Covariance-Based Registration

Charles V. Stewart
Dept. of Computer Science
Rensselaer Poly. Inst.
Troy, New York 12180
stewart@cs.rpi.edu

June 24, 2002

1 Introduction

The registration problem in computer vision is the problem of finding the transformation that best aligns (registers) a model with a data set or best registers two or more data sets. The goal is to bring the model and the data set or the multiple data sets into the same coordinate system. Solutions to this problem are required in many application domains. In industrial inspection, registration between model and data is necessary for comparing ideal ("nominal") parts with manufactured parts so that defects in the manufacturing process may be identified [32, 33]. In model construction, placing sensor measurements in the same coordinate system is the necessary prerequisite for building complete models rather than models dependent only on individual views. In medicine, registration facilitates treatment monitoring, mixing of sensed data from different modalities, and application of surgical plans developed off-line [10, 13, 15]. Each of these applications requires precise and accurate estimates of the transformation.

The registration problem has many forms. Differences are due to the type of data, the type of model (if any) and the type of transformation. Examples include image to image registration for mosaic construction [23, 40, 37], range data to range data registration for model construction in reverse engineering [3, 19, 29, 35, 9], and model to image registration for tracking, motion estimation, object recognition or camera calibration [14, 22, 27, 28, 44]. The particular instance of the registration problem considered here is registering a three-dimensional model to one or more range data sets *as precisely and accurately as possible*. This problem, which differs from the range data to range data registration problem because the model is known in advance and is not variable, will be what's meant by the "registration problem". It has particular relevance to industrial inspection applications [32].

Precise and accurate registration requires use of constraints based directly on the data points. Feature-based techniques, such used in traditional object recognition and pose determination [8, 12], and methods based on global shape descriptions [26], such as extended gaussian images, spherical attribute images [18], or spin images [24], are good for coarse positioning when a prior estimate of the transformation is not known, but must be followed by point-based registration for precision and accuracy [18, 24].

The main idea in most point-based registration techniques appears in several nearly simultaneous papers proposing "iterative closest point" (ICP) algorithms [4, 7, 32, 46]. ICP algorithms iterate (temporary) matching and pose estimation steps, specifically (1) finding the closest model point to each data point based on a current transformation (pose) estimate and (2) revising the pose estimate based on the collection of matches. Each of these algorithms minimizes a Euclidean distance metric in matching. The pose estimation constraints are either the Euclidean distance between each matched data and model point [4, 32, 46] or the Euclidean distance between a data point and a linearization of the model surface around the matched model point [7, 39]. The latter is called the "normal distance". ICP algorithms have been extended to registering volumetric images and to registering combined range [13] and color images [25]. An alternative method to ICP algorithms is described in [6, 41], where data to model distance measures are represented and computed using what's called an octree spline. This avoids the need for an explicit matching step, but makes pose estimation non-linear.

In light of the goal of making registration as precise and accurate as possible, it should be clear that what's missing from current point-based registration techniques are measures of uncertainty in the data. The significance of incorporating data uncertainty into estimation problems such as registration may be seen by examining two simpler problems, each provably equivalent to a special case of registration (Appendix A).

Multivariate location: The problem is to estimate a location $\boldsymbol{\mu}$ from n point measurements $\mathbf{x_i}$, each with an associated covariance matrix $\mathtt{S_i}$. The optimal estimate minimizes the sum of the squared Mahalanobis distances:

$$\sum_i (\mathbf{x_i} - \mu)^T \mathtt{S}_\mathtt{i}^{-1} (\mathbf{x_i} - \mu). \tag{1}$$

This estimate is easily shown to be

$$\hat{\boldsymbol{\mu}} = \left(\sum_i \mathtt{S}_\mathtt{i}^{-1} \right)^{-1} \left(\sum_\mathtt{i} \mathtt{S}_\mathtt{i}^{-1} \mathbf{x_i} \right)$$

This reduces to the ordinary average only if the covariance matrices are equal.

