Covariance Representations of Partial Geometric Constraints for 3D Pose Estimation

R. Fisher[1], M. Waite[1], M. Orr[2], J. Hallam[1]

(1) Dept. of Artificial Intelligence, University of Edinburgh
5 Forrest Hill, Edinburgh EH1 2QL, Scotland, United Kingdom
(2) Advanced Robotics Research Ltd., University Road,
Salford M5 4PP, England, United Kingdom

Abstract

We explore the potential of variance matrices to represent not just statistical error on object pose estimates but also partially constrained degrees of freedom. Using an iterated extended Kalman filter as an estimation tool, we generate, combine and predict partially constrained pose estimates from 3D range data. We find that partial constraints on the translation component of pose which occur frequently in practice are handled well by the method. One key advantage of the method is that it allows simultaneous representation of both lack of knowledge, weak constraints, *a priori* position constraints and statistical error in a framework that allows incremental reasoning.

1 Introduction

Most model-based part recognition or location vision systems establish all model-to-data pairings during an initial matching phase, and then estimate the pose from the consistent pairings. This is less than ideal, as insufficient features may have been segmented to estimate fully the pose, or it may be desirable to improve the pose estimate by locating additional features using the current pose estimate. Or, some features may only provide partial or weak pose constraints.

This paper integrates three themes in computer vision to show how model matching can be improved. The themes are:

1. incrementally improve pose estimates as new evidence is found,

2. represent both statistical error and lack of knowledge (i.e. partial pose constraints) and

3. use partial knowledge to guide model matching.

The paper demonstrates six examples of model-matching or pose estimation problems where partial knowledge is integrated and used to improve the quality of scene understanding. The domain of application used for examples here is 3D model matching using 3D image feature data, but the approach can be adapted for 2D-to-2D and 3D-to-2D problems. The examples shown in the paper are based on a surface-patch matching system where the data surface patches are extracted from range data (by some adaptations of [3, 9]) and the model surfaces are specialised instances of quadratic surfaces [7].

The foundation of the approach is based on representing the uncertainty by a variance matrix. This by itself is not new and a number of vision, robotics and tracking projects have followed this approach [2, 13, 19, 21]. That work has used the variance representation to encode fully constrained, but statistically erroneous, poses. The advantage of the statistical approach is that there are well-known and understood statistical tools for estimation (e.g. the Kalman filter) and decision problems (e.g. the χ^2 test). Using these, one can test the likelihood of the estimates, determine when new evidence is compatible with existing estimates, continually refine the parameter estimates, integrate evidences with different amounts of uncertainty and determine a most-likely parameter estimate. (Although it is not always easy to apply that theory to non-linear vision problems.)

However, there exists a class of problems where the uncertainty is not entirely due to statistical errors but has a component which would still be present even if the available measurements were perfectly accurate. Such problems occur when there are more parameters to estimate than measurements available – they are underconstrained problems. For example, the correspondence between a model point and a scene plane, in the sense that the transformed point must lie somewhere in the plane, is not sufficient to constrain the translational part of the transform even if the rotational part has been estimated by some other means. These problems leave degrees of freedom in the estimated transform which it would be convenient to represent in the same way as statistical uncertainty, i.e. using variance. As long as the uncertainty due to degrees of freedom is linear in parameter space, we can do this by introducing one or more large eigenvalues in the variance matrix.

The method that we use to solve this problem is to represent the open degrees-of-freedom (i.e. lack of knowledge) by one or more very large eigenvalues in the variance matrix. Unlike the interval bounding method on parameter space (see for example [4], [8]), the covariance (off-diagonal) terms in the variance matrix represent correlations between the components of the parameter and allow degrees of freedom in any direction in the parameter space, not just along the coordinate axes (as is typical of the interval arithmetic[4]). A simple illustration of a fundamental problem with the standard interval method is shown in Figure

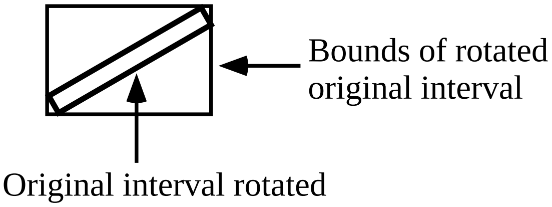
Bounds of rotated original interval
Original interval rotated

Figure 1: *The figure shows how the enclosed volume within coordinate-axis aligned bounds on a rotated interval bound may be much greater than the original volume.*

1, which shows how the bounds of a rotated interval may grow large even though the initial interval was quite tight. Other known problems of the interval method include: very slow to compute when using symbolic algebra and, as no statistical information about measurement error is represented, no mean or best estimate is available.

In practice, the non-linear rotational part of a 3D transform is often easier to constrain than the linear translational part. This is due in part to the robustness of correspondences between plane surface normals against occlusion and segmentation errors. Correspondences between points (e.g. centre of gravity, boundary points) which are used to constrain translation are relatively fragile and subject to occlusion and must sometimes be replaced by partial constraints (such as a match between a point and a plane). Our approach is useful in these situations when the rotation is fully constrained but the translation can only be partially constrained.

What we suggest is, assuming enough evidence is available to constrain rotation (to within measurement errors only), that pairings between scene points and model points which contain degrees of freedom (over and above measurement errors) can be used to generate partial constraints on the translation and that furthermore, combining two or more partially constrained estimates of the same pose can lead to a fully constrained estimate. We could either attach the degrees of freedom to the scene point or the model point. Where possible, we have chosen the model point because the appropriate amount of variation in different directions will be known *a priori* in the model so the variance only has to be set up once and for all. In some cases the directions of the degrees of freedom are unknown in the model space and they must be associated with the variance of the observed parameter.

Based on this representation of uncertainty, we show how six different problems can be solved:

1. A partially constrained translation can be estimated using a planar surface patch match.
2. A partially constrained translation can be estimated using a cylindrical surface patch match.

3. *A priori* problem knowledge can lead to partially constrained translation estimates.

4. A fully constrained translation can be estimated from multiple pieces of partially constraining evidence.

5. A fully constrained pose can be estimated from partially constrained poses for distinct subcomponents of an object.

6. A partial pose estimate can be used to guide image search for additional matchable features.

The solutions to these problems are discussed in separate subsections of Section 3 and the statistical techniques underlying the solutions are described in Section 2. In Section 5 we discuss the problems associated with representing partial rotation constraints and then present our conclusions in Section 6.

The work reported here builds on techniques which have become standard in robotics and vision through the work of, among others, groups at INRIA [21] and Oxford University [13]. The approach to partial evidence representation is similar to that of [4] and [8] except that there intervals, which are known to be inferior to variance matrices [17], were used to represent the bounds on the parameters. There are also links with early research into pose constraints from object relationships as specified in a robot programming language (RAPT) [18] though that work modeled relationships as exact (i.e. without statistical error).

2 The Statistical Framework

2.1 Kalman Filtering

The Kalman filter (and its extension for non-linear problems) is the basic estimation tool we are using. Here we merely give a brief description of its function; more details can be readily found elsewhere, e.g. [2, 12, 1].

Basically, the filter recursively processes observations to arrive at an estimate of an unobserved parameter of interest (the state). Knowledge at time step k (after processing the kth observation) about a parameter or state vector, $\mathbf{x}$, is represented by the estimated mean vector, $\hat{\mathbf{x}}_k$, and variance matrix, $\mathbf{X}_k$, of an assumed Gaussian probability distribution. Observations, $\mathbf{z}_k$, pertaining to the state are themselves uncertain with means, $\hat{\mathbf{z}}_k$, and variances, $\mathbf{Z}_k$. To link the observations to the state there are measurement equations of the form

$$\mathbf{f}_k(\mathbf{x},\ \mathbf{z}_k) \ = \ \mathbf{0}$$

which are usually non-linear and often under-constrained (i.e. cannot be put in the form $\mathbf{x} = \mathbf{g}_k(\mathbf{z}_k)$). The Kalman filter is a tool for incorporating the knowledge in

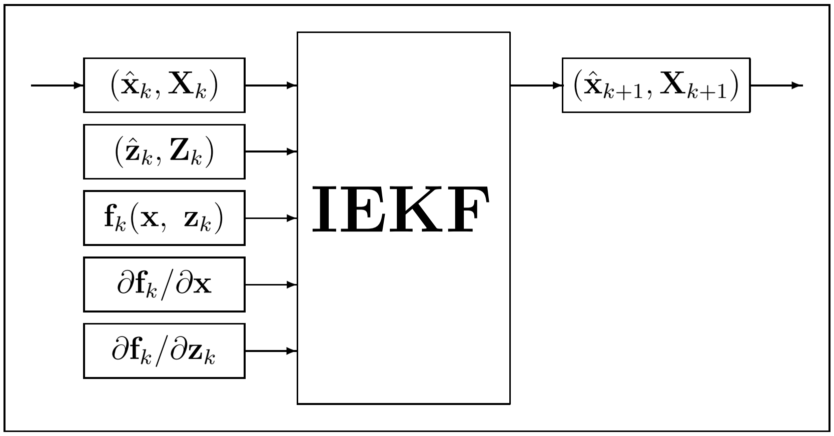
$(\hat{\mathbf{x}}_k, \mathbf{X}_k)$
$(\hat{\mathbf{z}}_k, \mathbf{Z}_k)$
$\mathbf{f}_k(\mathbf{x},\ \mathbf{z}_k)$
$\partial\mathbf{f}_k/\partial\mathbf{x}$
$\partial\mathbf{f}_k/\partial\mathbf{z}_k$
IEKF
$(\hat{\mathbf{x}}_{k+1}, \mathbf{X}_{k+1})$

Figure 2: *Inputs and outputs to the IEFK as it processes the* k*th observation.*

the observations into the state when the measurement equations are linear. The iterated extended Kalman filter (IEKF) is an adaptation of the basic filter to deal with non-linear equations. In both cases incorporating the kth observation leads to an update of the state estimate to a new mean, $\hat{\mathbf{x}}_{k+1}$, and a new variance, $\mathbf{X}_{k+1}$ (see Figure 2).

