$3D$ Distance Metric for Pose Estimation and Object Recognition from $2D$ Projections

Yacov Hel-Or

The Weizmann Institute of Science
Dept. of Applied Mathematics and Computer Science
Rehovot 76100, ISRAEL
email:toky@wisdom.weizmann.ac.il

Abstract

Model based object recognition and model based pose estimation require a distance metric to find the optimal pose and to measure the distance between the measurements and possible models during the recognition process. When the measurements are given in 2D (such as in orthographic and perspective projections) the commonly used distance between the $3D$ model features and the $2D$ image features is the $2D$ Euclidean distance measured in the image plane. However, this $2D$ distance does *not*, usually, increase monotonically with the real $3D$ distance and thus does not really represent the distance being measured. In this paper we propose a new scheme in which both the optimal positioning and the evaluation of similarity between the $2D$ image and the model is performed relative to the $3D$ distance. This distance is calculated between the model features and a $3D$ *predicted object* which is a permissible reconstruction of the measured object and is the "closest" to the model features.

1 Introduction

Model based object recognition and model based pose estimation are two complementary problems that arise frequently in the vision literature (for example [8, 9, 5, 4, 12, 6]). In model-based pose estimation the position in $3D$ space (translation + orientation) of a known object is determined from the object measurements. In model-based recognition a measured object is compared to a library of prototype models in order to find the model which is the "closest" to the viewed object. Commonly, this process requires the estimation of the best model positioning followed by a measure of similarity between the model and the measured data. Both the recognition process and the pose estimation process, use some distance metric to find the optimal pose and later to measure the distance between the measurements and the hypothesis model.

Many applications use feature points such as maximum curvature, segment endpoints and corners as a model description. In these applications, when measurements are given in $3D$ (such as range finder or stereo data), Euclidean distance between the $3D$ model features and the corresponding $3D$ measured features is calculated to guide the optimal pose estimation and the recognition process [11, 8, 5, 2]. If $\{\mathbf{X}'_i\}_{i=1\ldots n}$ and $\{\mathbf{X}_i\}_{i=1\ldots n}$ are two sets of $3D$ vectors representing the locations of the model-points and the measured points respectively, then the $3D$ Euclidean distance between these two sets is:

$$D_3 = \sum_i \|T(\mathbf{X}'_i) - \mathbf{X}_i\|^2$$

where T is the rigid transformation representing the model pose. This distance metric is reasonable since it describes the amount of $3D$ distortion a model has to be undergo in order to fit the measurements. In model-based pose estimation we aim to minimize this distance subject to the transformation T and in recognition tasks the minimization is performed subject to T and the model $\{\mathbf{X}'_i\}$.

More frequently, recognition tasks deal with measured features given in $2D$ (projected images). In this case a commonly used distance between the $3D$ model features and the $2D$ image features is the $2D$ Euclidean distance measured in the image plane. If $\{\mathbf{m}_i\}_{i=1\ldots n}$ are $2D$ vectors representing the locations of n corresponding measurements then the $2D$ Euclidean distance is defined as:

$$D_2 = \sum_i \|\Pi T(\mathbf{X}'_i) - \mathbf{m}_i\|^2$$

where Π is a projection operator. In this case, the measured features in the image are compared with the model features projected onto the image plane. The minimization of this $2D$ Euclidean distance is used to guide the positioning and the recognition processes [12, 3, 14, 15, 7]. However, the problem with this $2D$ distance is that in most cases, this distance does *not* increase monotonically with the real $3D$ distance and thus does not really represent the $3D$ distortions the model must undergo in order to fit the measured object. This inadequacy of the $2D$ distance causes imprecisions in the recognition and positioning especially when dealing with perspective projection. For example, in perspective images, a $2D$ distance between a measured point and its associated projected model point can have varying values in the real $3D$ distance, according to the depth of these points in $3D$. Therefore, the relative contribution of this pair in the total $2D$ distance may differ from its contribution in the total $3D$ distance. The same kind of problem arises when dealing with perturbations in the measurements or in the model features. These perturbations influence differently the $2D$ distance and the $3D$ distance.

In this paper we propose a new scheme which, given a $2D$ image and a candidate model, calculates the optimal positioning of the model and evaluates the "similarity" between the $2D$ image and the model. Both, the optimal positioning and the evaluation of similarity is performed relative to the $3D$ distance. This distance is calculated between the model features and a $3D$ *predicted object* which is a permissible reconstruction of the measured object. More precisely, we calculate the minimum distortion of the model which produces a permissible reconstruction of the measured features, where a reconstructed object is considered permissible if it does not contradict the measured data. The permissible reconstructed object which is the "closest" to the candidate model (in terms of minimum distortion) is the predicted object.

In this paper we describe a method that deals with recognition and pose-estimation tasks based on $3D$ distance metric. In this scheme we include uncertainty both, in the measurements and in the model features. We show that the superiority of our scheme is mainly when the uncertainty in the model is significant. This characteristic becomes important when modeling non-rigid objects or when dealing with uncertain models which are used to improve recognition of a particular instance of a class defined by a general model.

The rest of this paper is organized as follows: Sections 1 and 2 propose the general framework for dealing with noise free measurements and precise model definition. Sections

3,4, and 5 extend the proposed framework to deal with uncertain models and noisy measurements. Sections 6 and 7 present the process for pose estimation and recognition. Some simulated results are given in Section 8.

2 Definitions

A *model* of a $3D$ object is represented by a set of points:

$$\{\mathbf{X'}_i\}_{i=1\cdots n} \quad ,$$

where $\mathbf{X'}_i = (x'_i, y'_i, z'_i)^t$ is associated with the location of the i^{th} model point and represented in an object-centered frame of reference (x', y', z').

A *transformed model* is the coordinates of the model points as given relative to the viewer-centered frame of reference (x, y, z). If T denotes the rigid transformation between the object-centered frame and the viewer-centered frame then the transformed model is described by the collection: $\{\mathbf{X}_i\}_{i=1\cdots n}$, where $\mathbf{X}_i = (x_i, y_i, z_i)^t = T(\mathbf{X'}_i)$.

A *measurement* of a $3D$ object is a set:

$$\{\mathbf{m}_i\}_{i=1\cdots n} \ ,$$

where $\mathbf{m}_i$ represents a measurement of the i^{th} feature point of the object. This paper deals with the case where the measurements are obtained from a projection of the object onto a $2D$ image plane. Thus, $\mathbf{m}_i = (v_i, w_i)^t$ is represented in the image frame of reference (v, w).

A *predicted object* is a set of $3D$ coordinates of the form:

$$\{\mathbf{U}_i\}_{i=1\cdots n} \quad ,$$

where $\mathbf{U}_i$ is an estimate of the i^{th} feature point of the object and represented in the viewer-centered frame of reference (see Figure 1). The predicted object is the "closest" object to any transformed model satisfying the image constraints. Formally, we choose such an object $\{\mathbf{U}_i\}$ which minimizes the following quantity:

$$C = \sum_{i}^{n} \|\mathbf{T}(\mathbf{X'}_i) - \mathbf{U}_i\|^2 \tag{1}$$

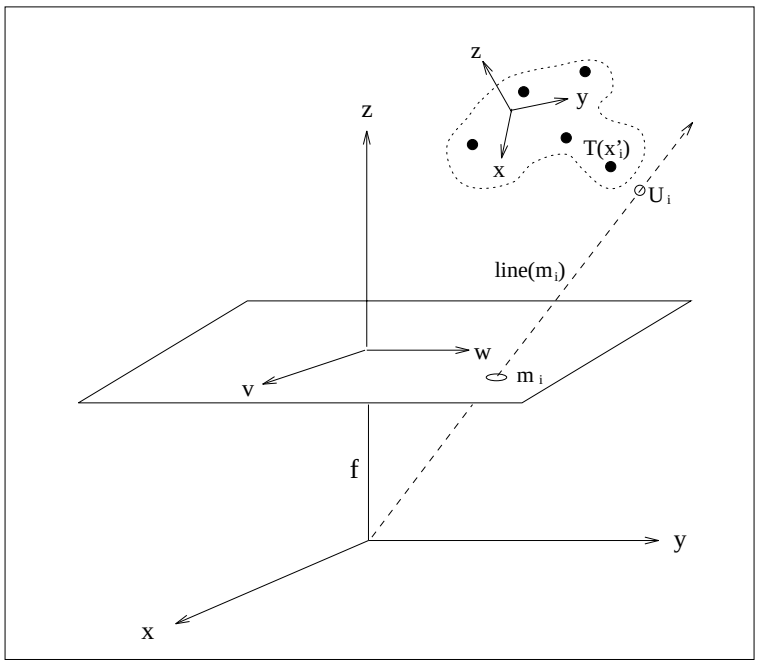

Figure 1: General configuration of a perspective projection. The frames (x', y', z'), (x, y, z) and (v, w) represent the object-centered, the viewer-centered and the image frame of reference, respectively.

under the set of constraints:

$$project(\mathbf{U}_i) = \mathbf{m}_i \quad \text{for each} \quad i = 1 \cdots n \quad , \tag{2}$$

where $project(\mathbf{U}_i)$ is the projection of $\mathbf{U}_i$ onto the image plane and where the transformation T is any rigid transformation.