Linear regression: Restricting attention to points in two-dimensions, so that $\mathbf{x}_i = (x_i, y_i)^T$, consider the difference between ordinary regression and "orthogonal regression". Mathematically the error metrics to be minimized are

$$\sum_i (y_i - mx_i - b)^2 \qquad \text{and} \qquad \sum_i (a_0 + a_1 x_i + a_2 y_i)^2 \tag{2}$$

respectively, the latter being subject to the constraint $a_1^2 + a_2^2 = 1$. (Of course, the two sets of line parameters may be translated back and forth except when $a_2 = 0$.) When the error covariance matrices are $\mathtt{S_i} = \mathrm{diag}(0, \sigma^2)$, minimization of the ordinary regression metric yields an *unbiased* (and therefore most accurate) estimate of the line parameters, whereas minimization of the orthogonal regression metric yields a biased estimate. The situation is reversed when the error covariance matrices are $\mathtt{S_i} = \mathrm{diag}(\sigma^2, \sigma^2)$. Different error covariance matrices require different estimators for unbiased estimates.

In both examples, obtaining the most accurate estimates requires use of measurement error represented as error covariance matrices.

The main questions addressed in this paper are (1) how to incorporate error covariance matrices into point-based registration algorithms, and (2) what are the practical consequences of doing so. The first question is addressed by formulating registration as a statistical optimization problem using distance metrics based on the Mahalanobis distance.[1] Two new ICP algorithms arise naturally when solving this minimization. One is a generalization of current ICP algorithms that use Euclidean distance in the pose refinement step. A second is a generalization of ICP algorithms that use normal distances in pose refinement. Octree spline registration algorithms are not easily extended to incorporate measurement error in the data because the octree spline representation depends on the model only. As a result such algorithms are not considered further here.

The practical consequences of incorporating error covariance matrices into point-based registration algorithms are two-fold. The first and obviously intended result is more accurate registration. The amount of improvement over standard ICP algorithms depends on a number of considerations, including the number and extent of the data points, the magnitude of the noise in the data, and the object pose itself. The second and somewhat surprising result is two additional new algorithms that are simpler and faster algorithm than current ICP algorithms in the special case that the error covariance matrices are (nearly) unidirectional. Since the measurement errors of most range sensors are concentrated along the optical axis of the cameras [5, 43], these new algorithms with their improved accuracy and efficiency should generally be preferred over current ICP algorithms.

2 Problem Formulation

The discussion begins by formulating registration as an optimization problem. This in turn begins with a discussion of the model, the data, and the transform

[1] See Chitrai, Weng and Jain for an important but special case solution to the problem of statistically optimal registration of two data sets.

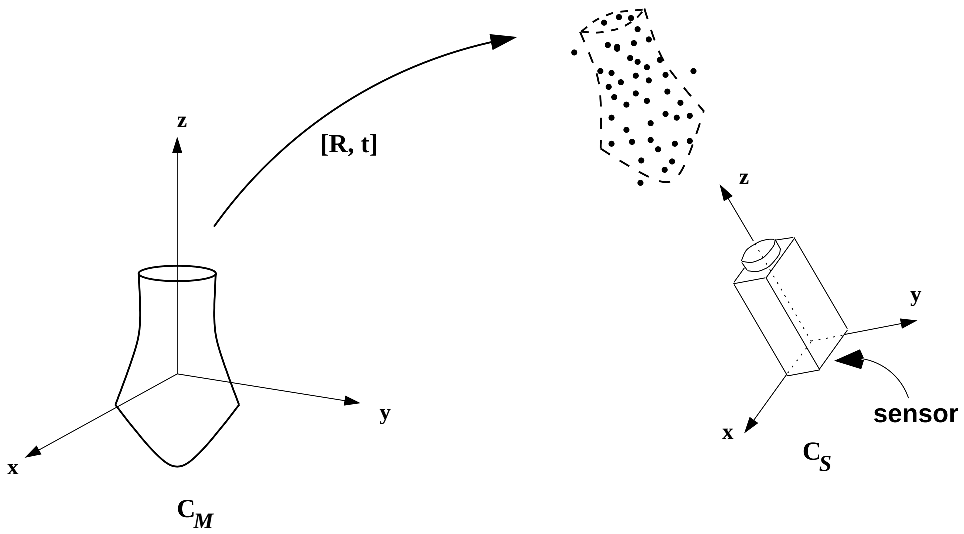

Figure 1: The registration problem is finding the rigid transformation best aligns the model with the data. The transformation is a mapping from the coordinate system $\mathtt{C_M}$ in which the model is described into the coordinate system $\mathtt{C_S}$ in which the data is acquired.

to be computed.