In addition to the prior state estimate, the kth observation and the kth measurement equation, the IEKF requires input of the Jacobians $\partial\mathbf{f}_k/\partial\mathbf{x}$ and $\partial\mathbf{f}_k/\partial\mathbf{z}_k$ which are functions of $\mathbf{x}$ and $\mathbf{z}_k$. These are necessary to perform the linearisation step inside the IEKF. The appendices list all the measurement functions (and their Jacobians) used in this paper.

The Iterated Extended Kalman Filter updating equations for the problems described below are now given (using the notation to be used throughout the paper). Let:

$$\begin{aligned} \mathbf{M}_k &= \frac{\partial\mathbf{f}_k}{\partial\mathbf{x}} \\ \mathbf{W}_k &= \frac{\partial\mathbf{f}_k}{\partial\mathbf{z}_k}\mathbf{Z}_k\frac{\partial\mathbf{f}_k}{\partial\mathbf{z}_k}^T \end{aligned}$$

where $\mathbf{Z}_k$ is the observation covariance and $\mathbf{f}_k$ is evaluated using the current values $(\hat{\mathbf{x}}_k,\ \hat{\mathbf{z}}_k)$.

Then, the update equation of the *extended* Kalman filter for producing the next estimate $\hat{\mathbf{x}}_{k+1}$ of the state vector is:

$$\hat{\mathbf{x}}_{k+1} \ = \ \hat{\mathbf{x}}_k \ - \ \mathbf{K}_k\,\mathbf{f}_k(\hat{\mathbf{x}}_k,\ \hat{\mathbf{z}}_k)$$

and its associated covariance estimate is:

$$\mathbf{X}_{k+1}^{-1} = \mathbf{X}_k^{-1} \ + \ \mathbf{M}_k^T\mathbf{W}_k^{-1}\mathbf{M}_k$$

where

$$\mathbf{K}_k \ = \ \mathbf{X}_k \mathbf{M}_k^T (\mathbf{W}_k \ + \ \mathbf{M}_k \mathbf{X}_k \mathbf{M}_k^T)^{-1}$$

is the Kalman gain.

If the estimate $\hat{\mathbf{x}}_k$ around which the Taylor expansion is performed is too far from the its true value linearisation may not be very good. A method to reduce the linearisation error is the *iterated* extended Kalman filter. This applies the update equation for the new state mean iteratively, substituting $\hat{\mathbf{x}}_{k+1}$ for $\hat{\mathbf{x}}_k$ and re-evaluating $\mathbf{M}_k$ and $\mathbf{K}_k$ until convergence.

Inconsistency in measurements is detected if the Mahalanobis distance test fails. The error statistic is:

$$\mathbf{f}_k^T(\hat{\mathbf{x}}_k, \ \hat{\mathbf{z}}_k)(\mathbf{W}_k \ + \ \mathbf{M}_k \mathbf{X}_k \mathbf{M}_k^T)^{-1} \mathbf{f}_k(\hat{\mathbf{x}}_k, \ \hat{\mathbf{z}}_k)$$

and a χ^2 test with $\mid \mathbf{f} \mid$ degrees of freedom is performed. If the test fails, then we conclude that the measurements are inconsistent and the current hypothesis should be rejected.

2.2 Representing Lack of Knowledge

The state variance matrix, $\mathbf{X}$, represents the size of an assumed Gaussian probability distribution in n-dimensional space (n is the dimension of the state vector, $\mathbf{x}$). Loosely speaking, it can be thought of as representing an n-dimensional ellipsoid centred on the mean, $\hat{\mathbf{x}}$, and containing the true state vector, $\mathbf{x}$. The ellipsoidal axes are parallel to the eigenvectors of $\mathbf{X}$ in direction and proportional to the square roots of the eigenvalues of $\mathbf{X}$ in length.

The uncertainty in a parameter estimate which has one *linear* degree of freedom can be represented by a variance matrix with a large eigenvalue in the appropriate direction. Two degrees of freedom in two different directions can be represented with two large eigenvalues, and so on. In cases where the degree of freedom is partial (over a finite range) rather than unbounded then suitably sized eigenvalues can be chosen. Of course, for very small uncertainties, non-linear constraints can be linearly approximated, and this is the basis of the usefulness of variance as a general representation of statistical measurement error.

As an example, consider the constraint that a point lies somewhere on a line. The position of the point, $\mathbf{x}$, is the state we wish to estimate and estimates of the end-point, $\mathbf{e}$, and direction, $\mathbf{d}$ (a unit vector), of the line as well as the distance, λ, of the point from the end-point are the observations supplied. If the uncertainty of the estimate of λ is very large then the variance matrix of the estimated position will have a large eigenvalue in the direction of $\mathbf{d}$. A similar problem to this occurred in [21] where it was necessary to represent lines from a stereo system which had good estimates for position and orientation but poor estimates (due to occlusion) for length.

One way to calculate the variance matrix of a partially constrained vector is to generate a first order approximation to the variance (as in [21]). For the problem of a point lying on a line this method starts with the equation for the point position

$$\mathbf{x} \;=\; \mathbf{e} + \lambda \mathbf{d} \tag{1}$$

and the first order approximation for the variance

$$\mathbf{X} \;=\; \frac{\partial \mathbf{x}}{\partial \mathbf{e}} \mathbf{E} \frac{\partial \mathbf{x}}{\partial \mathbf{e}}^T + \frac{\partial \mathbf{x}}{\partial \lambda} \Lambda \frac{\partial \mathbf{x}}{\partial \lambda}^T + \frac{\partial \mathbf{x}}{\partial \mathbf{d}} \mathbf{D} \frac{\partial \mathbf{x}}{\partial \mathbf{d}}^T \tag{2}$$

where $\mathbf{E}$, Λ and $\mathbf{D}$ are the variances of the estimates for, respectively, $\mathbf{e}$, λ and $\mathbf{d}$. From (1) the Jacobians can be derived and substituted in (2), which leads, in this case, to

$$\mathbf{X} \;=\; \mathbf{E} + \Lambda \hat{\mathbf{d}} \hat{\mathbf{d}}^T + \hat{\lambda}^2 \mathbf{D} \tag{3}$$

where $\hat{\lambda}$ and $\hat{\mathbf{d}}$ are the means of the estimates for, respectively, λ and $\mathbf{d}$. To obtain a one-dimensional degree of freedom of the point $\mathbf{x}$ along the line, Λ is set to some suitably large value.

This sort of computation is very like what the IEKF does. The main differences are (1) being recursive, the IEKF requires an initial state estimate which influences the new state estimate and (2) when the measurement equation is non-linear, the iterative linearisation produces a more accurate result. In the above example equation (1) would be used as the measurement equation. A nominal initial estimate could be set up as

$$\begin{aligned} \hat{\mathbf{x}}_0 &= \hat{\mathbf{e}} + \hat{\lambda} \hat{\mathbf{d}} \\ \mathbf{X}_0 &= \sigma^2 \mathbf{I} \end{aligned}$$

($\hat{\mathbf{e}}$ is the mean of the estimate for $\mathbf{e}$) and the (single) observation is

$$\begin{aligned} \hat{\mathbf{z}}_0 &= \begin{bmatrix} \hat{\mathbf{e}} \\ \hat{\mathbf{d}} \\ \hat{\lambda} \end{bmatrix} \\ \mathbf{Z}_0 &= \begin{bmatrix} \mathbf{E} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \Lambda \end{bmatrix} \end{aligned}$$

The uncertainty of the initial estimate, σ^2, can be made high to diminish its influence and the variance, $\mathbf{X}_1$, calculated by the filter will match the result of evaluating (3). Note that since $\mathbf{d}$ is a unit vector its variance, $\mathbf{D}$, is singular (for details see [21]).

To combine estimates a Kalman filter is required, but for merely generating variances with degrees of freedom an alternative method can be used. The steps involved are (1) set up a diagonal matrix where one or more of the diagonals are large (corresponding to the degrees of freedom) and the others are small or zero and (2) rotate this matrix so that the large eigenvalues line up with the directions of the degrees of freedom. However, this method depends on being able to sensibly choose the diagonal entries and the rotation matrix and it is not always obvious how to do this. We could, for example, represent the uncertainty of a point which lies somewhere along a line whose length is of the order of σ by

$$\mathbf{\Phi} \begin{bmatrix} \epsilon^2 & 0 & 0 \\ 0 & \epsilon^2 & 0 \\ 0 & 0 & \sigma^2 \end{bmatrix} \mathbf{\Phi}^T$$

where $\mathbf{\Phi}$ is any rotation matrix which rotates the z-axis into the line direction. The smaller eigenvalue, ϵ^2, can either be chosen to represent measurement error, in the case where the parameter vector is a measured quantity, or be set to zero for a model parameter[1]. This is the method we use to construct partially constrained observation vector variances for the illustrative examples later in the paper. While this is fine for illustration purposes, applications where the accuracy of the uncertainty estimate is more critical may demand one of the two more elaborate methods of calculating the variance.