3 The Predicted Object and the Optimal Transformation

Minimization of C in Equation 1 requires evaluation of both, the optimal transformation $\mathbf{T}$ and the "closest" predicted object $\{\mathbf{U}_i\}_{i=1\cdots n}$. However, knowing the optimal transformation, it is straightforward to find the predicted object. In the following we elaborate on finding the optimal transformation $\mathbf{T}$ with respect to *any* predicted object.

Denote by $line(\mathbf{m}_i)$ the collection of points in $3D$ space where the feature point $\mathbf{U}_i$ can

be located i.e. $line(\mathbf{m}_i) = \{\mathbf{V} \mid project(\mathbf{V}) = \mathbf{m}_i\}$. If the measurement is an orthographic projection onto the plane $z = 0$ then $line(\mathbf{m}_i)$ is a line parallel to the z axis and passing through the point $(\mathbf{m}_i^t, 0)^t$. If the measurement is a perspective projection, as depicted in Figure 1, then this line passes through the focal point $(0,0,0)$ and the point $(\mathbf{m}_i^t, f)$ where f is the focal length.
If $\mathbf{Q}$ is a point in $3D$ we denote by $\|\mathbf{Q} - line(\mathbf{m}_i)\|^2$ the squared Euclidean distance between $line(\mathbf{m}_i)$ and the point Q.

Lemma:
If $\{\mathbf{U}_i\}$ is the optimal predicted object minimizing Equation 1, then:

$$\|\mathbf{T}(\mathbf{X'}_i) - \mathbf{U}_i\|^2 = \|\mathbf{T}(\mathbf{X'}_i) - line(\mathbf{m}_i)\|^2 \qquad \text{for each} \quad i = 1 \cdots n \quad .$$

Proof:
By definition

$$\|\mathbf{T}(\mathbf{X'}_i) - line(\mathbf{m}_i)\|^2 = \min_{\mathbf{V} \in line(\mathbf{m}_i)} \|\mathbf{T}(\mathbf{X'}_i) - \mathbf{V}\|^2$$

However, due to the image constraint $\mathbf{U}_i \in line(\mathbf{m}_i)$. Since $\mathbf{U}_i$ is chosen s.t. $\|\mathbf{T}(\mathbf{X'}_i) - \mathbf{U}_i\|^2$ will be minimized, we have:

$$\|\mathbf{T}(\mathbf{X'}_i) - \mathbf{U}_i\|^2 = \min_{\mathbf{V} \in line(\mathbf{m}_i)} \|\mathbf{T}(\mathbf{X'}_i) - \mathbf{V}\|^2$$

and thus $\|\mathbf{T}(\mathbf{X'}_i) - \mathbf{U}_i\|^2 = \|\mathbf{T}(\mathbf{X'}_i) - line(\mathbf{m}_i)\|^2$. $\square$

From the lemma it is clear that the optimal transformation $\hat{T}$ can be derived as follows:

$$\hat{T} = \arg\left\{ \min_T \sum_i^n \|T(\mathbf{X'}_i) - line(\mathbf{m}_i)\|^2 \right\}$$

After $\hat{T}$ is calculated, the second stage is to find the predicted object. The location of each feature point is chosen so that Equation 1 is minimized, thus:

$$\mathbf{U}_i = \arg\left\{ \min_{\mathbf{V} \in line(\mathbf{m}_i)} \|\hat{T}(\mathbf{X'}_i) - \mathbf{V}\|^2 \right\} \quad \text{for each} \quad i = 1 \cdots n \quad .$$

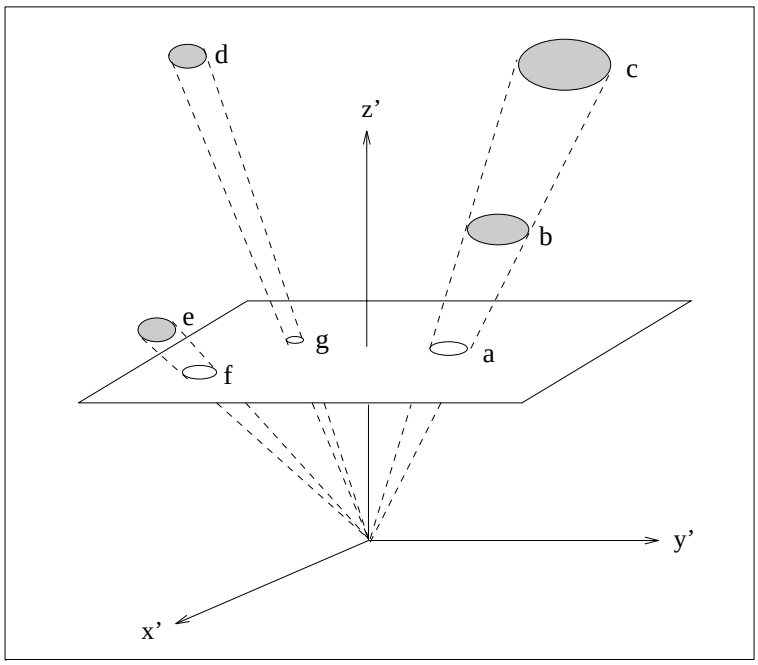

Figure 2: Perspective projection of noise. Similar noise in the model points (d and e) are projected differently onto the image plane (f and g), depending on the distance from the focal point. Conversely: noise in the image plane (a) can have a varying influence on the predicted object, according to the depth of the predicted point (b and c).

4 Uncertainty in the Model and in the Measurements

In the previous section no knowledge about the measurement uncertainties and the model uncertainties was assumed. Since we work in a noisy environment, it is important to characterize these uncertainties especially when the measurements are taken following a perspective projection. With this kind of projection, similar noise in the model points (d and e in Figure 2) are projected differently onto the image plane (f and g in Figure 2), depending on the distance from the focal point. And conversely: noise in the image plane (a in Figure 2) can have a varying influence on the predicted object, according to the depth of the predicted point (b and c in Figure 2).

In this section we consider measurements and model points which are associated with some uncertainty. That is, each measurement is now represented by a pair:

$$measurement(i) = (\hat{\mathbf{m}}_i, \Lambda_i)$$

where $\hat{\mathbf{m}}_i$ is the actual measurement of a real value $\mathbf{m}_i$ s.t. $\hat{\mathbf{m}}_i = \mathbf{m}_i + \varepsilon_i$. We assume that the noise term ε_i is of zero mean and its covariance matrix Λ_i is known. The covariance matrix depicts the uncertainty in the actual measurement and is mainly due to two factors:

- Uncertainty due to measurement noise (e.g. digitization, blurring and chromatic aberrations).
- Uncertainty dependent upon the feature detection process. For example, a detected end-point of a line segment will have low positional uncertainty in the direction perpendicular to the line segment and a high uncertainty in its direction.

Similar to the measurements, we associate for each model-point a covariance matrix Σ_i so that each one of the model points is represented by a pair:

$$model\text{-}point(i) = (\hat{\mathbf{X}}'_i, \Sigma_i)$$

where $\hat{\mathbf{X}}'_i$ is an estimated location of the i^{th} model point with a 3×3 covariance matrix Σ_i. This covariance matrix denotes the uncertainty of the estimated location $\hat{\mathbf{X}}'_i$ that may arise from three sources:

- Uncertainty due to imprecise modeling of the $3D$ object.
- Uncertainty due to modeling a class of $3D$ objects, for example, modeling a general face.
- Uncertainty due to modeling non rigid objects such as rubber objects.