2.1 The Model and the Data

The model is assumed to be described mathematically in some convenient coordinate system, $\mathtt{C_M}$ (Fig. 1). This description may be an implicit function, a parametric model, spline patches, or a triangulated surface mesh. For simplicity in the derivations, the model is assumed to be described implicitly as the set of pointss $\mathbf{p}$ such that $f(\mathbf{p}) = 0$. The resulting algorithms are easily adjusted for different model representations. In fact, as discussed later, some offline preprocessing of the model is generally necessary prior to registration.

The data are a set of sensed points from an instance of the object. Assume there are N data points in the set $Q = \{\mathbf{q}_i\}$, where $\mathbf{q}_i = (x_i, y_i, z_i)^T$, and each data point has an associated covariance matrix $\mathtt{S_i}$. These points and matrices are in the sensor coordinate system, $\mathtt{C_S}$ (Fig. 1). In the case of range data, the points may be conveniently stored in or converted to an image format (a "range image"), where each pixel location (u, v) stores a measured point in $\Re^3$ denoted by $\mathbf{q}(u, v)$. A camera projection function, usually perspective or weak perspective, denoted by $\mathcal{P}$ determines the mapping from range points to pixel locations.

The covariance matrices, $\mathtt{S_i}$, which depend on the sensor and the data point locations, $\mathbf{q}_i$, are symmetric and positive semi-definite. It will be important to consider the case of $\mathtt{S_i}$ being singular and even rank 1, which represents the most extremely anisotropic distribution. Rough approximations to the covariance matrices may be obtained by analyzing the sensing technique and camera geometry. More complete characterization requires experimental analysis, often as part of a rigorous calibration process [17]. This analysis will result in a look-up table mapping sensed point positions to covariance matrices. Sometimes this mapping may even depend on surface orientation, due to angle of incidence and reflectivity effects. Such a dependence causes a problem for registration because surface orientation is not known in advance. A straightforward solution to this problem is discussed later in the paper.

The registration problem is to find the model-to-data transformation, which is assumed here to be a rigid transformation described by a rotation $\mathtt{R}$ and translation $\mathbf{t}$. The inverse transformation mapping data-to-model is $\mathtt{R}^\mathtt{T}$ and $-\mathtt{R}^\mathtt{T}\mathbf{t}$.[2] Since the goal is precise registration, it is assumed that initial estimates of $\mathtt{R}$ and $\mathbf{t}$ are known or easily provided. These may be obtained by using application constraints, or by using feature-based or shape description methods, or, for example, by alignment of 0th and 1st order moments in the model and in the data.

2.2 Registration as an Optimization Problem

Given a data point and covariance matrix, $\mathbf{q}_i$ and $\mathtt{S_i}$, given a particular model point $\mathbf{p}$, and given a fixed $\mathtt{R}$ and $\mathbf{t}$, the squared Mahalanobis distance between $\mathbf{q}_i$ and the transformed model point $\mathtt{R}\,\mathbf{p}+\mathbf{t}$ is

$$(\mathtt{R}\,\mathbf{p}+\mathbf{t}-\mathbf{q}_\mathtt{i})^\mathtt{T}\,\mathtt{S_i}^{-1}\,(\mathtt{R}\,\mathbf{p}+\mathbf{t}-\mathbf{q}_\mathtt{i}). \tag{3}$$

When $\mathtt{S_i}$ is singular, the error vector $\mathtt{R}\,\mathbf{p}+\mathbf{t}-\mathbf{q}_\mathtt{i}$ must be entirely within the column space of $\mathtt{S_i}$; otherwise the Mahalanobis distance is infinite.

Since the correspondence between data points and the model is not known, it is important to formulate the Mahalanobis distance in terms of the data point and the model surface. This results in a constrained minimization:

$$\begin{aligned} D_M^2(\mathbf{q}_i, \mathtt{S_i}; \mathtt{f}; \mathtt{R}, \mathbf{t}) = \min_{\mathbf{p}} & (\mathtt{R}\,\mathbf{p}+\mathbf{t}-\mathbf{q}_\mathtt{i})^\mathtt{T}\,\mathtt{S_i}^{-1}\,(\mathtt{R}\,\mathbf{p}+\mathbf{t}-\mathbf{q}_\mathtt{i}) \\ & \text{subject to } f(\mathbf{p}) = 0 \end{aligned} \tag{4}$$

This defines the square Mahalanobis distance between a data point and the model surface as the minimum Mahalanobis distance over all model surface points. The "closest" point, which is not made explicit in (4) but will be required later, is denoted by $\mathbf{p}_i$. It depends on the data point and covariance matrix, the model surface, and the transformation $\mathtt{R}, \mathbf{t}$.