2.3 Representing Partial Visual Knowledge

In this research, the key geometric representation is the pose, represented here by a 6-vector $\vec{P}$ and a covariance matrix Λ. The 6-vector consists of the three standard (X, Y, Z) translation parameters and a three parameter exponential-form rotation specification $\vec{r} = \theta\vec{\omega}$ where θ is the amount of the rotation about the unit vector rotation axis $\vec{\omega}$. We make the assumption that the errors in the pose parameters are represented by a multi-variate Gaussian distribution (this is undoubtedly false, but it is a simplifying and effective model).

As described above, small variances in the covariance matrix represent the statistical uncertainty arising from erroneous image measurements. To represent lack of knowledge, we use large covariances. The typical types of partial knowledge and their representation are:

- **a point lies in a given plane:** An example of this scenario occurs when a planar data patch is observed and we know that that it must be coplanar with a suitably rotated and translated model plane (i.e. the data patch is an observed fragment of the model patch). As the data patch is (usually) only a subset of the full object patch, any given model point does not necessarily have a corresponding point in the data patch. Hence, we can only conclude that any point in the model plane must lie in the plane

[1] We find it advantageous to avoid singular matrices and instead use a tiny number in place of zero.

defined by the data. Similarly, we can also exploit the fact that the model and data surface normals must be parallel, which constrains the rotation, as discussed below.

This constraint has two degrees of freedom, so is represented by a "pancake" shaped Gaussian distribution in the translation portion of the parameter space. The thickness of the pancake is the variance determined by the measurement error on the point, but the width variance of the pancake is a large value. The mean position for the point can be any point that lies in the given plane, but we use a point near the centre-of-mass.

- **a point lies on a given line:** A scenario of this type arises when we estimate the axis of a cylindrical data patch and conclude that it must be colinear with a suitably rotated and translated model cylinder axis. The translation constraint has one degree of freedom, so is represented by a "cigar" shaped Gaussian distribution in the translation portion of the parameter space. The thickness of the cigar is the variance determined by the measurement error on the point, but the length variance of the cigar is some large value. The mean position for the point can be any point that lies in the given line, but we use a point near the centre-of-mass.

One problem with this representation is that the multivariate Gaussian distribution, even with large variances, expresses some constraint in the supposedly unconstrained directions, and hence has a most likely parameter, whereas we want to represent true lack of knowledge. Ideally some form of infinite variance representation should be used and manipulated. However, by setting up the problems using initial mean estimates that are close to the true mean and then using very large variances (so that the distribution is approximately flat in the region of the allowed parameter values), we avoid significant distortion of the results. Use of the information form of the Kalman filter might avoid this problem.

This approach to partial evidence representation is similar to that of [4] and [8], except that there an interval encoding was used to represent the bounds on the parameters (and also included limits on the "within-plane" degree-of-freedom). Using the covariance matrix allows encoding correlations between parameters, and hence can record degrees of freedom not aligned with the coordinate axes.

3 Six Applications of Partial Constraints

This section shows how a number of scene understanding problems can be represented and solved using this uncertainty approach. The first two problems (Sections 3.1 and 3.2) are examples where partial translation constraints can be generated from matches between points, one of which is partially constrained. Section 3.3 shows how, in a similar manner, the representation can also support some types of *a priori* evidence about feature positions. The next two Sections both illustrate the combining of partial pose estimates, Sections 3.4 for estimates of the same pose and Section 3.5 for estimates of the poses of distinct subcomponents of an object. Section 3.6 shows how partially or fully constraining position evidence can be used to predict the location of additional features.

3.1 Planar Patch Matching

Suppose the model-matching and reasoning module of a vision system has paired a number of model and data planar patch surface normals and from these estimated a rotation by using the method described in Appendix G, and with variances estimated by using an IEKF with the measurement equation detailed in Appendix A. An estimate of the translation has yet to be made but a constraint is available from the pairing of a model patch central point and an observed (possibly partially occluded) scene patch (the true central point is unknown, due to occlusion or segmentation effects). Three (of the six) spatial degrees of freedom are already constrained. One translational degree of freedom can be constrained by the requirement that the transformed model point must lie in the plane of the data surface and there are loose constraints on the other two because the incomplete data patch must lie within the boundaries of the transformed model patch.

One way to account for the partial constraint is to create a pairing between the infinite plane parameters of the model and data patches. However, a better method, which accounts, at least in a crude way, for the finite size of the patch, is to create a pairing between the scene point and the model point and give the model point large variance eigenvalues in the plane of the model patch. The variance of the model point then has the characteristic elliptical shape

$$\mathbf{\Phi} \begin{bmatrix} \sigma_1^2 & 0 & 0 \\ 0 & \sigma_2^2 & 0 \\ 0 & 0 & \epsilon^2 \end{bmatrix} \mathbf{\Phi}^T$$

where σ_1 and σ_2 are the major and minor axes of the smallest ellipse fitting around the model patch and ϵ^2 is the variance estimate of the perpendicular position of the data plane. $\mathbf{\Phi}$ is the rotation matrix which rotates the z-axis into the surface normal of the patch and the x-axis into the major axis of the surrounding ellipse.

Having created the point-to-point pairing and attached the variances as indicated, the constraint can be processed into a new estimate for the translation using a Kalman filter and the measurement equation in Appendix B.

3.2 Cylindrical Patch Matching

As for the previous section, we suppose the rotation component of a pose estimate has already been established, but this time we suppose that the constraint on translation comes from a pairing between a cylindrical model patch and a cylindrical data patch. When rotated and translated into position, the model patch must have the same axis as the data patch (within measurement errors).

We can account for this partial constraint by pairing up the central point of the scene patch axis with the central point on the model axis and by giving the model point a degree of freedom in the direction of the cylinder axis. The variance matrix of the model point is

$$\mathbf{\Phi} \begin{bmatrix} \epsilon^2 & 0 & 0 \\ 0 & \epsilon^2 & 0 \\ 0 & 0 & \sigma^2 \end{bmatrix} \mathbf{\Phi}^T$$

where σ is of the order of half the model axis length, ϵ^2 is the variance of the estimated translation of the data axes, and $\mathbf{\Phi}$ is any rotation matrix which rotates the z-axis into the data axis.

As in the previous subsection, we can then process the constraint using the measurement equation in Appendix B.

3.3 *A Priori* Knowledge

This statistical framework is also suitable for exploiting *a priori* knowledge of the position of the object. For example, we might know that the object is face up. This knowledge defines rotation and translation constraints analogously to those constraints defined from observed feature relationships. (However, when no fully constrained estimate of the rotation is available, such *a priori* knowledge usually leads to non-linear coupled constraints between translation and rotation which cannot be represented with a variance matrix.) We illustrate this idea with four types of constraints.

1. *A known model point lies in a known scene plane:* An example of this constraint is when an object is known to be lying such that one of its corners is lying on the work surface. This knowledge alone does not constrain the orientation of the part in a way that is representable with a variance matrix. However, once the rotation is known, the point constraint defines the translation to lie in some plane. Since we cannot tell *a priori* in which direction the surface normal is in the model frame, we are forced, unlike in Section 3.1, to attach the degrees of freedom to the data point, using a variance matrix of the form

   $$\mathbf{\Phi} \begin{bmatrix} \sigma^2 & 0 & 0 \\ 0 & \sigma^2 & 0 \\ 0 & 0 & \epsilon^2 \end{bmatrix} \mathbf{\Phi}^T$$

   where $\mathbf{\Phi}$ is any rotation matrix which rotates the model z-axis into the surface normal of the work bench, σ is the size of the work bench and ϵ represents measurement error.

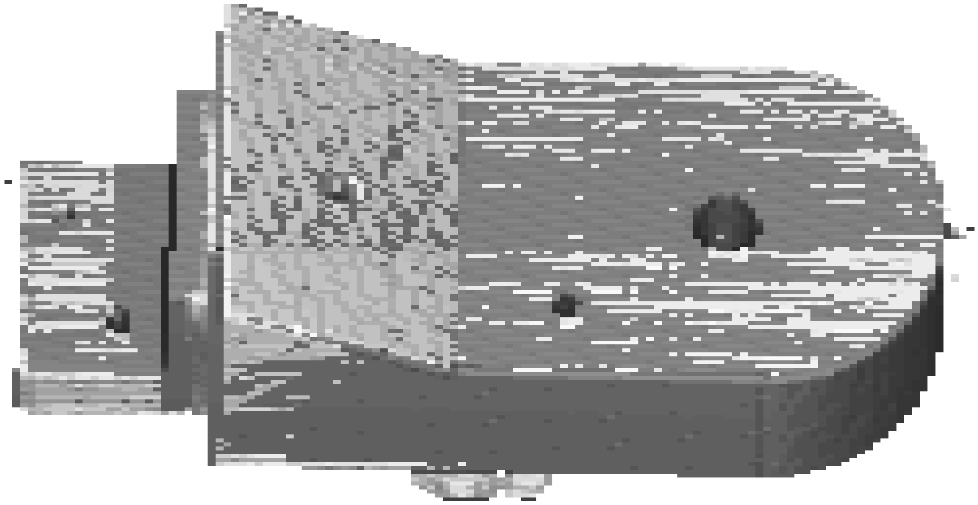

Figure 3: *This figure shows the estimated position of the object using position constraints from the sloping and top surfaces, plus the a priori assumption that one of the points (not visible here) on the bottom of the object lies on the ground plane. The mottled effect arises from the interweaving of the model surface (dark pixels) and the range data (light pixels), because the range data has a standard deviation of approximately 0.15 mm and the estimated object position leaves the model surfaces within that distance of the corresponding range points.*

Figure 3 shows the result of using an *a priori* position constraint as part of the evidence used to estimate the object's position.