In some cases there is no knowledge about the noise in the measurements or in the model. In such cases we associate an identical covariance matrix with each measurement or model-point such that $\Lambda_i = \alpha I$ or $\Sigma_i = \beta I$ where I is the identity matrix and α, β are some scalars.

When dealing with an uncertain model and noisy measurements it is clear that it is erroneous to find a predicted object which minimizes C of Equation 1. In this equation all terms added in the right-hand side influence the solution equally where in our case the

influence should be inversely proportional to the uncertainty of each term. Therefore the predicted object has to be calculated subject to a Mahalanobis distance:

$$C = \sum_{i}^{n} [(T(\mathbf{X}'_i) - \mathbf{U}_i)^t W_i^{-1} (T(\mathbf{X}'_i) - \mathbf{U}_i)] \tag{3}$$

where W_i is a 3×3 covariance representing the uncertainty of $(T(\mathbf{X}'_i) - \mathbf{U}_i)$. This uncertainty can be deduced from Λ_i and Σ_i as will be explained later on (Section 7.1). It should be noted that even if we don't have any *a priori* knowledge about the uncertainty in the model and in the measurements so that all Λ_i are identical and all Σ_i are identical as well, the term C in Equation 1 is incorrect and the transformation T should be found subject to minimizing Equation 3. This is true since W_i in Equation 3 depends also on the depth of each point from the image plane as can be seen in Figure 2.

5 The Mahalanobis Distance to an Uncertain Line

According to the lemma given in Section 3, $\{\mathbf{U}_i\}$ in Equation 3 can be replaced by $\{line(\mathbf{m}_i)\}$. However, in this case, these lines are uncertain since they are deduced from uncertain measurements. Therefore, we are actually interested in the Mahalanobis distance (M.D.) between the transformed model to these uncertain lines. This section elaborates the representation of an uncertain line and the measure of the M.D. to it.

Let $\mathbf{u}$ be a 3 dimensional vector of random variables where $E\{\mathbf{u}\} = \hat{\mathbf{u}}$ and $\mathrm{var}\{\mathbf{u}\} = \Sigma_{\mathbf{u}}$. $E\{\mathbf{u}\}$ denotes the expectation value of $\mathbf{u}$ and $\mathrm{var}\{\mathbf{u}\} = E\{(\mathbf{u} - \hat{\mathbf{u}})^t (\mathbf{u} - \hat{\mathbf{u}})\}$ denotes its 3×3 covariance matrix. If $\mathbf{p}$ is some point in $3D$ space (with zero uncertainty) then the squared Mahalanobis distance between $\mathbf{p}$ and $\hat{\mathbf{u}}$ is defined as:

$$d^2 = (\hat{\mathbf{u}} - \mathbf{p})^t \Sigma_{\mathbf{u}}^{-1} (\hat{\mathbf{u}} - \mathbf{p})$$

Assume the covariance matrix $\Sigma_{\mathbf{u}}$ is diagonal, thus there is no correlation between the components of $\mathbf{u}$:

$$\Sigma_{\mathbf{u}} = \begin{pmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_z^2 \end{pmatrix}$$

If we set σ_z^2 to be an infinite value then the contours of constant M.D. from $\hat{\mathbf{u}}$ will be elliptic cylinders parallel to the z axis centered at $\hat{\mathbf{u}}$ (See Figure 3). The cross section of these

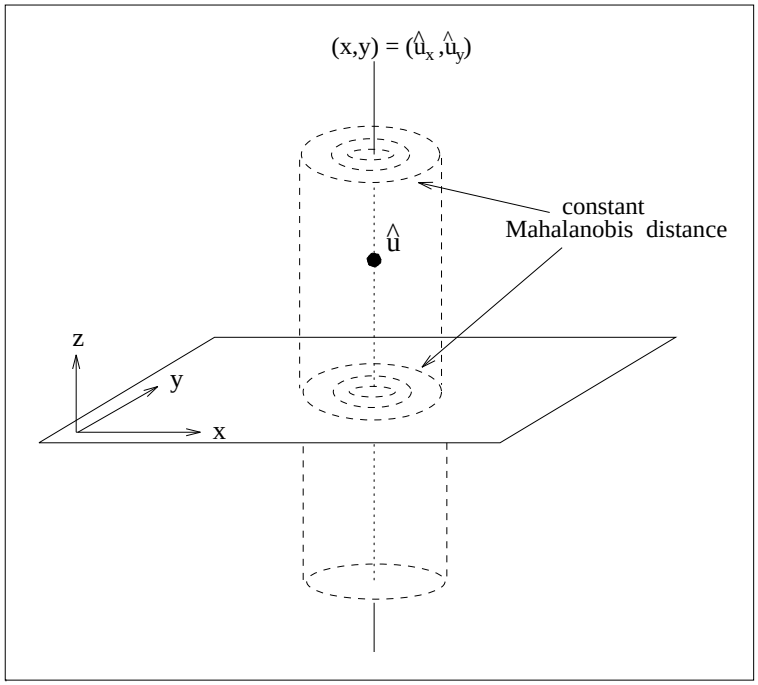

Figure 3: The Mahalanobis Distance to an uncertain line. When the uncertain line is parallel to the z axis (i.e. the diagonal covariance matrix $\Sigma_{\mathbf{u}}$ associated with the line has an infinite value for σ_z^2) then the contours of constant Mahalanobis Distance will be elliptic cylinders parallel to the z axis. The cross section of these cylinders along a plane parallel to the (x, y) plane is composed of enclosed ellipses having length and breadth proportional to σ_x^2 and σ_y^2 in the principal directions.

cylinders along a plane parallel to the (x, y) plane is composed of enclosed ellipses having length and breadth proportional to σ_x^2 and σ_y^2 in the principal directions. Therefore, in this case, the M.D. between a point $\mathbf{p}$ and $\hat{\mathbf{u}}$ depends on the distance and direction of $\mathbf{p}$ from a line perpendicular to the (x, y) plane and passing through the point $\hat{\mathbf{u}}$. Explicitly this distance is given by:

$$d^2 = (\hat{\mathbf{u}} - \mathbf{p})^t \begin{pmatrix} \frac{1}{\sigma_x^2} & 0 & 0 \\ 0 & \frac{1}{\sigma_y^2} & 0 \\ 0 & 0 & 0 \end{pmatrix} (\hat{\mathbf{u}} - \mathbf{p}) = \frac{(\hat{\mathbf{u}}_x - \mathbf{p}_x)^2}{\sigma_x^2} + \frac{(\hat{\mathbf{u}}_y - \mathbf{p}_y)^2}{\sigma_y^2} \tag{4}$$

We can view this distance as a M.D. from an uncertain line with some uncertainty in the directions perpendicular to the line direction. This uncertainty is given by the covariance matrix:

$$\Sigma_{line} = \begin{pmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{pmatrix} \quad .$$

Note, that the M.D. between $\mathbf{p}$ to *any* point on this line is equal to the M.D. between $\mathbf{p}$ and $\hat{\mathbf{u}}$ as can be seen from Equation 4. This proposition is not limited to lines parallel to one of the axes and is valid for any uncertain line. For example, consider an uncertain line passing through point $\hat{\mathbf{v}}$ with direction represented by the unit vector $R\hat{\mathbf{z}}$ where R is some rotation matrix and $\hat{\mathbf{z}}$ is a unit vector aligned with the z axis. Assuming the uncertainty of this line is the same as the previous one, this line can be represented by an uncertain point $(\hat{\mathbf{v}}, \Sigma_{\mathbf{v}})$ where its covariance matrix is as follows:

$$\Sigma_{\mathbf{v}} = R \begin{pmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \infty \end{pmatrix} R^t$$

The M.D. between point $\mathbf{p}$ to any point on this line is in fact the M.D. from $\mathbf{p}$ to the uncertain point $\hat{\mathbf{v}}$.

In the following sections we show how to use this idea to find the optimal transformation of a model and to find the predicted object closest to the model in the sense of $3D$ Euclidean distance. The method described below fuses the information from all the measured points and estimates the model transformation T by incremental refinements using $Kalman\text{-}filter$ [13, 16]. At each step a new uncertain line is generated from the associated measurement and an updated solution is produced.