Given these definitions, registration becomes the problem of minimizing the combined square Mahalanobis distances of all data points.

$$Q(\{\mathbf{q}_i, \mathtt{S_i}\}; \mathtt{f}; \mathtt{R}, \mathbf{t}) = \sum_\mathtt{i} \mathtt{D}_\mathtt{M}^2(\mathbf{q}_\mathtt{i}, \mathtt{S_i}; \mathtt{f}; \mathtt{R}, \mathbf{t}). \tag{5}$$

[2] For notational convenience, rotations are described using orthonormal matrices. In practice, any convenient representation may be used [11, Ch. 5].

This defines the minimization problem we want to solve to estimate $\mathtt{R}$ and $\mathbf{t}$. A formalization and generalization of the problem addressed in current ICP papers, it involves a two-tiered minimization, first in the Mahalanobis distances of the individual data points and second in the global transformation parameters. These correspond to the two iterated ICP steps of matching and pose estimation described in the introduction.

3 Solving D_M^2

The first step to minimizing $Q(\{\mathbf{q}_i, \mathtt{S_i}\}; \mathtt{f}; \mathtt{R}, \mathbf{t})$ is to understand better the data point to model surface minimization, $D_M(\mathbf{q}_i, \mathtt{S_i}; \mathtt{f}; \mathtt{R}, \mathbf{t})$ defined in (4). This minimization is solved first for linear models and then for general models.

3.1 Linear Models

Linear models (planes in $\Re^3$ and lines in $\Re^2$) are of the form

$$f_L(\mathbf{p}) = \hat{\boldsymbol{\eta}}^T(\mathbf{p} - \mathbf{p}_0) = 0 \tag{6}$$

where $\hat{\boldsymbol{\eta}}^T$ is a unit normal and $\mathbf{p}_0$ is any fixed point on the model surface. Incorporating this into (4) and rewriting D_M^2 as D_L^2 to signify the specialization of the model gives

$$D_L^2(\mathbf{q}_i, \mathtt{S_i}; \hat{\boldsymbol{\eta}}, \mathbf{p}_0; \mathtt{R}, \mathbf{t}) = \min_{\mathbf{p}} (\mathtt{R}\,\mathbf{p} + \mathbf{t} - \mathbf{q_i})^{\mathtt{T}}\ \mathtt{S_i}^{-1}\ (\mathtt{R}\,\mathbf{p} + \mathbf{t} - \mathbf{q_i})$$
$$\text{subject to } \hat{\boldsymbol{\eta}}^T(\mathbf{p} - \mathbf{p}_0) = 0 \tag{7}$$

This is solved by writing the minimization using Lagrange multipliers:

$$F(\mathbf{p}, \lambda) = (\mathtt{R}\,\mathbf{p} + \mathbf{t} - \mathbf{q_i})^{\mathtt{T}}\, \mathtt{S_i}^{-1}\, (\mathtt{R}\,\mathbf{p} + \mathbf{t} - \mathbf{q_i}) + 2\lambda\hat{\boldsymbol{\eta}}^{\mathtt{T}}(\mathbf{p} - \mathbf{p}_0). \tag{8}$$

Taking derivatives with respect to $\mathbf{p}$ and λ, equating the results to $\mathbf{0}$ and 0, and writing in matrix form yields

$$\begin{pmatrix} \mathtt{R^T S_i^{-1} R} & \hat{\boldsymbol{\eta}} \\ \hat{\boldsymbol{\eta}}^T & 0 \end{pmatrix} \begin{pmatrix} \mathbf{p} \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathtt{R^T S_i^{-1}}(\mathbf{q_i} - \mathbf{t}) \\ \boldsymbol{\eta}^T \mathbf{p}_0 \end{pmatrix} \tag{9}$$

Solving for $\mathbf{p}$ results in

$$\mathbf{p} = \mathtt{R}^{\mathtt{T}}\,(\mathbf{q_i} - \mathbf{t}) + \frac{\hat{\boldsymbol{\eta}}^{\mathtt{T}}\,(\mathbf{p}_0 - \mathtt{R}^{\mathtt{T}}\,(\mathbf{q_i} - \mathbf{t}))}{\hat{\boldsymbol{\eta}}^{\mathtt{T}}\,\mathtt{R}^{\mathtt{T}}\,\mathtt{S_i}\,\mathtt{R}\hat{\boldsymbol{\eta}}} \mathtt{R}^{\mathtt{T}}\,\mathtt{S_i}\,\mathtt{R}\,\hat{\boldsymbol{\eta}}, \tag{10}$$

where scalar terms have been gathered in the fraction.