2. *Two known planar model points (or a given model edge) lie in a known scene plane:* An example of this constraint is when an object is known to be lying such that one of its straight edges is lying on the work surface. When the rotation has already been estimated, this knowledge does not constrain the translation beyond that of a single point (see last case), as the pair of points can still move freely within the scene plane. If a rotation estimate has not yet been made the information from the constraint cannot satisfactorily be represented in the variance matrix even though the direction of the vector between the points is constrained to lie in the plane. Coupling between the rotation and translation ensures the nature of the constraint is non-linear.

3. *A known model direction is parallel to a known scene direction:* Examples of this constraint arise from knowing two surface normals are aligned, or the axes of two cylinders, or that a cylinder axis is perpendicular to the surface on which it sits, etc. The pairing of model to scene directions defines a rotation with a single degree of freedom. This can be crudely represented by a rotation variance whose smallest axis lies parallel to the vector difference of the two directions. It is not a particularly useful constraint to have

unless it can be combined with other partially constrained rotations (where the methods of Appendix G can be used). Because the actual constraint is stronger than that expressible in the variance matrix, predictions of the object's orientation on the basis of the variance (see Section 3.6) are not sufficiently constrained. Section 5 contains further discussion of partial rotation constraints.

4. *A given model plane lies in a known scene plane:* An example of this constraint is when we know that an object's base is lying on a particular scene surface. This constraint is equivalent to the combination of two previous constraints: the aligned direction constraint (from the surface normals – see case 3 above) and the point-in-plane constraint. Although stronger than the two-points-in-plane constraint of case 2 above, the combined constraint still cannot be represented in a variance matrix due of the non-linear coupling between translation and rotation.

3.4 Integration of Partial Estimates

In general, if model-matching has produced a sufficient number of direction matches to constrain the rotation, then there will be just as many partial constraints on translation by pairing up model and data points, since each surface patch contributes one normal and one central point. The combination of three or more partial constraints from point matches will, except in degenerate cases, lead to a fully constrained translation estimate where the eigenvalues of the variance matrix are primarily determined by the measurement errors. (But these might be quite large and additional measurements could reduce errors.)

To achieve this combination of constraints, each point-to-point pairing is processed by the IEKF using the measurement equation and Jacobians given in Appendix C. The output state estimate from the processing of one pair becomes the input estimate for the next pairing. The initial estimate contains the previously estimated rotation and an estimated mean translation (see Appendix G). The final estimate, barring accidental alignment of degrees of freedom, will not have any large variance eigenvalues (assuming that sufficient independent observations have been made).

For example, this approach can be used to estimate the position of an object consisting of several surfaces, three of which are observed, by first estimating its rotation from paired surface normals (or other constrained directions, such as axes of curvature or the vector between two points) and then using paired points to constrain its translation. The model points have large eigenvalues in the model planes and only the combination of all three pairings is sufficient to constrain translation to within measurement errors. Figure 4 shows the relative position of the model and data after each of the translation constraints have been incorporated into the pose estimate. When a model surface (dark) is close to a

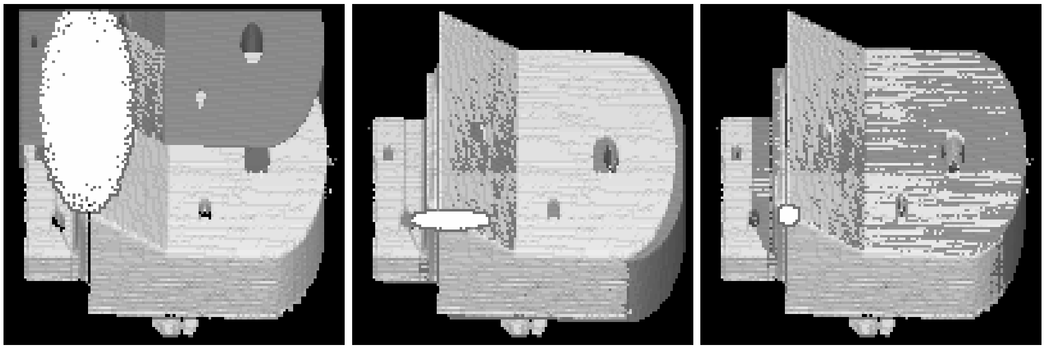

Figure 4: *A series of images showing the increased agreement between the mean position of an object model (dark) and the position of some real data (light) as three partial translation constraints are used to refine the pose estimate. The first image uses a constraint from the sloped surface. The second merges a constraint from the cylindrical front surface of the object and the final uses the large top surface. The decreasing variance of the model position is also depicted by showing the decreasing size of the uncertainty ellipsoid associated with one of the model vertices.*

data surface (light) the graphics program which produced these figures tends to intermingle dark and light pixels. The intermingling between the dark model and light data points arises because the range data has a standard deviation of approximately 0.15 mm and the estimated object position leaves the model surfaces within that distance of the corresponding range points. The effect shows clearly which surface, or surfaces, have been used to constrain the translation in each image. In each image the variance of one of the object model's vertices has been depicted by drawing an ellipse around the predicted position of the point whose size corresponds to the square root of the eigenvalues and which is aligned with the eigenvectors. The ellipse can be seen to shrink in size as the second and third translation constraints are added.

3.5 Integration of Subcomponent Positions

Most pose estimation processes use raw feature information (i.e. point positions and vector directions) as their inputs. However, if a model subcomponent hierarchy is used, it is also possible to use partially or fully constrained subcomponent positions to estimate the pose of the full object [8]. This allows a "hierarchical synthesis" [20], bottom-up recognition of the object from previously recognised subcomponents. Abstractly, the pose estimation process requires three support functions [16]: 1) inversion of the transform between the subcomponent and object frames, 2) composition of the subcomponent pose estimate with the inverted transform to obtain a pose estimate for the parent, and 3) merging the new estimate with the old. With the IEKF and a suitable measurement function we can

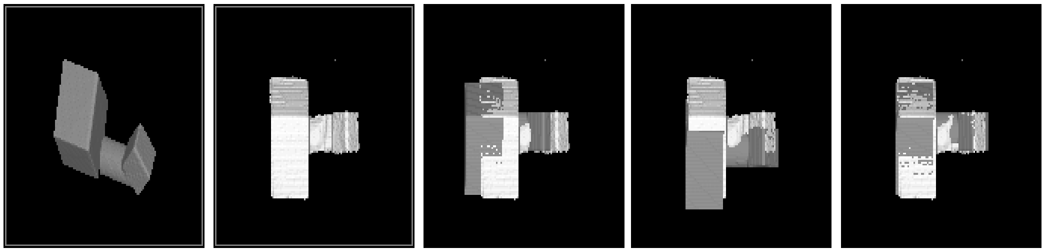

Figure 5: *A series of images showing the increased agreement between the mean position of an object model (dark) and the position of some real data (light) as two partially constrained subcomponent positions are used to refine the pose estimate. The model, which consists of one small block and one large block subcomponents, as seen in the first picture. The data is shown in the second picture. Neither subcomponent alone can accurately estimate the object's pose (third image for when using the large block subcomponent and fourth image for when using the small block subcomponent). The large dark model patches not completely overlapping the lighter data patches indicates that the part is incompletely constrained and has drifted. The combination of both lead to an accurate estimate (image on the right).*

combine all three into one.

If $\mathbf{p}_{cs}$ is the position of the subcomponent in the camera frame and $\mathbf{p}_{ps}$ is the position of the subcomponent in the parent object's coordinate system (given in the model), then the parent object's position in the camera frame, $\mathbf{p}_{cp}$, is the composition of $\mathbf{p}_{cs}$ with the inverse of $\mathbf{p}_{ps}$. We can write

$$\mathbf{p}_{cp} \ = \ \mathrm{compose}\,(\mathbf{p}_{cs},\ \mathrm{inverse}\,(\mathbf{p}_{ps}))\ .$$

If two or more estimates of the parent object's position, $\mathbf{p}_{cp}^{(1)},\ \mathbf{p}_{cp}^{(2)},\ \ldots$ arise from several subcomponents, then the estimates can be merged (averaged)

$$\mathbf{p}_{cp} \ = \ \mathrm{merge}\,(\mathbf{p}_{cp}^{(1)},\ \mathbf{p}_{cp}^{(2)},\ \ldots)\ .$$

The observed poses may be only partially constrained, having been generated from pairings of the type discussed in Sections 3.1 and 3.2. In general, any degrees of freedom in the translation parts of the subcomponent poses will intersect to give a final estimate for $\mathbf{p}_{cp}$ in which the variance eigenvalues are mainly the result of measurement error alone (assuming at least three independent position constraints; otherwise some degrees-of-freedom will remain). Appendix E gives the measurement function and Jacobians which effectively combine the compose, merge and inverse functions required to perform these pose estimates.