6 Converting the Measurements to Uncertain Lines

As stated in Section 3 the optimal transformation is calculated subject to minimizing the distance between the transformed model and the lines $\{line(\hat{\mathbf{m}}_i)\}$. These lines are in fact uncertain lines due to the uncertainty in the measurements. Each of these lines can be represented by a $3D$ point $\hat{\mathbf{M}}_i$ located on $line(\hat{\mathbf{m}}_i)$ and having a covariance matrix Γ_i. The point uncertainty in the direction of projection is infinite, and its uncertainty in the perpendicular directions is deduced from the associated measurement uncertainty. This uncertain $3D$ point (uncertain line), $(\hat{\mathbf{M}}_i, \Gamma_i)$, can be considered as a $3D$ measurement of the i^{th} transformed model-point $T(\hat{\mathbf{X}}'_i)$. In this section we explain how to convert the measurements into uncertain lines and in the following section we give the algorithm for finding the optimal transformation using this information.

Assume that the measurements are performed on the image plane using the coordinate system (v, w) where the k^{th} measured point is:

$$measurement(k) = [\hat{\mathbf{m}}_k = (\hat{v}_k, \hat{w}_k), \Lambda^k_{vw}] \quad ,$$

Λ^k_{vw} is a 2×2 covariance matrix describing the uncertainty of the actual measurement $(\hat{v}_k, \hat{w}_k)$. For simplicity we ommit the subscript k at this time so that this measurement is represented by the pair $[(\hat{v}, \hat{w}), \Lambda_{vw}]$. We separate our discussion into two cases: orthographic projection and perspective projection.

Orthographic projection:

In the case where the projection is along the z-axis (orthographic) the associated uncertain line is represented as:

$$(\hat{\mathbf{M}}, \Gamma) = [(\hat{v}, \hat{w}, \hat{z}), \begin{pmatrix} \Lambda_{vw} & 0 \\ 0 & \infty \end{pmatrix}] \quad ,$$

where $\hat{z}$ is any estimate of the z coordinate.

Perspective projection:

In the case of perspective projection, the modeling of the uncertainty is more complex. Assume that the origin of the viewer-centered frame of reference (x, y, z) is at the focal point as shown in Figure 4 and the focal length is equal to one. We aim to transform the measurement $\hat{\mathbf{m}} = (\hat{v}, \hat{w})$ given in the image-plane coordinate system (v, w) into a representation in the Cartesian system (x, y, z).

Considering the spherical coordinate system (r, ϕ, θ) (Figure 4). The vector $(\hat{v}, \hat{w})$ determines the angular coordinates (ϕ, θ) but leaves the value of r undetermined:

$$\begin{aligned} \hat{\phi} &= \arctan(\sqrt{\hat{v}^2 + \hat{w}^2}) \\ \hat{\theta} &= \arccos(\frac{\hat{v}}{\sqrt{\hat{v}^2 + \hat{w}^2}}) \quad . \end{aligned}$$

Additionally, the uncertainty of $(\hat{v}, \hat{w})$ is transformed into a covariance matrix in the (ϕ, θ) system as follows:

$$\Lambda_{\phi\theta} = \left(\frac{\partial(\phi, \theta)}{\partial(v, w)}\right) \Lambda_{vw} \left(\frac{\partial(\phi, \theta)}{\partial(v, w)}\right)^t \quad ,$$

where $\frac{\partial(\phi,\theta)}{\partial(v,w)}$ is the Jacobian of the transform from (v, w) to (ϕ, θ), and the derivative is taken

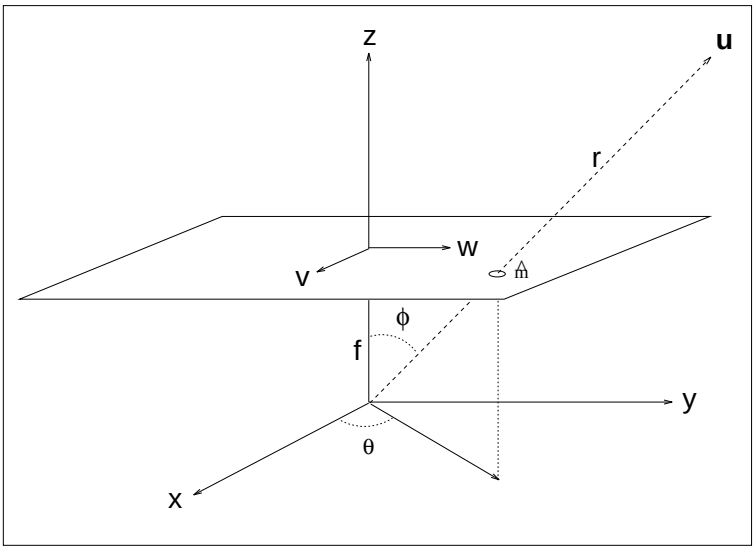

Figure 4: A perspective projection of the point **U** onto the image plane (v, w). The point can be represented in either, a Cartesian system (x, y, z) or a spherical system (r, ϕ, θ).

at point $(\hat{v}, \hat{w})$. The Jacobian matrix is:

$$\frac{\partial(\phi,\theta)}{\partial(v,w)} = \begin{pmatrix} \hat{v}\kappa^2\psi & \hat{w}\kappa^2\psi \\ -\hat{w}\psi^2 & \hat{v}\psi^2 \end{pmatrix} \quad ,$$

where

$$\begin{aligned} \kappa &\equiv \frac{1}{\sqrt{\hat{v}^2 + \hat{w}^2 + 1}} \\ \psi &\equiv \frac{1}{\sqrt{\hat{v}^2 + \hat{w}^2}} \; . \end{aligned}$$

The transformation into spherical coordinates, as an intermediary stage, allows a simple representation of the associated uncertain line:

$$(\hat{\mathbf{M}}, \Gamma) = [(\hat{r}, \hat{\phi}, \hat{\theta}), \Lambda_{r\phi\theta}] \; ,$$

where

$$\Lambda_{r\phi\theta} = \begin{pmatrix} \infty & 0 & 0 \\ 0 & & \\ 0 & & \Lambda_{\phi\theta} \end{pmatrix}$$

and $\hat{\phi}$, $\hat{\theta}$, $\Lambda_{\phi\theta}$ are the expressions described above. $\hat{r}$ is unknown but an estimation of $\hat{r}$ will be chosen as is explained later in this section.

In practice we are interested in representing the uncertain line in Cartesian coordinates, thus, the representation is transformed again from the spherical coordinates to Cartesian coordinates (x, y, z) as follows:

$$(\hat{\mathbf{M}}, \Gamma) = [(\hat{x}, \hat{y}, \hat{z}), \Lambda_{xyz}] \quad ,$$

where

$$\begin{aligned} \hat{x} &= \hat{r} \sin\hat{\phi} \cos\hat{\theta} = \hat{r}\kappa\hat{v} \\ \hat{y} &= \hat{r} \sin\hat{\phi} \sin\hat{\theta} = \hat{r}\kappa\hat{w} \\ \hat{z} &= \hat{r} \cos\hat{\phi} = \hat{r}\kappa \end{aligned}$$

and the covariance matrix is:

$$\Lambda_{xyz} = \left(\frac{\partial(x, y, z)}{\partial(r, \phi, \theta)}\right) \Lambda_{r\phi\theta} \left(\frac{\partial(x, y, z)}{\partial(r, \phi, \theta)}\right)^t \quad .$$

The Jacobian is

$$\frac{\partial(x, y, z)}{\partial(r, \phi, \theta)} = \begin{pmatrix} \hat{v}\kappa & \hat{r}\hat{v}\kappa\psi & -\hat{r}\hat{w}\kappa \\ \hat{w}\kappa & \hat{r}\hat{w}\kappa\psi & \hat{r}\hat{v}\kappa \\ \kappa & -\hat{r}\kappa/\psi & 0 \end{pmatrix} \quad ,$$

where the derivative is taken at the point $(\hat{r}, \hat{\phi}, \hat{\theta})$. Here too, all values are known except for $\hat{r}$. Since the process that estimates the optimal transformation incrementally improves the estimation of T, i.e, at each step k, there exists an estimate $\hat{T}^{k-1}$ from the previous step, we use this estimate to calculate an estimate of $\hat{r}$ at step k as follows:

$$\hat{r}^k = \|\hat{T}^{k-1}(\hat{\mathbf{X}}')\| \quad ,$$

where $\hat{\mathbf{X}}'$ is the location of the corresponding point in the model. We emphasize that the uncertainty of this estimate, as expressed in the covariance matrix, is infinite.