Now, define $\mathbf{q}_i' = \mathtt{R}^{\mathtt{T}}(\mathbf{q_i} - \mathbf{t})$ and $\mathtt{S}_\mathtt{i}' = \mathtt{R^T S_i R}$. These are the data point and its covariance matrix transformed back into the model coordinate system, though they do not depend on the model. Using them, (10) simplifies to

$$\mathbf{p} = \mathbf{q}_i' + \frac{\hat{\boldsymbol{\eta}}^T(\mathbf{p}_0 - \mathbf{q}_i')}{\hat{\boldsymbol{\eta}}^T \mathtt{S}_\mathtt{i}' \hat{\boldsymbol{\eta}}} \mathtt{S}_\mathtt{i}' \hat{\boldsymbol{\eta}} \tag{11}$$

This gives the point $\mathbf{p}$ on the plane $f_L(\mathbf{p}) = 0$ minimizing the square Mahalanobis distance. From here on, this point will be denoted by $\mathbf{p}_i$.

The Mahalanobis distance may now be calculated from (11), thereby solving (7). Using $\mathbf{q}'_i$ and $\mathtt{S}'_\mathtt{i}$ this results in

$$\begin{aligned} D_L^2(\mathbf{q}'_i, \mathtt{S}'_\mathtt{i}; \mathbf{p}_0, \hat{\boldsymbol{\eta}}) &= (\mathbf{p}_i - \mathbf{q}'_i)^T \, {\mathtt{S}'_\mathtt{i}}^{-1} \, (\mathbf{p}_i - \mathbf{q}'_i) \\ &= \frac{[\hat{\boldsymbol{\eta}}^T(\mathbf{p}_0 - \mathbf{q}'_i)]^2}{\hat{\boldsymbol{\eta}}^T \, \mathtt{S}'_\mathtt{i} \, \hat{\boldsymbol{\eta}}}, \end{aligned} \tag{12}$$

This distance metric is calculated in model coordinates, whereas (4) is defined in data coordinates. This is not an issue because the Mahalanobis distance is a unit-less measure. With quantities described in their original coordinate systems, this becomes

$$D_L^2(\mathbf{q}_i, \mathtt{S}_\mathtt{i}; \mathbf{p}_0, \hat{\boldsymbol{\eta}}; \mathtt{R}, \mathbf{t}) = \frac{[\hat{\boldsymbol{\eta}}^\mathtt{T}(\mathbf{p}_0 + \mathtt{R}^\mathtt{T}\,\mathbf{t} - \mathtt{R}^\mathtt{T}\,\mathbf{q}_\mathtt{i})]^2}{\hat{\boldsymbol{\eta}}^\mathtt{T}\,\mathtt{R}^\mathtt{T}\,\mathtt{S}_\mathtt{i}\,\mathtt{R}\,\hat{\boldsymbol{\eta}}}. \tag{13}$$

In either case, the result is quite simple. Importantly, it does not rely on the inverse of the covariance matrix, so it is appropriate for singular covariance matrices (errors along only one or two directions) as well.[3] In fact, as the plane normal $\mathtt{R}\hat{\boldsymbol{\eta}}$ approaches the null space of $\mathtt{S}_\mathtt{i}$, the Mahalanobis distance approaches infinity.

3.2 General Implicit Models

The solution for linear models will be used as part of an iterative solution for general models.

The derivation for the general case starts as above by combining the distance and the constraint in (4) into a Lagrangian form:

$$F(\mathbf{p}, \lambda) = \min_{\mathbf{p}} \left[(\mathtt{R}\,\mathbf{p} + \mathbf{t} - \mathbf{q}_\mathtt{i})^\mathtt{T} {\mathtt{S}_\mathtt{i}}^{-1} (\mathtt{R}\,\mathbf{p} + \mathbf{t} - \mathbf{q}_\mathtt{i}) + 2\lambda \mathtt{f}(\mathbf{p}) \right] \tag{14}$$

Taking derivatives with respect to both $\mathbf{p}$ and λ and setting the results equal to 0 yields

$$\begin{aligned} \frac{\partial F}{\partial \mathbf{p}} &= \mathtt{R}^\mathtt{T}\mathtt{S}_\mathtt{i}^{-1}\mathtt{R}\mathbf{p} + \mathtt{R}^\mathtt{T}\mathtt{S}_\mathtt{i}^{-1}(\mathbf{t} - \mathbf{q}_\mathtt{i}) + \lambda \nabla \mathtt{f}(\mathbf{p}) = \mathbf{0} \\ \frac{\partial F}{\partial \lambda} &= f(\mathbf{p}) = 0 \end{aligned} \tag{15}$$

This must be solved iteratively by linearizing f around an initial closest point estimate, using the linear model solution to obtain an updated estimate, and repeating until convergence. The Mahalanobis distance at the resulting point is the solution to (4).