Figure 5 shows an example involving the accurate estimation of an object's pose from estimates of the pose of its two subcomponents even though neither subcomponent's translation is fully constrained. The model consists of two rectangular blocks, one large and one small. Both blocks have received pose estimates on the basis of two direction pairings and two partially constraining point pairings of the sort in Section 3.1. Alone, each subcomponent can only generate a partially constrained estimate of the parent's pose, but together the pose estimate contains no degrees of freedom, only measurement error.

3.6 Search for Missing Features

Once a few model features are recognised and a complete pose is estimated, the pose estimate can be used to predict the image position of additional, unmatched model features (e.g. [5, 8, 10]). Direct image verification can then occur.

In the context of our approach to representing degrees of freedom, it is possible to make such predictions even if only partial pose estimates are known. For example, given an estimate of position $\mathbf{p}$ and an estimate of model point $\mathbf{x}_m$, the range of possible *scene* positions for the feature is given by the estimate of the point $\mathbf{x}_d$, the transform of $\mathbf{x}_m$ by $\mathbf{p}$. With this information, one could predict the range of *image* positions for which it is (e.g.) 95% likely that the feature appears. Observed features in this region are then likely candidates for the desired model feature.

The problems of predicting transformed points, directions and subcomponent positions can be solved with the IEKF by a suitable adjustment of the state and observation vectors and rearrangement of the Jacobians used for the corresponding pose estimation problems (see Appendices B, D and F). This is how the ellipsoids in Figure 4 were produced. They represent the uncertainty of the predicted position of one of the object model vertices. The estimated object pose, the vertex position and the measurement equation and Jacobians in Appendix D were given as input to the IEKF and the output was the mean position and variance of the point in scene coordinates. This was used to generate the size, position and orientation of the ellipsoid in the image. As the object pose estimate gets more accurate the ellipsoid size shrinks.

Figure 6 shows the predicted position of two unmatched planar and cylindrical model patches superimposed over the raw range data, where the missing patch positions were predicted from a position estimated from matching other patches.

4 Practical Details

Section 3 described the general theory of the partial constraints as used in the IMAGINE II system. Unfortunately, there is somewhat more to their use: in order to obtain the best results, it is necessary to use good initial states to the Kalman filter. We believe that this is a consequence of the non-linear rotation aspects of the problem.

When estimating the pose of an object, the main steps in the process are:

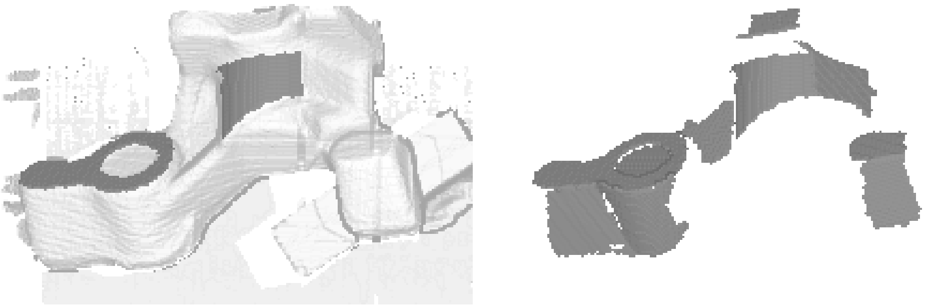

Figure 6: *The dark grey area in the left figure shows the predicted position of two unmatched model patches superimposed on the range data, when using the estimated mean position of the object obtained from matching other patches. The right figure shows the full model used in the match in its estimated position.*

```
Set up an empty constraint set
Add a priori constraints
Add planar surface rotation and translation constraints
Add cylindrical surface rotation constraints
Add subcomponent constraints
If rotation still underconstrained
    add centroid rotation constraints
If translation still underconstrained
    add cylindrical surface translation constraints
Merge constraints to get mean pose
Use Kalman filter to refine and get pose variance
```

Each of these steps is described in the subsections in Appendix G.

5 Partial Rotation Constraints

The method is not able to cope with non-linear constraints such as the coupled constraints that are often generated between rotation and translation when there is no initial rotation estimate (see Section 3.3, cases 2, 4).

For example, a single direction pairing constrains the rotation vector to lie on a closed curve lying in the plane of symmetry between the two vectors. The curve constraint, of course, cannot be represented in a variance matrix though the plane could (by having two large eigenvalues and one small one, with its eigenvector perpendicular to the plane). Since the pairing subtracts two degrees of freedom from the rotation vector, the Kalman filter produces a variance which has only one large eigenvalue and a mean which is near the point on the curve closest to the initial guess. This updated mean may not be very good.

The covariance representation is clearly inadequate — the straight-line distribution representing the one degree of freedom must model an entire cardioid. However, we note that for the single model and data vector constraint, the locus of rotations, when represented as unit quaternions is a great circle[2] on the unit sphere. This gives us a possible way of adapting the covariance representation to encode partial constraints: Anisotropically transform quaternion space so that the great circles become straight lines in a 3D subspace. The intersection of two such lines is a fully constrained rotation estimate. Because the representation is linearized, the Kalman filter can then be used as described above. The difficulty with this method is that we have not yet discovered a transformation that will do this for an arbitrary great circle. The Gauss map works for certain circles, requiring that the representation space be rotated to favorably position the first constraint. Additionally, back-projecting the covariance matrices into the non-deformed space may only be done when the covariance matrix is full rank.

In other circumstances, at least the first two pairs of matched directions must be processed together by concatenating the observation vectors and the measurement functions [14].

One other partial rotation constraint, where the rotation axis is fixed but the rotation angle is variable, can be represented by a variance where there is a single large eigenvalue along the axis. However, this constraint does not often arise in practice.

6 Conclusions

The examples show that large variances are effective for encoding partial translation constraints, and that the Kalman filter is an effective tool for resolving the constraints to produce fully constrained pose estimates. Moreover, the pose estimates are very good, as demonstrated by the interweaving observed between the raw range data and the projected model surfaces in the illustrations (Figures 4 and 5). The method is a significant improvement over previous methods which used bounding intervals to represent uncertainty for three main reasons:

1. Many natural constraints are linear or planar in Euclidean space, but not necessarily aligned with the coordinate axes.

2. Variance-covariance matrices are good at representing linear and planar constraints.

[2] That is, a locus of unit quaternions lying in a 2D subspace of 4D quaternion space. The actual plane is $\mathit{span}\{\|\mathbf{u}+\mathbf{v}\|^2+2\mathbf{u}\times\mathbf{v}, 0+(\mathbf{u}+\mathbf{v})\}$, where quaternions are written as the sum of a scalar and a vector.

3. The covariance matrix approach allows representation of both lack of knowledge (i.e. unbound degrees of freedom), *a priori* position constraints (i.e. those known in advance of any actual observations), weak constraints (e.g. direction constraints formed by linking centres-of-mass) and statistical measurement error in the same framework.

The incrementally constrained position estimates are also useful in that the partially constrained positions can be then used to predict the location of additional image features, which might then further constrain the pose.

We have observed that best performance is achieved with good initial mean estimates, which might be a consequence of the linearization of the rotation factors in the Kalman filter. Further, the method is not able to cope with non-linear constraints such as the coupled constraints that are often generated between rotation and translation when there is only a partial rotation estimate (e.g. Section 3.3, cases 2, 4).

Future work could investigate the possibility of analysing the variance matrix to deduce which large degrees of freedom remain, and thus what type of constraints would be useful for optimally reducing the uncertainty and where to search in the image for them.

Acknowledgements

This research was funded by SERC (IED grant GR/F/38310). Other facilities provided by University of Edinburgh. This paper benefited greatly from discussions with Dibio Borges and Manuel Trucco, and Andrew Fitzgibbon helped greatly with the preparation of several of the examples.

References

[1] N. Ayache and O.D. Faugeras. Maintaining representations of the environment of a mobile robot. In *Robotics Research 4*, pages 337–350. MIT Press, USA, 1988.

[2] Y. Bar-Shalom and T.E. Fortmann. *Tracking and Data Association*. Academic Press, UK, 1988.

[3] P.J. Besl. *Surfaces in Range Image Understanding*. Springer-Verlag, 1987.

[4] R.A. Brooks. Symbolic reasoning among 3D models and 2-d images. *Artificial Intelligence Journal*, 17:285–348, 1981.

[5] G. Falk. Interpretation of imperfect line data as a three-dimensional scene. *Artificial Intelligence*, 3:101–144, 1983.

[6] O.D. Faugeras. A few steps towards artificial 3D vision. In M. Brady, editor, *Robotics Science*. MIT Press, USA, 1989.

[7] R.B. Fisher. SMS: A suggestive modeling system for object recognition. *Image and Vision Computing*, 5(2):98–104, 1987.

[8] R.B. Fisher. *From Surfaces to Objects: Computer Vision and Three Dimensional Scene Analysis*. John Wiley, UK, 1989.

[9] A. W. Fitzgibbon and R. B. Fisher. Invariant fitting of arbitrary single-extremum surfaces. In *British Machine Vision Association Conference*, pages 569–578. BMVA Press, 1993.

[10] E.C. Freuder. A computer system for visual recognition using active knowledge. In *5th International Joint Conference on Artificial Intelligence*, pages 671–677, 1977.

[11] R.A. Horn and C.R. Johnson. *Matrix Analysis*. Cambridge University Press, USA, 1985.

[12] A.H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, USA, 1970.

[13] J. J. Leonard, H.F. Durrant-Whyte, and I.J. Cox. Dynamic map building for an autonomous mobile robot. *International Journal of Robotics Research*, 11(4):286–298, 1992.