7 Estimation of the Optimal Transformation

As stated, the first step in finding the predicted object is to estimate a transformation $\mathbf{T}$ which optimally translates the points $\{(\hat{\mathbf{X}}'_k, \Sigma_k)\}$ of the model onto the corresponding uncertain lines $\{(\hat{\mathbf{M}}_k, \Gamma_k)\}$, generated from the measurements $\{(\hat{\mathbf{m}}_k, \Lambda_k)\}$. The optimality of this transformation is in the sense of minimizing Equation 3. The transformation $\mathbf{T}$ is a

vector representing a rigid 3-D transformation (rotation + orientation) of the model from its local coordinates to the viewer coordinates. $\mathbf{T}$ is estimated from the generated uncertain lines using the Kalman filter tools (K.F) . The estimation process is composed of an incremental refinement, for which at each step $k-1$, there exists an estimate $\hat{\mathbf{T}}^{k-1}$ of the transformation $\mathbf{T}$ and a covariance matrix Ω^{k-1} which represents the "quality" of this estimate:

$$\Omega^{k-1} = E\{(\hat{\mathbf{T}}^{k-1} - \mathbf{T})(\hat{\mathbf{T}}^{k-1} - \mathbf{T})^t\} \quad .$$

Given a new measurement $(\hat{\mathbf{m}}_k, \Lambda_k)$, an associate uncertain line, $\{(\hat{\mathbf{M}}_k, \Gamma_k)\}$, is generated, and the current estimate is updated to be $\hat{\mathbf{T}}^k$ with an associated uncertainty Ω^k. The accuracy of the estimate increases, as additional measurements are fused, i.e. $\Omega^k \leq \Omega^{k-1}$ ($\Omega^{k-1} - \Omega^k$ is nonnegative definite). The process terminates as soon as no additional measurements can be supplied.

7.1 The Kalman Filter for Parameter Estimation

The Kalman filter is a tool for parameter estimation from given measurements. In our case the parameter vector to be estimated is the transformation vector $\mathbf{T}$ which is composed of two components:

- The translation component, expressed by the vector t;

$$\mathbf{t} = (t_x, t_y, t_z)^t \quad . \tag{5}$$

- The rotation component, described by the quaternion $\tilde{\mathbf{q}}$ (see Appendix A) [10]:

$$\tilde{\mathbf{q}} = (q_0, \mathbf{q}) = (q_0, q_1 i + q_2 j + q_3 k) \quad .$$

The rotation quaternion should satisfy the normality constrains: $\tilde{\mathbf{q}}\tilde{\mathbf{q}}^* = q_0^2 + \|\mathbf{q}\|^2 = 1$, where $\tilde{\mathbf{q}}^*$ is the conjugate of $\tilde{\mathbf{q}}$.

In practice we represent the rotation component by the vector:

$$\mathbf{s} \equiv \frac{\mathbf{q}}{q_0}$$

from which the quaternion $\tilde{\mathbf{q}}$ can be reconstructed:

$$q_0 = \frac{1}{\sqrt{1 + \mathbf{s}^t\mathbf{s}}} \quad ; \quad \tilde{\mathbf{q}} = (q_0, q_0\mathbf{s}) \quad .$$

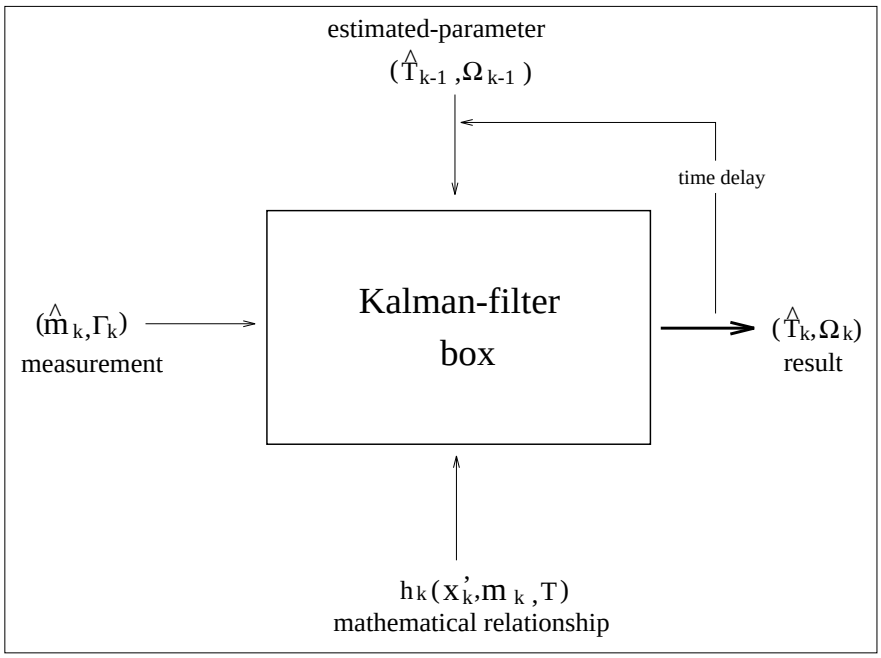

Figure 5: The Kalman filter for static-parameter estimation; The three inputs and the estimation output.

The vector $\mathbf{s}$ is a convenient representation of the rotational component; in addition to being minimal (having 3 parameters) the rotation equation is linear in $\mathbf{s}$ as will be shown later. In order to avoid singularities in the representation when $q_0 = 0$ we always use two object-centered coordinate systems, simultaneously.

Considering these two components, the parameter vector to be estimated during the filtering process is:

$$\mathbf{T} = \begin{pmatrix} \mathbf{s} \\ \mathbf{t} \end{pmatrix} \quad .$$

The Kalman filter produces an estimate $\hat{\mathbf{T}}$ of the transformation vector, given the uncertain lines. At each step, the filter receives three inputs and supplies a single output (see Figure 5). The inputs are:

1. An *a priori* estimate of the evaluated parameter vector and the uncertainty associated with it. In our case, in the k^{th} step, the *a priori* estimation will be the estimate evaluated at the previous step $\hat{\mathbf{T}}^{k-1}$ and its associated covariance Ω^{k-1}. The covariance

matrix Ω^0 in the initial step will be set to infinity since no *a priori* knowledge about $\mathbf{T}$ is assumed and the choice of $\hat{\mathbf{T}}^0$ does not affect the end result.

2. The current measurement and its uncertainty, in our case this measurement is the uncertain line $(\hat{\mathbf{M}}_k, \Gamma_k)$ generated from actual $2D$ measurement $(\hat{\mathbf{m}}_k, \Lambda_k)$ where $\hat{\mathbf{M}}_k$ and its associated covariance matrix Γ_k is calculated as elaborated in the previous section.

3. A mathematical relationship between the evaluated parameters and the measurements. This mathematical relationship should be linear in the evaluated parameters. In our case the relationship is:

$$\tilde{\mathbf{M}}_k = \tilde{\mathbf{q}}\tilde{\mathbf{X}}'_k\tilde{\mathbf{q}}^* + \tilde{\mathbf{t}} \quad , \tag{6}$$

where $\tilde{\mathbf{M}}_k, \tilde{\mathbf{X}}'_k, \tilde{\mathbf{t}}$ are quaternions associated with the vectors $\mathbf{M}_k, \mathbf{X}'_k, \mathbf{t}$ respectively. Given that $\tilde{\mathbf{q}}\tilde{\mathbf{q}}^* = 1$, multiplying Equation (6) by $\tilde{\mathbf{q}}$ yields:

$$\tilde{\mathbf{M}}_k\tilde{\mathbf{q}} = \tilde{\mathbf{q}}\tilde{\mathbf{X}}'_k + \tilde{\mathbf{t}}\tilde{\mathbf{q}} \quad .$$

Isolating the vector component of this quaternion equation and dividing by q_0 we obtain the matrix equation:

$$h_k(\mathbf{X}'_k, \mathbf{M}_k, \mathbf{T}) \equiv <\mathbf{M}_k + \mathbf{X}'_k> \mathbf{s} + (\mathbf{M}_k - \mathbf{X}'_k) - (I_3 - <\mathbf{s}>)\mathbf{t} = \mathbf{0} \quad , \tag{7}$$

where $\mathbf{s} \equiv \frac{\mathbf{q}}{q_0}$ as previously defined, I_3 is the 3×3 identity matrix and $<\cdot>$ denotes the matrix form of a cross product, i.e:

$$<\mathbf{v}> = \begin{pmatrix} 0 & -v_z & v_y \\ v_z & 0 & -v_x \\ -v_y & v_x & 0 \end{pmatrix} \quad ; \quad <\mathbf{v}>\mathbf{m} = -<\mathbf{m}>\mathbf{v} = \mathbf{v} \times \mathbf{m} \quad .$$

The equation $h_k(\mathbf{X}'_k, \mathbf{M}_k, \mathbf{T}) = 0$ is not linear as required in the K.F., therefore, we use the *extended Kalman filter* (E.K.F.) [13, 16] which is a generalization of the Kalman filter to non-linear systems where transition from step $k-1$ to step k is performed using a linear approximation of h_k by taking the first order Taylor expansion around $(\hat{\mathbf{X}}'_k, \hat{\mathbf{M}}_k, \hat{\mathbf{T}}^{k-1})$. The linearization of the measurement equations for our case are given in Appendix B.