[3]The derivation above is not a sufficient proof for the case of singular $\mathtt{S}_\mathtt{i}$, but a careful proof would be too distracting for our purposes.

(1) **repeat**
(2) $k = k + 1$
(3) $\mathbf{p}^0 = \mathbf{q}'_i + \dfrac{\hat{\boldsymbol{\eta}}_{i,k}^T(\mathbf{p}_{i,k} - \mathbf{q}'_i)}{\hat{\boldsymbol{\eta}}_{i,k}^T \mathtt{S}'_{\mathtt{i}} \hat{\boldsymbol{\eta}}_{\mathtt{i,k}}} \mathtt{S}'_{\mathtt{i}} \hat{\boldsymbol{\eta}}_{\mathtt{i,k}}$
(4) **repeat**
(5) $\mathbf{p}_{i,k+1} = \mathbf{p}^0 + s(\mathbf{q}'_i - \mathbf{p}^0)$ for s such that $f(\mathbf{p}^0 + s(\mathbf{q}'_i - \mathbf{p}^0)) = 0$.
(6) **if** $((\mathbf{q}_i - \mathbf{p}_{i,k+1})^T\, \mathtt{S}_{\mathtt{i}}^{-1}(\mathbf{q}_{\mathtt{i}} - \mathbf{p}_{\mathtt{i,k+1}}) \;<\; (\mathbf{q}_{\mathtt{i}} - \mathbf{p}_{\mathtt{i,k}})^{\mathtt{T}}\, \mathtt{S}_{\mathtt{i}}^{-1}(\mathbf{q}_{\mathtt{i}} - \mathbf{p}_{\mathtt{i,k}}))$
(7) break;
(8) **else**
(9) $\mathbf{p}^0 = (\mathbf{p}^0 + \mathbf{p}_{i,k})/2$;
(10) **until** $\|\mathbf{p}_{i,k} - \mathbf{p}_{i,k+1}\| < \epsilon$;
(11) $\hat{\boldsymbol{\eta}}_{i,k+1} = \nabla f(\mathbf{p}_{i,k+1}) / \|\nabla f(\mathbf{p}_{i,k+1})\|$
(12) **until** $\|\mathbf{p}_{i,k} - \mathbf{p}_{i,k+1}\| < \epsilon$;

Figure 2: Procedure D_M^2, an iterative procedure for finding the model surface point $\mathbf{p}_i$ minimizing the Mahalanobis distance to the inverse rotated and translated data point. See the text for discussion of the details. The point and normal estimates from the final iteration will be denoted $\mathbf{p}_i$ and $\hat{\boldsymbol{\eta}}_i$ in the algorithm description.

This iterative solution is not as straightforward as it might first appear. To make the explanation clear the following notation is used: index the iterations using k, denote the estimated closest point in iteration k by $\mathbf{p}_{i,k}$, and denote the associated unit surface normal by $\hat{\boldsymbol{\eta}}_{i,k}$. Then, using (11) the updated closest point is

$$\mathbf{p}_{i,k+1} = \mathbf{q}'_i + \frac{\hat{\boldsymbol{\eta}}_{i,k}^T(\mathbf{p}_{i,k} - \mathbf{q}'_i)}{\hat{\boldsymbol{\eta}}_{i,k}^T \mathtt{S}'_{\mathtt{i}} \hat{\boldsymbol{\eta}}_{\mathtt{i,k}}} \mathtt{S}'_{\mathtt{i}} \hat{\boldsymbol{\eta}}_{\mathtt{i,k}} \tag{16}$$

Unfortunately, $\mathbf{p}_{i,k+1}$ will not satisfy $f(\mathbf{p}_{i,k+1}) = 0$ unless $\mathbf{p}_{i,k+1} = \mathbf{p}_{i,k}$ or f is locally linear. Additional steps are generally required in each iteration to project $\mathbf{p}_{i,k+1}$ onto the model surface and to ensure that the resulting point in fact reduces the Mahalanobis distance. These steps are built into Procedure D_M^2 to find the closest model point to inverse transformed data point $\mathbf{q}'_i$. This procedure is given in Fig. 2 and illustrated in Fig. 3.