[14] R. McLachlan. Estimating 3D rotations using an iterated extended Kalman filter. Technical Report EPCC-SS92-20, Edinburgh Parallel Computing Centre, 1992.

[15] M.J.L. Orr. On composing rotations. Department of Artificial Intelligence, Edinburgh University, Working Paper 242, 1992.

[16] M.J.L. Orr and R.B. Fisher. Geometric reasoning for computer vision. *Image and Vision Computing*, 5(3):233–238, 1987.

[17] M.J.L. Orr, R.B. Fisher, and J. Hallam. Uncertain reasoning: Intervals versus probabilities. In *British Machine Vision Conference*, pages 351–354. Springer-Verlag, 1991.

[18] R.J. Popplestone, A.P. Ambler, and I.M. Bellos. An interpreter for a language describing assemblies. *Artificial Intelligence*, 14:79, 1980.

[19] J. Porril, S.B. Pollard, and J.E.W. Mayhew. Optimal combination of multiple sensors including stereo vision. *Image and Vision Computing*, 5(2):174–180, 1987.

[20] K.J. Turner. *Computer Perception of Curved Objects Using a Television Camera*. PhD thesis, Department of Artificial Intelligence, Edinburgh University, 1974.

[21] Z. Zhang and O.D. Faugeras. A 3D world model builder with a mobile robot. *International Journal of Robotics Research*, 11(4):269–285, 1992.

Appendix: Partial Derivatives for the IEKF

A: Estimating Rotations from Matched Directions

This is the problem of estimating a rotation vector, $\mathbf{r}$ (the product of the rotation axis and angle), from pairs of matched vectors, $\mathbf{u}_k$ and $\mathbf{v}_k$, such that $\mathbf{v}_k$ is the rotation (by $\mathbf{r}$) of $\mathbf{u}_k$. The state vector is $\mathbf{x} = \mathbf{r}$, the observation vectors are $\mathbf{z}_k = [\mathbf{v}_k^T \ \mathbf{u}_k^T]^T$ and the measurement equation for each observation is

$$\mathbf{f}(\mathbf{x},\ \mathbf{z}_k) \ = \ \mathbf{v}_k - \mathbf{\Phi}\,\mathbf{u}_k \ = \ \mathbf{0}$$

where

$$\begin{aligned}\mathbf{\Phi} &= \mathbf{I} + \frac{\sin\phi}{\phi}\mathbf{H} + \frac{1-\cos\phi}{\phi^2}\mathbf{H}^2 \\ \phi &= \ \| \, \mathbf{r} \, \|\end{aligned}$$

and

$$\mathbf{H} \ = \ \begin{bmatrix} 0 & -r_3 & r_2 \\ r_3 & 0 & -r_1 \\ -r_2 & r_1 & 0 \end{bmatrix} .$$

The derivatives of the measurement function are

$$\begin{aligned}\frac{\partial \mathbf{f}}{\partial \mathbf{x}} &= \left[-\frac{\partial \mathbf{\Phi}}{\partial r_1}\mathbf{u}_k \quad -\frac{\partial \mathbf{\Phi}}{\partial r_2}\mathbf{u}_k \quad -\frac{\partial \mathbf{\Phi}}{\partial r_3}\mathbf{u}_k \right] \\ \frac{\partial \mathbf{f}}{\partial \mathbf{z}_k} &= [\mathbf{I} \quad -\mathbf{\Phi}]\end{aligned}$$

where

$$\begin{aligned}\frac{\partial \mathbf{\Phi}}{\partial r_i} \ = \ & \frac{\sin\phi}{\phi}\mathbf{H}_i + \frac{r_i}{\phi^3}\left(\phi\cos\phi - sin\phi\right)\mathbf{H} + \\ & \frac{r_i}{\phi^4}\left(\phi\sin\phi - 2\left(1-\cos\phi\right)\right)\mathbf{H}^2 + \frac{1-\cos\phi}{\phi^2}\left(\mathbf{H}\mathbf{H}_i + \mathbf{H}_i\mathbf{H}\right)\end{aligned}$$

for $i = 1, 2, 3$ and the basis matrices $\mathbf{H}_i$ are given by

$$\mathbf{H}_1 \ = \ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} , \quad \mathbf{H}_2 \ = \ \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} , \quad \mathbf{H}_3 \ = \ \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} .$$

These equations have appeared before in the literature [6] but we include them here for completeness.

B: Predicting Rotated Directions

When the problem in Appendix A is turned around so that we start with an estimate of the rotation, $\mathbf{r}$, and the model direction, $\mathbf{u}_k$, and want to predict the rotated scene direction, $\mathbf{v}_k$, we must change the state to $\mathbf{x}_k = \mathbf{v}_k$ and the observation to $\mathbf{z}_k = [\mathbf{r}^T \;\; \mathbf{u}_k^T]^T$. We can keep the same measurement equation, namely

$$\mathbf{f}(\mathbf{x}_k,\ \mathbf{z}_k) \;=\; \mathbf{v}_k - \mathbf{\Phi}\,\mathbf{u}_k \;=\; \mathbf{0}$$

but the Jacobians change to

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}_k} \;=\; \mathbf{I}$$

$$\frac{\partial \mathbf{f}}{\partial \mathbf{z}_k} \;=\; \left[-\frac{\partial \mathbf{\Phi}}{\partial r_1}\mathbf{u}_k \quad -\frac{\partial \mathbf{\Phi}}{\partial r_2}\mathbf{u}_k \quad -\frac{\partial \mathbf{\Phi}}{\partial r_3}\mathbf{u}_k \quad -\mathbf{\Phi} \right] .$$

Expressions for $\mathbf{\Phi}$ and $\partial\mathbf{\Phi}/\partial r_i,\ i = 1, 2, 3$ are given in Appendix A.

C: Estimating Translation from Matched Points

This is the problem of estimating the translational component of a 3D transform from pairs of matched points, $\mathbf{p}_k$ and $\mathbf{q}_k$, such that $\mathbf{q}_k$ is the transform (by an already estimated rotation $\mathbf{r}$ and unknown translation $\mathbf{t}$) of $\mathbf{p}_k$. The state vector is $\mathbf{x} = [\mathbf{r}^T \; \mathbf{t}^T]$, the observation vectors are $\mathbf{z}_k = [\mathbf{q}_k^T \; \mathbf{p}_k^T]^T$ and the measurement equation for each observation is

$$\mathbf{f}(\mathbf{x},\ \mathbf{z}_k) \;=\; \mathbf{q}_k - \mathbf{\Phi}\,\mathbf{p}_k - \mathbf{t} \;=\; \mathbf{0}$$

where $\mathbf{\Phi}$ (a function of $\mathbf{r}$, see Appendix A) is the rotation matrix. The Jacobians are

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \;=\; \left[-\frac{\partial \mathbf{\Phi}}{\partial r_1}\mathbf{p}_k \quad -\frac{\partial \mathbf{\Phi}}{\partial r_2}\mathbf{p}_k \quad -\frac{\partial \mathbf{\Phi}}{\partial r_3}\mathbf{p}_k \quad -\mathbf{I} \right]$$

$$\frac{\partial \mathbf{f}}{\partial \mathbf{z}_k} \;=\; [\mathbf{I} \quad -\mathbf{\Phi}] \;.$$

Expressions for $\mathbf{\Phi}$ and $\partial\mathbf{\Phi}/\partial r_i,\ i = 1, 2, 3$ are given in Appendix A.

D: Predicting Transformed Points

When the problem in Appendix C is turned around so that we start with an estimate of the rotation, $\mathbf{r}$, translation, $\mathbf{t}$, and the model point, $\mathbf{p}_k$, and want to predict the transformed scene point, $\mathbf{q}_k$, we must change the state to $\mathbf{x}_k = \mathbf{q}_k$ and the observation to $\mathbf{z}_k = [\mathbf{r}^T \;\; \mathbf{t}^T \;\; \mathbf{p}_k^T]^T$. We can keep the same measurement equation, namely

$$\mathbf{f}(\mathbf{x}_k,\ \mathbf{z}_k) \;=\; \mathbf{q}_k - \mathbf{\Phi}\,\mathbf{p}_k - \mathbf{t} \;=\; \mathbf{0}$$

but the Jacobians change to

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}_k} = \mathbf{I}$$

$$\frac{\partial \mathbf{f}}{\partial \mathbf{z}_k} = \left[-\frac{\partial \mathbf{\Phi}}{\partial r_1}\mathbf{p}_k \quad -\frac{\partial \mathbf{\Phi}}{\partial r_2}\mathbf{p}_k \quad -\frac{\partial \mathbf{\Phi}}{\partial r_3}\mathbf{p}_k \quad -\mathbf{I} \quad -\mathbf{\Phi} \right] .$$

Expressions for $\mathbf{\Phi}$ and $\partial\mathbf{\Phi}/\partial r_i,\ i = 1, 2, 3$ are given in Appendix A.