The output of the K.F fuser is an updated estimation of the evaluated parameters and its associated uncertainty; in our case $\hat{\mathbf{T}}^k$ and Ω^k respectively. The K.F. fuser is of the form:

$$\hat{\mathbf{T}}^k = f(\hat{\mathbf{T}}^{k-1}, \Omega^{k-1}, \hat{\mathbf{X}}'_k, \Sigma_k, \hat{\mathbf{M}}_k, \Gamma_k, h_k) \quad .$$

Thus, at each stage k, there is no need of retaining any of the previously considered measurements. Only the current estimate $\hat{\mathbf{T}}^{k-1}$ and its associated uncertainty Ω^{k-1} need be retained. The K.F. updating equations are given in Appendix C.

The K.F. updating equations yield an unbiased estimate of $\mathbf{T}$ which is optimal in the linear minimal variance criterion [1], i.e. $\hat{\mathbf{T}}^k$ minimizes

$$C = \sum_{i=1}^{k} \hat{h}_i(\mathbf{T}) W_i^{-1} \hat{h}_i^t(\mathbf{T}) \quad , \tag{8}$$

where $\hat{h}_i(\mathbf{T}) = h_i(\hat{\mathbf{X}}'_i, \hat{\mathbf{M}}_i, \mathbf{T})$ and W_i is the covariance matrix of $\hat{h}_i$ (Appendix D). The value C in Equation 8 is in fact the same value as in Equation 3 which we aimed to minimize. Thus, the desired solution as formulated in Sections 3 and 4 is the same solution as obtained from the K.F. updating equations.

Since the measurement equations are linear approximations of non-linear equations the initial solution obtained by the K.F. will not necessarily be the correct solution. This case will happen when the linearization is around a point that is not close enough to the correct solution. In order to reduce the influence of the linearization *local iterative K.F.* [13, page 349] is applied. In the iterations the constraints are relinearized around the updated solution obtained by the K.F. and another cycle of K.F. is performed using the new version of the linearized constraints.

Note, that the general K.F. deals with a parameter vector that is changing with time, whereas in our case the estimated transformation, $\mathbf{T}$, is static and does not change between measurements.

8 From Predicted Objects to Recognition

After the optimal transformation $\hat{\mathbf{T}}^n$ is estimated from the n measurements, the second stage in the recognition task is to find the "similarity" between the predicted object and the

candidate model. This "similarity" is calculated with the $3D$ distance, however it has to be consider with the uncertainty of the model features and the measurements. Fortunately, there is no need to reconstruct the predicted object in order to calculate its similarity to the model: Given a transformation $\mathbf{T}$, the M.D. between the predicted object and the transformed model is given by C of Equation 8 where the summation is over the n measurements. Thus, the similarity between the predicted object and a candidate model is given by the value C calculated for the final transformation $\hat{\mathbf{T}}^n$.
If a reconstruction of the predicted object is required, the calculation of its feature locations is simple: We transform each model-point $\hat{\mathbf{X}}'_i$ by the optimal transformation $\hat{\mathbf{T}}^n$ and find a point $\mathbf{U}_i$ such that the sum of the M.D. to the uncertain line $(\hat{\mathbf{M}}_i, \Gamma_i)$ and the M.D. to the transformed model point $T(\hat{\mathbf{X}}_i)$ is minimal. The mathematical formulation of this reconstruction is elaborated in Appendix E. Note that if the i^{th} measurement is a perfect measurement (Λ_i is zero) then $\mathbf{U}_i$ will be on the line $line(\hat{\mathbf{m}}_i)$.

9 Results

We tested our method by simulating a model as a collection of points. The points of the model were chosen by random sampling in the cube $[0..100]^3$. The model points were transformed by a transformation $\mathbf{T}$ composed of a rotation $\mathbf{s}$ and a translation $\mathbf{t}$. The model points and the measurements were contaminated by white Gaussian noise. All model points were contaminated by $3D$ noise having identical uncertainty (and diagonal covariance matrix) and all the measurements were contaminated with $2D$ noise which was added to the image plane and having identical uncertainty as well.

Graphs 6-8 show the convergence of the estimates of the rotation $\hat{\mathbf{s}}$ and the translation $\hat{\mathbf{t}}$ as a function of the number of measurements. The vertical ordinate represents the squared deviation of the estimate from the real value, i.e:

$$t_i^{error} = \|\hat{\mathbf{t}}_i - \mathbf{t}\|^2 \quad \text{left graphs} \quad \text{and} \quad s_i^{error} = \|\hat{\mathbf{s}}_i - \mathbf{s}\|^2 \quad \text{right graphs} \quad .$$

The results shown in the graphs were averaged over 100 processes of 100 randomly generated objects. In the graphs two cases are shown: the convergence of the estimate when $2D$ distance metric is used as the minimization criterion, and the convergence when the $3D$ distance is used. The variance of the noise was not given to the algorithms and it was assumed that each measured point is contaminated by the same amount.

Graphs 6-8 show three typical cases: Graph 6 shows a case where the noise in the model-points is dominant (the s.t.d of the model noise was proportional to 8% of the total size of the body where the image noise was proportional to 1%), Graph 7 shows a case where the noise in the image is dominant (the s.t.d of the image noise was proportional to 5% of the total size of the body where the model noise was proportional to 1%), and Grpah 8 shows a case where the noise in the model-points is dominant (the same as in Graph 6) but the distance of the body from the image plane is quite large (100 times the focal length) relative to the case shown in graphs 6 (where the distance is about 10 times the focal length).

It is demonstrated that the $3D$ distance metric is advantageous over the $2D$ distance metric when the model noise is dominant and the object is located quite close to the image plane. In the cases where the image noise is dominant or when the object is far away from the image plane the improvement of the estimate using the $3D$ distance metric is negligible. These results are reasonable since the algorithms assume the same amount of noise for each measurement. When the image noise is dominant this assumption is correct and the $2D$ metric gives the same results as the $3D$ metric. If the model noise is dominant, its projection onto the image plane is not identical for each measurement and therefore $3D$ metric is essential. In the case where the object is far away from the image plane, the projection of the model noise onto the image plane is almost identical for all the measurements and therefore the the $2D$ metric is again useful.

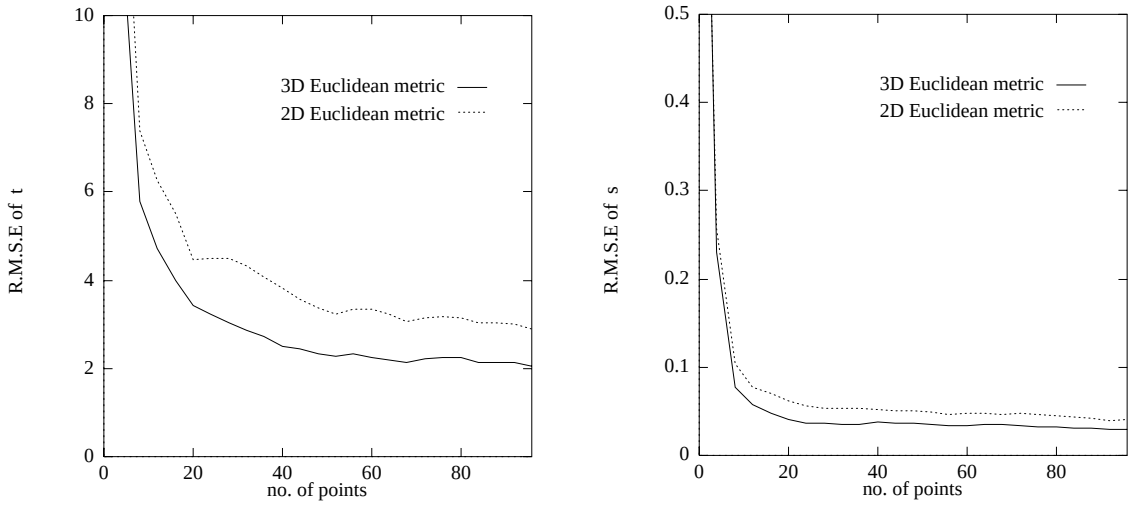

Figure 6: A comparison between a $2D$ distance metric based algorithm and a $3D$ distance metric based algorithm. Left graph: convergence of the deviation of the translation estimate $\hat{\mathbf{t}}$. Right graph: convergence of the deviation of the rotation estimate $\hat{\mathbf{s}}$. In this case the s.t.d of the model noise was proportional to 8% of the total size of the body where the image noise was proportional to 1%.

