Lopez, A. Colmenarez, 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, 1997.

[10] R. Brunelli, T. Poggio, Face recognition: Features versus templates, IEEE Transactions on Pattern Analysis and Machine Intelligence 15 (10) (1993) 1042–1052.

[11] G. Hager, P. Belhumeur, Efficient region tracking with parametric models of geometry and illumination, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (10) (1998) 1025–1039.

[12] G. Hager, Real-time feature tracking and projective invariance as a basis for hand-eye coordination, in: Proceedings IEEE Computer Society Conference on Computer Vision Pattern Recognition, 1994, pp. 533–539.

[13] M. Woo, J. Neider, T. Davis, The OpenGL Programming Guide, Addison-Wesley, Reading, MA, 1996.

[14] K. Nickels, S. Hutchinson, Weighting observations: The use of kinematic models in object tracking, in: Proceedings IEEE International Conference Robotics and Automation, 1998.

[15] A. Singh, P. Allen, Image flow computation: An estimation-theoretic framework and a unified perspective, Computer Vision Graphics and Image Processing: Image Understanding 56 (2) (1992) 152–177.

[16] S. Birchfield, C. Tomasi, A pixel dissimilarity measure that is insensitive to image sampling, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (4) (1998) 401–406.

[17] P. J. Burt, C. Yen, X. Xu, Local correlation measures for motion analysis: A comparative study, in: Proceedings IEEE Conference Pattern Recognition Image Processing, 1982, pp. 269–274.

[18] G. Hager, P. Belhumeur, Real-time tracking of image regions with changes in geometry and illumination, in: Proceedings IEEE Computer Society Conference on Computer Vision Pattern Recognition, 1996, pp. 403–410.

[19] A. Rosenfeld, A. Kak, Digital Picture Processing, 2nd Edition, Academic Press, Cambridge, MA, 1982.

[20] A. Singh, An estimation-theoretic framework for image-flow computation, in: Proceedings International Conference Computer Vision, 1990, pp. 168–177.

[21] A. Kosaka, A. C. Kak, Fast vision-guided robot navigation using model-based reasoning and prediction of uncertainties, Computer Vision and Image Understanding 56 (3) (1992) 271–329.

[22] K. Nickels, S. Hutchinson, Integrated object models for robust visual tracking, in: Proceedings of Workshop on Robust Vision for Vision-based Control of Motion, 1998.

[23] A. J. Isaksson, F. Gustafsson, Comparison of some Kalman filter based methods for manoeuvre tracking and detection, in: Proceedings of the Conference on Decision and Control, 1995, pp. 1525–1531.

[24] A. Zolghadri, An algorithm for real-time failure detection in Kalman filters, IEEE Transactions on Automatic Control 41 (10) (1996) 1537–1539.

[25] R. Deriche, O. Faugeras, Tracking line segments, Image and Vision Computing 8 (4) (1991) 261–270.

[26] W. J. Wilson, C. C. W. Hulls, G. S. Bell, Relative end-effector control using Cartesian position based visual servoing, IEEE Transactions on Robotics and Automation 12 (5) (1996) 684–696.