E: Estimating the Composition of Two Transforms

Suppose we have estimates for the position, $\mathbf{p}_{cs}$, of a subcomponent object in the camera frame and the position, $\mathbf{p}_{ps}$, of the same subcomponent in its parent object's frame and we want to derive an estimate for the position, $\mathbf{p}_{cp}$, of the parent object in the camera frame. This problem is one of composing the estimate for $\mathbf{p}_{cs}$ with an estimate of the inverse of $\mathbf{p}_{ps}$. The state vector is $\mathbf{x} = \mathbf{p}_{cp} = [\mathbf{r}_{cp}^T \ \ \mathbf{t}_{cp}^T]^T$, the observation vector is $\mathbf{z} = [\mathbf{p}_{cs}^T \ \ \mathbf{p}_{ps}^T]^T = [\mathbf{r}_{cs}^T \ \ \mathbf{t}_{cs}^T \ \ \mathbf{r}_{ps}^T \ \ \mathbf{t}_{ps}^T]^T$, and the measurement equation is

$$\mathbf{f}(\mathbf{x},\ \mathbf{z}) = \left[\begin{array}{c} \mathbf{r}_{cp} - \mathbf{g}(\mathbf{r}_{cs},\ \mathbf{r}_{sp}) \\ \mathbf{t}_{cp} - \mathbf{t}_{cs} + \mathbf{\Phi}_{cs}\mathbf{\Phi}_{ps}^T\mathbf{t}_{ps} \end{array} \right]$$

(a 6D vector) where $\mathbf{r}_{sp} = -\mathbf{r}_{ps}$ (to invert the rotation) and $\mathbf{\Phi}_{cs}$ and $\mathbf{\Phi}_{ps}$ are rotation matrices (as given in Appendix A). The function $\mathbf{g}$ (derived in [15]) expresses rotation composition and is

$$\mathbf{g}(\mathbf{r}_{cs},\ \mathbf{r}_{sp}) = \lambda\mathbf{w}$$

where

$$\lambda = \frac{2\ \arccos\left(c_{cp}\right)}{\sqrt{1 - c_{cp}^2}}$$

$$c_{cp} = c_{cs}c_{sp} - \frac{s_{cs}s_{sp}}{\phi_{cs}\phi_{sp}}\mathbf{r}_{cs}^T\mathbf{r}_{sp}$$

$$\mathbf{w} = \frac{s_{cs}c_{sp}}{\phi_{cs}}\mathbf{r}_{cs} + \frac{s_{sp}c_{cs}}{\phi_{sp}}\mathbf{r}_{sp} + \frac{s_{cs}s_{sp}}{\phi_{cs}\phi_{sp}}\mathbf{r}_{cs} \times \mathbf{r}_{sp}$$

and where

$$\begin{aligned} c_{cs} &= \cos(\phi_{cs}/2) \\ s_{cs} &= \sin(\phi_{cs}/2) \\ c_{sp} &= \cos(\phi_{sp}/2) \\ s_{sp} &= \sin(\phi_{sp}/2) \\ \phi_{cs} &= \parallel \mathbf{r}_{cs} \parallel \\ \phi_{sp} &= \parallel \mathbf{r}_{sp} \parallel \end{aligned}$$

The Jacobians are

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \mathbf{I}$$

$$\frac{\partial \mathbf{f}}{\partial \mathbf{z}} = \left[\begin{array}{cccc} -\partial \mathbf{g}/\partial \mathbf{r}_{cs} & \mathbf{0} & \partial \mathbf{g}/\partial \mathbf{r}_{sp} & \mathbf{0} \\ (\partial \mathbf{\Phi}_{cs}/\partial \mathbf{r}_{cs})\mathbf{\Phi}_{ps}^T \mathbf{t}_{ps} & -\mathbf{I} & \mathbf{\Phi}_{cs}(\partial \mathbf{\Phi}_{ps}/\partial \mathbf{r}_{ps})^T \mathbf{t}_{ps} & \mathbf{\Phi}_{cs}\mathbf{\Phi}_{ps}^T \end{array} \right] .$$

Note that $\partial \mathbf{\Phi}/\partial \mathbf{r}$ is a tensor, not a matrix, so expressions like $(\partial \mathbf{\Phi}/\partial \mathbf{r})\mathbf{t}$ where $\mathbf{t}$ is a vector are shorthand for $[(\partial \mathbf{\Phi}/\partial r_1)\mathbf{t} \;\; (\partial \mathbf{\Phi}/\partial r_2)\mathbf{t} \;\; (\partial \mathbf{\Phi}/\partial r_3)\mathbf{t}]$. The matrices $\partial \mathbf{\Phi}/\partial r_i$, $i = 1, 2, 3$ are given in Appendix A. Expressions for $\partial \mathbf{g}/\partial \mathbf{r}_{cs}$ and $\partial \mathbf{g}/\partial \mathbf{r}_{sp}$ (see [15] for details) are

$$\begin{aligned} \frac{\partial \mathbf{g}}{\partial \mathbf{r}_{cs}} &= \mathbf{w}\frac{\partial \lambda}{\partial \mathbf{r}_{cs}} + \lambda \frac{\partial \mathbf{w}}{\partial \mathbf{r}_{cs}} \\ \frac{\partial \mathbf{g}}{\partial \mathbf{r}_{sp}} &= \mathbf{w}\frac{\partial \lambda}{\partial \mathbf{r}_{sp}} + \lambda \frac{\partial \mathbf{w}}{\partial \mathbf{r}_{sp}} \end{aligned}$$

The derivatives of λ and $\mathbf{w}$ are rather messy. Those for λ are

$$\begin{aligned} \frac{\partial \lambda}{\partial \mathbf{r}_{cs}} &= \frac{\lambda c_{cp} - 2}{1 - c_{cp}^2} \left(\frac{s_{cs} s_{sp}}{\phi_{cs}^3 \phi_{sp}} \mathbf{r}_{sp}^T \mathbf{H}_{cs}^2 - \left(\frac{s_{cs} c_{sp}}{2\phi_{cs}} + \frac{s_{sp} c_{cs} \mathbf{r}_{cs}^T \mathbf{r}_{sp}}{2\phi_{cs}^2 \phi_{sp}} \right) \mathbf{r}_{cs}^T \right) \\ \frac{\partial \lambda}{\partial \mathbf{r}_{sp}} &= \frac{\lambda c_{cp} - 2}{1 - c_{cp}^2} \left(\frac{s_{cs} s_{sp}}{\phi_{sp}^3 \phi_{cs}} \mathbf{r}_{cs}^T \mathbf{H}_{sp}^2 - \left(\frac{s_{sp} c_{cs}}{2\phi_{sp}} + \frac{s_{cs} c_{sp} \mathbf{r}_{cs}^T \mathbf{r}_{sp}}{2\phi_{sp}^2 \phi_{cs}} \right) \mathbf{r}_{sp}^T \right) \end{aligned}$$

where $\mathbf{H}_{cs}$ and $\mathbf{H}_{sp}$ are the same type of anti-symmetric matrix duals for $\mathbf{r}_{cs}$ and $\mathbf{r}_{sp}$ as $\mathbf{H}$ was for $\mathbf{r}$ in Appendix A. Finally, the derivatives of $\mathbf{w}$ are

$$\frac{\partial \mathbf{w}}{\partial \mathbf{r}_{cs}} = \left(c_{sp}\mathbf{I} - \frac{s_{sp}}{\phi_{sp}} \mathbf{H}_{sp} \right) \left(\frac{c_{cs}}{2}\mathbf{I} + \left(\frac{c_{cs}}{2\phi_{cs}^2} - \frac{s_{cs}}{\phi_{cs}^3} \right) \mathbf{H}_{cs}^2 \right) - \frac{s_{cs} s_{sp}}{2\phi_{cs}\phi_{sp}} \mathbf{r}_{sp}\mathbf{r}_{cs}^T$$

$$\frac{\partial \mathbf{w}}{\partial \mathbf{r}_{sp}} = \left(c_{cs}\mathbf{I} + \frac{s_{cs}}{\phi_{cs}} \mathbf{H}_{cs} \right) \left(\frac{c_{sp}}{2}\mathbf{I} + \left(\frac{c_{sp}}{2\phi_{sp}^2} - \frac{s_{sp}}{\phi_{sp}^3} \right) \mathbf{H}_{sp}^2 \right) - \frac{s_{cs} s_{sp}}{2\phi_{cs}\phi_{sp}} \mathbf{r}_{cs}\mathbf{r}_{sp}^T$$

F: Predicting the Composition of Two Transforms

We use the term *prediction* for problems in which we know an estimate for the position of the parent object and wish to predict from it estimates for features, such as directions, point positions or, as here, positions of subcomponent objects. The latter case is like the problem in Appendix E except here we have estimates for the position, $\mathbf{p}_{cp}$, of the parent object in the camera frame and the position, $\mathbf{p}_{ps}$, of the subcomponent in the parent object's frame and wish to derive an estimate for the position, $\mathbf{p}_{cs}$, of a subcomponent object in the camera frame. The problem is to compose the estimate of $\mathbf{p}_{cp}$ with the estimate of $\mathbf{p}_{ps}$. The state vector is $\mathbf{x} = \mathbf{p}_{cs} = [\mathbf{r}_{cs}^T \;\; \mathbf{t}_{cs}^T]^T$, the observation vector is $\mathbf{z} = [\mathbf{p}_{cp}^T \;\; \mathbf{p}_{ps}^T]^T = [\mathbf{r}_{cp}^T \;\; \mathbf{t}_{cp}^T \;\; \mathbf{r}_{ps}^T \;\; \mathbf{t}_{ps}^T]^T$, and the measurement equation is

$$\mathbf{f}(\mathbf{x},\ \mathbf{z}) \ = \ \left[\begin{array}{c} \mathbf{r}_{cs} - \mathbf{g}(\mathbf{r}_{cp},\ \mathbf{r}_{ps}) \\ \mathbf{t}_{cs} - \mathbf{t}_{cp} - \mathbf{\Phi}_{cp}\mathbf{t}_{ps} \end{array} \right]$$