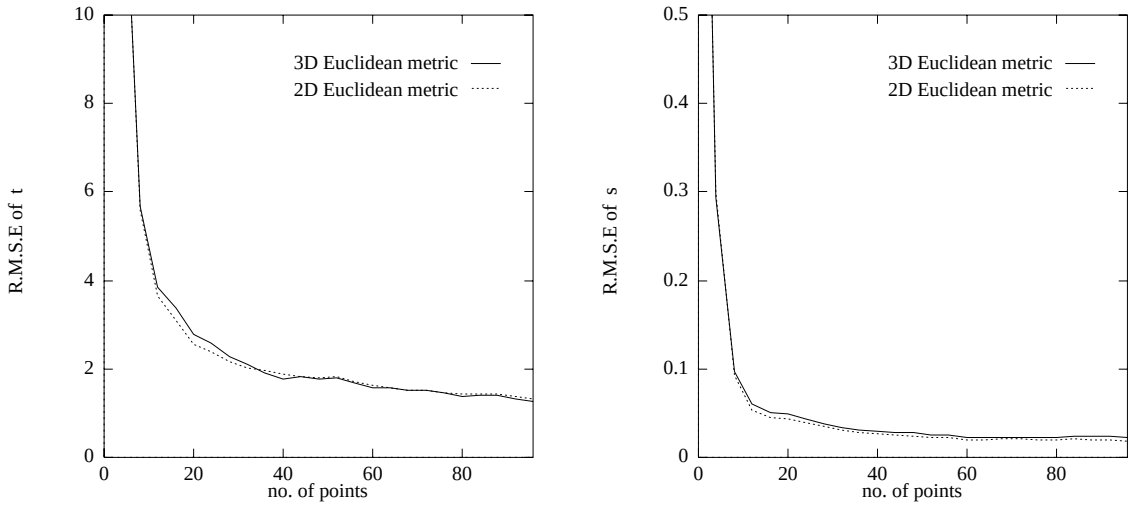

Figure 7: A comparison between a $2D$ distance metric based algorithm and a $3D$ distance metric based algorithm. Left graph: convergence of the deviation of the translation estimate $\hat{\mathbf{t}}$. Right graph: convergence of the deviation of the rotation estimate $\hat{\mathbf{s}}$. In this case the s.t.d of the model noise was proportional to 1% of the total size of the body where the image noise was proportional to 5%.

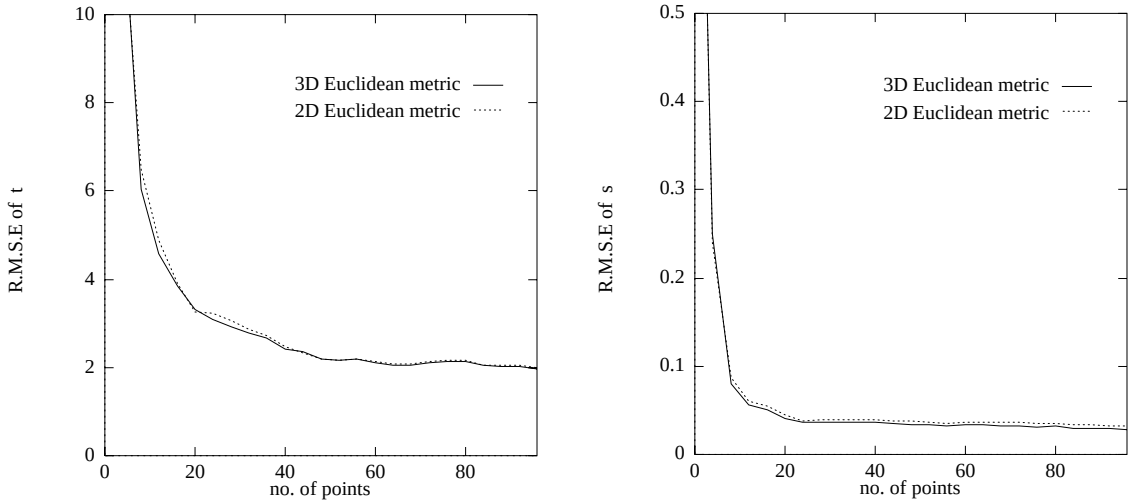

Figure 8: A comparison between a $2D$ distance metric based algorithm and a $3D$ distance metric based algorithm. Left graph: convergence of the deviation of the translation estimate $\hat{\mathbf{t}}$. Right graph: convergence of the deviation of the rotation estimate $\hat{\mathbf{s}}$. In this case the s.t.d of the model noise and the image noise were the same as in graphs 6, but the distance of the body from the image plane was about 100 times the focal length.

10 Conclusion

This paper presents a new scheme which defines a $3D$ distance metric between model features and $2D$ measurements obtained following a projection. This $3D$ distance is defined as the minimum $3D$ distortion the model must undergo in order to fit some permissable reconstruction of the measurements. It was demonstrated that the proposed $3D$ distance metric is usefull when the model noise is dominant and when the object is close to the image plane.

Acknowledgements

I wish to thank Ronen Basri for a fruitful discussion which led me to this paper.

Appendix:

A Rotation Quaternion

A quaternion $\tilde{\mathbf{q}}$ is composed of two parts - the scalar part q_0 and the vector part $\mathbf{q}$:

$$\tilde{\mathbf{q}} = (q_0, \mathbf{q}) = (q_0, q_1 i + q_2 j + q_3 k) \quad .$$

If $\mathbf{v}'$,$\mathbf{v}$ are vectors in $\Re^3$ such that $\mathbf{v} = R\mathbf{v}$, when R is a rotation matrix, then the corresponding expression in quaternion form is:

$$\tilde{\mathbf{v}}' = \tilde{\mathbf{q}}\tilde{\mathbf{v}}\tilde{\mathbf{q}}^* \quad .$$

The quaternions $\tilde{\mathbf{v}}, \tilde{\mathbf{v}}'$ correspond to the vectors $\mathbf{v}$,$\mathbf{v}'$ respectively as follows:

$$\tilde{\mathbf{v}} = (0, \mathbf{v}) \quad ; \quad \tilde{\mathbf{v}}' = (0, \mathbf{v}')$$

and $\tilde{\mathbf{q}}^*$ is the conjugate of $\tilde{\mathbf{q}}$;

$$\tilde{\mathbf{q}}^* = (q_0, -\mathbf{q}) \quad .$$

$\tilde{\mathbf{q}}$ represents a rotation of the vector $\mathbf{v}$ by angle θ around a unit vector $\hat{\mathbf{n}}$ where:

$$q_0 = \cos(\frac{\theta}{2}) \quad ; \quad \mathbf{q} = \sin(\frac{\theta}{2})\hat{\mathbf{n}}$$

so that

$$\|\tilde{\mathbf{q}}\|^2 = \tilde{\mathbf{q}}\tilde{\mathbf{q}}^* = q_0^2 + \|\mathbf{q}\|^2 = 1 \quad .$$

B Linearization of the Measurement Equations for the E.K.F

At the k^{th} step there is a non-linear measurement equation: $h_k(\mathbf{X}'_k, \mathbf{M}_k, \mathbf{T}) = 0$ as written in Equation 7. Since the K.F. deals only with linear processes we use the linear approximation of h_k by taking the first order Taylor expansion around $(\hat{\mathbf{X}}'_k, \hat{\mathbf{M}}_k, \hat{\mathbf{T}}^{k-1})$:

$$h_k(\mathbf{X}'_k, \mathbf{M}_k, \mathbf{T}) = 0 \approx h_k(\hat{\mathbf{X}}'_k, \hat{\mathbf{M}}_k, \hat{\mathbf{T}}^{k-1}) + \frac{\partial h_k}{\mathbf{X}'_k}(\mathbf{X}'_k - \hat{\mathbf{X}}'_k) + \frac{\partial h_k}{\mathbf{M}_k}(\mathbf{M}_k - \hat{\mathbf{M}}_k) + \frac{\partial h_k}{\mathbf{T}}(\mathbf{T} - \hat{\mathbf{T}}^{k-1}) \tag{9}$$