Several points about Procedure D_M^2 require explanation. It assumes $\mathbf{p}_{i,k}$ has been properly initialized for $k = 0$. Initialization will be considered later as part of the entire optimization. Next, as set in step (3), $\mathbf{p}^0$ minimizes the Mahalanobis distance on the plane determined by $\mathbf{p}_{i,k}$ and $\hat{\boldsymbol{\eta}}_{i,k}$, but is not necessarily on the model surface. Step (5) projects this point onto the model surface. The direction of projection is along the line through $\mathbf{q}'_i$ and $\mathbf{p}^0$, which preserves any constraints imposed by singularities in $\mathtt{S}'_{\mathtt{i}}$, but, depending on the model representation, may not be the most efficient direction of search. If the point found, $\mathbf{p}_{i,k+1}$, does in fact reduce the Mahalanobis distance, it is taken as

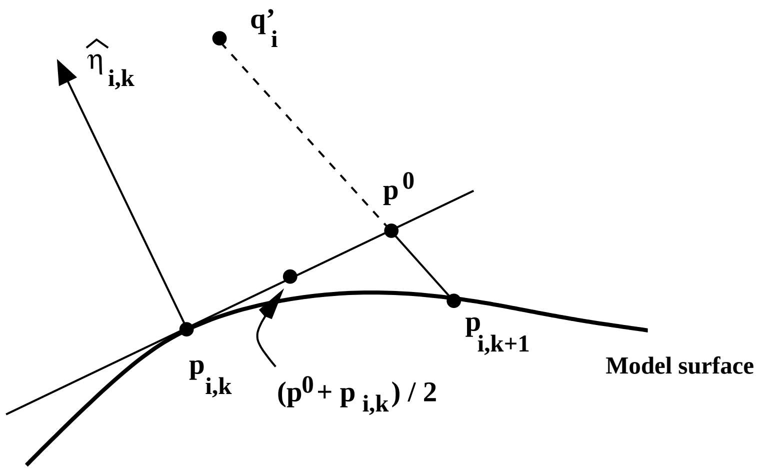

Figure 3: Illustration of Procedure D_M^2.

the next point on the model surface, and the inner loop ends. (This will usually be the case.) Otherwise, the distance between $\mathbf{p}_{i,k}$ and $\mathbf{p}^0$ is halved and the inner loop continues. As a final note on Procedure D_M^2, it is straightforward to see that $\mathbf{p}^0 = (\mathbf{p}^0 + \mathbf{p}_{i,k})/2$ still yields a smaller Mahalanobis distance than $\mathbf{p}_{i,k}$. The reason is that the Mahalanobis distance from $\mathbf{q}_i'$ to the planar surface determined by $\mathbf{p}_{i,k}$ and $\hat{\boldsymbol{\eta}}_{i,k}$ is a quadratic function whose unique minimum is the initial $\mathbf{p}^0$. Moving $\mathbf{p}^0$ toward $\mathbf{p}_{i,k}$ monotonically increase this distance.

Overall, while Procedure D_M^2 appears complicated, it is in fact no more complex in form than the matching step of current ICP algorithms. These find the model point minimizing the Euclidean distance between a data point and the model surface using a procedure that is a special case of Procedure D_M^2. It is obtained by replacing $\mathtt{S_i}$ with the 3×3 identity matrix in Procedure D_M^2. The additional cost of Procedure D_M^2 as written is therefore in computing $\hat{\boldsymbol{\eta}}_{i,k}^T \mathtt{S}_\mathtt{i}' \hat{\boldsymbol{\eta}}_\mathtt{i,k}$ and $\mathtt{S}_\mathtt{i}' \hat{\boldsymbol{\eta}}_\mathtt{i,k}$ in step (3) and computing the Mahalanobis distance instead of the Euclidean distance in step (6). As we will see, these costs are much reduced when $\mathtt{S}$ is less than full rank.