(a 6D vector) where $\mathbf{\Phi}_{cp}$ is a rotation matrix (see Appendix A). The function $\mathbf{g}$ is the same as given in Appendix E with substitution of $\mathbf{r}_{cs}$ for $\mathbf{r}_{cp}$ and $\mathbf{r}_{ps}$ for $\mathbf{r}_{sp}$. Since neither of the composed positions need be inverted, the Jacobians are a little simpler than those in Appendix E, and are

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \ = \ \mathbf{I}$$

$$\frac{\partial \mathbf{f}}{\partial \mathbf{z}} \ = \ \left[\begin{array}{cccc} -\partial \mathbf{g}/\partial \mathbf{r}_{cp} & \mathbf{0} & -\partial \mathbf{g}/\partial \mathbf{r}_{ps} & \mathbf{0} \\ -(\partial \mathbf{\Phi}_{cp}/\partial \mathbf{r}_{cp})\mathbf{t}_{ps} & -\mathbf{I} & \mathbf{0} & -\mathbf{\Phi}_{cp} \end{array} \right]$$

The derivatives $\partial \mathbf{g}/\partial \mathbf{r}_{cp}$ and $\partial \mathbf{g}/\partial \mathbf{r}_{ps}$ are identical to those in Appendix E after swapping indices *cs* and *cp* and substituting *ps* for *sp*. Derivatives for rotation matrices with respect to their rotation vector components are given in Appendix A.

G: Implementation Details

The notation used here is:

$\vec{n}_d^{(i)}$	ith data patch direction (e.g. normal or axis)
$\vec{n}_m^{(i)}$	ith model patch direction (e.g. normal or axis)
$\vec{C}_d^{(i)}$	ith data patch centroid
$\vec{C}_m^{(i)}$	ith model patch centroid

Let $\mathbf{M}$ be a 3×3 matrix accumulating model evidence, and $\mathbf{D}$ be a 3×3 matrix accumulating data evidence; both are used in a generalised least-squares estimation process. Let ω_i be a weight factor reflecting the confidence in the evidence. Let S be the set of translation constraints.

Add *a priori* constraints

If the constraint is a model plane lying flush against a known scene plane (e.g. the object's base is lying on a given surface), then the planar surface constraints given in the next subsection are used, where the known scene plane is used for the data surface.

If the constraint is a model cylindrical surface is fitting a known scene cylindrical surface (e.g. a cylindrical peg is in a hole), then the cylindrical surface constraints of the second section following this are used, where the known scene cylindrical surface is used for the data surface.

Similar constraints can be developed if the *a priori* known contact is a single point or an edge (which we treat as if it were a single point).

Add planar surface rotation and translation constraints

The plane surface normals $\vec{n}_m^{(i)}$ and $\vec{n}_d^{(i)}$ are used to form the constraints.

Add $\vec{n}_m^{(i)}(\vec{n}_d^{(i)})^T\omega_i$ to $\mathbf{M}$.

Add $\vec{n}_d^{(i)}(\vec{n}_d^{(i)})^T\omega_i$ to $\mathbf{D}$.

Add $\{[\vec{n}_m^{(i)}, \vec{n}_m^{(i)}(\vec{C}_m^{(i)})^T\omega_i], [\vec{n}_d^{(i)}, \vec{n}_d^{(i)}(\vec{C}_d^{(i)})^T\omega_i]\}$ to S.

Here, $\omega_i = \omega_{plane} = 1.0$.

Add cylindrical surface rotation constraints

Here, the cylinder axes $\vec{n}_m^{(i)}$ and $\vec{n}_d^{(i)}$ are used to form the constraints.

As there is no preferential direction to the cylinder axis, there are two possibilities. The two cases are handled at a higher level by splitting the hypotheses into two and using either the model cylinder axis or its negative. Here, we assume the original axis is used

Add $\vec{n}_m^{(i)}(\vec{n}_d^{(i)})^T\omega_i$ to $\mathbf{M}$.

Here, $\omega_i = \omega_{cyl} = 10^{-5}$, which reflects the less reliable constraints obtained from the estimates of cylindrical patch axes.

Add subcomponent constraints

For simplicity, we assume that the subcomponent is expressed in the same coordinate system as the model (i.e. a null transformation between the coordinate systems). For full generality, the model subcomponent evidence needs to be transformed into the full model's coordinate system. In addition, there may be multiple instances of the subcomponent in the object model, and the subcomponent may have axes of symmetry. This entails creating duplicate hypotheses and merging the constraints in the different permutations (most of which will be eliminated because of inconsistency). These complexities are ignored here.

Add the subcomponent's $\mathbf{M}$ to object's $\mathbf{M}$.

Add the subcomponent's $\mathbf{D}$ to object's $\mathbf{D}$.

Append the subcomponent's S to object's S.

If rotation still underconstrained

If the rotation is still underconstrained after the subcomponent, plane normal and cylinder axis constraints, then we form weaker rotation constraints by making paired direction vectors between the centroids of patches.

Given a pair of unoccluded model patches j and k, the model direction $\vec{n}_m^{(i)}$ is given by

$$\vec{n}_m^{(i)} = \frac{\vec{C}_m^{(j)} - \vec{C}_m^{(k)}}{\| \vec{C}_m^{(j)} - \vec{C}_m^{(k)} \|}$$

A similar calculation gives the corresponding data direction. The occlusion of model patches is recorded explicitly in the view group information in the models [7] and for data patches can be determined from depth discontinuity boundaries in the data for data patches.

Here, $\omega_i = \omega_{cent} = 10^{-5}$.

If translation still underconstrained

If the translation is still underconstrained after the *a priori*, subcomponent and plane centroid constraints, then we form weaker constraints by using a point along the axis of the cylindrical patches.

At this point, we have a fully constrained rotation $\mathbf{R}$. The technique we use is to express the cylinder axis as the intersection of two arbitrary planes, and then add two plane translation constraints to $\mathbf{D}$ and S.

If $\vec{P}_d^{(i)}$ is a point on the data axis and $\vec{P}_m^{(i)}$ is a point on the model axis, the constraints added to S are:

$$\{[\vec{n}_{mk}^{(i)}, \vec{n}_{mk}^{(i)}(\vec{P}_m^{(i)})^T\omega_i], [\vec{n}_{dk}^{(i)}, \vec{n}_{dk}^{(i)}(\vec{P}_d^{(i)})^T\omega_i]\}$$

for $k = 1, 2$.

Merge constraints to get mean pose

Let $\mathbf{U\Lambda V^T}$ be the singular value decomposition (see [11], page 431) of $\mathbf{M}$. Then, the mean rotation $\mathbf{R}$ is given by $\mathbf{UV^T}$. If there are less than three independent direction constraints on the rotation, then it is possible that $\mathbf{R}$ also includes a reflection. Thus, if $|\mathbf{R}| < 0$, then the column of $\mathbf{V}$ corresponding to the smallest diagonal element of $\mathbf{\Lambda}$ is negated.

If the rank of $\mathbf{D}$ is less than 3, we add additional translation constraints using patch centroids in the direction of the null space of $\mathbf{D}$.

Thereafter, if $S = \{[\vec{n}_m^{(i)}, d_m^{(i)}], [\vec{n}_d^{(i)}, d_d^{(i)}]\}$ then the mean translation is given by:

$$\vec{t} = \mathbf{D^{-1}}\Sigma_i \vec{n}_d^{(i)}(d_d^{(i)} - d_m^{(i)})$$

where the sum is over all constraints in S.

Use Kalman filter to refine and get pose variance

Here, we give the actual covariance matrices used to model the data and model input vector and point position covariances. The means are calculated using the methods described above. Vector directions have the covariance matrix $\mathbf{P}diag(s, s, c)\mathbf{P^T}$, where

$$s = \int_0^{\pi/2} sin^2(x)e^{-\frac{x^2}{2\sigma^2}}dx$$

$$c = \int_0^{\pi/2} cos^2(x) e^{-\frac{x^2}{2\sigma^2}} dx$$

and $\mathbf{P}$ rotates the vector $(0,0,1)$ onto the direction vector. This transforms the angular variance σ about the direction vector into a cartesian variance.

For model planes, $\sigma = 10^{-3}$ radians. For data planes, $\sigma = 0.01$ radians. For model cylinder axes, $\sigma = 10^{-3}$ radians. For data cylinder axes, $\sigma = 0.2$ radians.

Point positions have the covariance matrix $\mathbf{P} diag(r,s,t) \mathbf{P^T}$, where $\mathbf{P}$ rotates the vector $(0,0,1)$ onto the surface normal or cylinder axis in the case of model features and $\mathbf{P} = \mathbf{I}$ for data features.

For model planes, $(r,s,t) = (100(x\_extent)^2, 100(y\_extent)^2, 0.1)$. For data planes, $(r,s,t) = (\sigma_p, \sigma_p, \sigma_p)$ (here, $\sigma_p = 3$mm). For model cylinders, $(r,s,t) = (\sigma_c, \sigma_c, \infty)$ (here, $\sigma_c = 1$mm). For data cylinders, $(r,s,t) = (\sigma_c, \sigma_c, \infty)$ (here, $\sigma_c = \frac{radius}{10}$ mm).

With the initial mean values estimated in the previous sections, and the covariance estimates constructed here and in Appendices A–F, the IEKF is used to estimate mean values and covariances of the desired state vectors.