Equation 9 can be rewritten as a linear equation:

$$\mathbf{z}_k = H_k \mathbf{T} + \eta_k \quad , \tag{10}$$

where

$$\begin{aligned}
\mathbf{z}_k &= <\hat{\mathbf{s}}^{k-1}> \hat{\mathbf{t}}^{k-1} + \hat{\mathbf{X}}'_k - \hat{\mathbf{M}}_k \\
H_k &= [<\hat{\mathbf{M}}_k + \hat{\mathbf{X}}'_k - \hat{\mathbf{t}}^{k-1}>, (<\hat{\mathbf{s}}^{k-1}> - I_3)] \\
\eta_k &= [I_3 - <\hat{\mathbf{s}}^{k-1}>](\mathbf{M}_k - \hat{\mathbf{M}}_k) - [I_3 + <\hat{\mathbf{s}}^{k-1}>](\mathbf{X}'_k - \hat{\mathbf{X}}'_k) \quad .
\end{aligned}$$

$\mathbf{z}_k$ represents the new "measurement", H_k is the matrix denoting a linear connection between the "measurement" and the actual transformation $\mathbf{T}$. Both $\mathbf{z}_k$ and H_k can be derived from $\hat{\mathbf{M}}_k$, $\hat{\mathbf{X}}'_k$, $\hat{\mathbf{T}}^{k-1}$. The term η_k - depicts the noise in the "measurement" $\mathbf{z}_k$ and satisfies:

$$\begin{aligned}
E\{\eta_k\} &= 0 \\
\mathrm{var}\{\eta_k\} &= [I_3 - <\hat{\mathbf{s}}^{k-1}>]\Lambda_k[I_3 - <\hat{\mathbf{s}}^{k-1}>]^t + [I_3 + <\hat{\mathbf{s}}^{k-1}>]\Sigma_k[I_3 + <\hat{\mathbf{s}}^{k-1}>]^t = W_k \quad .
\end{aligned}$$

Notice that according to the K.F. definition it is assumed that there is no correlation between the different measurement noise ($\mathrm{cov}\{\eta_i, \eta_j\} = 0 \ \forall i \neq j$). This assumption is not always valid. When there is correlation between several measurements, we may consider these measurements as a single measurement by grouping the measurement values into a single vector and by combining their corresponding equations into a single vector equation.

C The Kalman Filter Equations for Static Systems

Assume the measurement equations are as written in Equation 10. The recursive K.F. updating equations for time step k are:

$$\begin{aligned} &\textit{state estimate update}: && \hat{\mathbf{T}}^k = \hat{\mathbf{T}}^{k-1} + K_k(\mathbf{z}_k - H_k\hat{\mathbf{T}}^{k-1}) \\ &\textit{state covariance update}: && \Omega^k = \Omega^{k-1} - K_k H_k \Omega^{k-1} \\ &\textit{Kalman gain matrix}: && K_k = \Omega^{k-1} H_k^t (H_k \Omega^{k-1} H_k^t + W_k)^{-1} \quad . \end{aligned}$$

D Covariance Matrix of $\hat{h}_i$

The value of h_i (Equation 7) can be linearly approximated, in the i^{th} step, by taking the first order Taylor expansion around $(\hat{\mathbf{X}}'_i, \hat{\mathbf{M}}_i)$:

$$h_i = 0 \approx \hat{h}_i(\mathbf{T}) + \frac{\partial h_i}{\mathbf{X}'_i}(\mathbf{X}'_i - \hat{\mathbf{X}}'_i) + \frac{\partial h_i}{\mathbf{M}_i}(\mathbf{M}_i - \hat{\mathbf{M}}_i)$$

where the derivatives are taken at $(\hat{\mathbf{X}}'_i, \hat{\mathbf{M}}_i)$ and $\hat{h}_i = h_i(\hat{\mathbf{X}}'_i, \hat{\mathbf{M}}_i, \mathbf{T})$. From the above it is clear that $\hat{h}_i$ is a zero mean random process with the covariance:

$$W_i = E\{\hat{h}_i \hat{h}_i^t\} = (\frac{\partial h_i}{\mathbf{X}'_i})\Sigma_i(\frac{\partial h_i}{\mathbf{X}'_i})^t + (\frac{\partial h_i}{\mathbf{M}_i})\Gamma_i(\frac{\partial h_i}{\mathbf{M}_i})^t \quad .$$

E Reconstruction of Predicted Object

Let $\hat{\mathbf{T}} = \begin{pmatrix} \hat{\mathbf{s}} \\ \hat{\mathbf{t}} \end{pmatrix}$ be the final transformation estimated after fusing all the available measurements. From $\hat{\mathbf{s}}$ we can build an associated rotation matrix $\hat{R}$ [17] and an expression for the transformed model:

$$\begin{aligned} \hat{X}_i &= \hat{R}\hat{X}'_i + \hat{t} \\ \Sigma'_i &= \hat{R}\Sigma_i\hat{R}^t \end{aligned}$$

Given the transformed model-point $(\hat{X}_i, \Sigma'_i)$ and the associated measured point $(\hat{M}_i, \Gamma_i)$ (uncertain line) the predicted object is obtain by fusing these two points:

$$U_i = \hat{X}_i + K(\hat{M}_i - \hat{X}_i)$$

$$\text{where} \qquad K \;=\; \Sigma_i'(\Sigma_i' + \Gamma_i)^{-1}$$

This equation can be obtain from the K.F. updating equations where $\hat{X}_i$ is regarded as the state vector and $\hat{M}_i$ as a measurement.

References

[1] B.D.O. Anderson and J.B. Moore. *Optimal Filtering.* Prentice-Hall, Inc., N.J., 1979.

[2] K.S. Arun, T.S. Huang, and S.D. Blostein. Least squares fitting of two 3D point sets. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 9(5):698–700, Sept. 1987.

[3] R. Basri and D. Weinshal. Distance metric between 3D models and 2D images. In *The 9th Israeli Symposium on Artificial Intelligence and Computer Vision*, pages 337–356, Dec. 1992.

[4] T. Fan, G. Medioni, and R. Nevatia. Recognizing 3D objects using surface descriptions. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11(11):1140–1157, Nov. 1989.

[5] O.D. Faugeras and M. Hebert. The representation, recognition, and positioning of 3D shapes from range data. In A. Rosenfeld, editor, *Techniques for 3D Machine Perception*, pages 13–51. Elsevier Science, 1986.

[6] M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, June 1981.

[7] W.E.L. Grimson. On the verification of hypothesized matches in model-based recognition. In *European Conference on Computer Vision*, pages 489–498, 1990.

[8] W.E.L. Grimson and T. Lozano-Perez. Model-based recognition and localization from sparse range or tactile data. *International Journal of Robotics Research*, 3(3):3–35, 1984.

[9] Y. Hel-Or and M. Werman. Absolute orientation from uncertain data: A unified approach. In *Conference on Computer Vision and Pattern Recognition*, pages 77–82, 1992.

[10] B.K.P. Horn. *Robot Vision.* MIT Press, 1986.

[11] B.K.P Horn. Closed-form solution of absolute orientation using unit quaternion. *J. Opt. Soc. Am.*, 4(4):629–642, 1987.

[12] D.P. Huttenlocher and S. Ullman. Object recognition using alingment. In *International Conf. on Computer Vision*, pages 102–111, 1987.

[13] A.H. Jazwinski. *Stochastic Process and Filtering Theory*. Academic Press, 1970.

[14] D.G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31:355–395, 1987.

[15] D.G. Lowe. Fitting parametrized 3-D models to images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13:441–450, May 1991.

[16] P.S. Maybeck. *Stochastic Models, Estimation, and Control*, volume 1. Academic Press, 1979.

[17] B. Sabata and J.K. Aggarwal. Estimation of motion from a pair of range images: A review. *Computer Vision, Graphics and Image Processing*, 54(3):309–324, Nov 1991.