4 Two General Covariance-Based ICP Algorithms

The foregoing iterative procedure minimizing $D_M^2(\mathbf{q}_i, \mathtt{S_i}; \mathtt{f}; \mathtt{R}, \mathbf{t})$ for fixed $\mathtt{R}$ and $\mathbf{t}$ corresponds to the matching step of current ICP algorithms. It also helps to frame the problem of minimizing the overall objective function defined in Equation 5 and repeated here:

$$Q(\{\mathbf{q}_i, \mathtt{S_i}\}; \mathtt{f}; \mathtt{R}, \mathbf{t}) = \sum_\mathtt{i} \mathtt{D}_\mathtt{M}^2(\mathbf{q}_\mathtt{i}, \mathtt{S_i}; \mathtt{f}; \mathtt{R}, \mathbf{t}). \tag{17}$$

The natural approach would be to differentiate $D_M^2(\mathbf{q}_i, \mathtt{S_i}; \mathtt{f}; \mathtt{R}, \mathbf{t})$ with respect to the parameters describing $\mathtt{R}$ and $\mathbf{t}$. Since evaluating $D_M^2(\mathbf{q}_i, \mathtt{S_i}; \mathtt{f}; \mathtt{R}, \mathbf{t})$ itself requires iterative minimization, this differentiation would necessarily be numerical. While this would result in a gradient descent algorithm that is known to converge, it would be extremely expensive.

The alternative is to simplify (17) using an approximation to D_M^2. This approximation should be based on the set of (current) nearest model points $\{\mathbf{p}_i\}$. In an iterative framework, a new estimate of $\mathtt{R}$ and $\mathbf{t}$ should be computed using the approximation, and then D_M^2 should be minimized again to find new model points $\mathbf{p}_i$. This follows exactly the two iterative steps of current ICP algorithms. Two different approximations are offered below. These approximations result in two different algorithms, the first a generalization of ICP algorithms that use point-to-point distance in pose refinement and the second a generalization of ICP algorithm that use normal distance in pose refinement.

4.1 C-ICP1

The first approximation replaces $D_M^2(\mathbf{q}_i, \mathtt{S_i}; \mathtt{f}; \mathtt{R}, \mathbf{t})$ in (17) with the Mahalanobis distance between $\mathbf{q}_i$ and the current closest point. The summation in (17) becomes

$$\sum_i (\mathtt{R}\,\mathbf{p_i} + \mathbf{t} - \mathbf{q_i})^{\mathtt{T}}\ \mathtt{S_i}^{-1}\ (\mathtt{R}\,\mathbf{p_i} + \mathbf{t} - \mathbf{q_i}). \tag{18}$$

We refer to the iterative algorithm that uses Procedure D_M^2 to find the model surface points $\mathbf{p}_i$ and uses (18) in pose refinement as C-ICP1 (Covariance-based ICP, Algorithm 1).

Algorithm C-ICP1 is a generalization of the original Besl-McKay ICP algorithm [4] to use Mahalanobis distances in matching and pose refinement. Like the Besl-McKay algorithm, convergence of C-ICP1 is easily proved: (1) Mahalanobis distances are being reduced at each step (i.e. each time Procedure D_M^2 is run and each minimization of (18)), and (2) the model points $\mathbf{p}_i$ are always kept on the model constraint surface. There are important disadvantages to C-ICP1, however. First, like Besl-McKay algorithm, which requires 50 or more iterations, it will be slow to converge. Second and more important, singularities in the covariance matrices can not be tolerated. Procedure D_M^2 adapts to singularities by ensuring that the error vector $\mathtt{R}\ \mathbf{p_i} + \mathbf{t} - \mathbf{q_i}$ is entirely in the column space of $\mathtt{S_i}$. Unfortunately, this does not carry over to (18) because as it is written, simultaneously keeping all error vectors in the appropriate column spaces as $\mathtt{R}$ and $\mathbf{t}$ change is not possible.

4.2 C-ICP2

The second approximation, and the one we adopt, is based on linearizing $D_M^2(\mathbf{q}_i, \mathtt{S_i}; \mathtt{f}; \mathtt{R}, \mathbf{t})$ around the model point $\mathbf{p}_i$ found by Procedure D_M^2. The linearization effectively replaces $D_M^2(\mathbf{q}_i, \mathtt{S_i}; \mathtt{f}; \mathtt{R}, \mathbf{t})$ in the summation of (17) with the linear Mahalanobis distance, $D_p^2(\mathbf{q}_i, \mathtt{S_i}; \mathbf{p_i}, \hat{\boldsymbol{\eta}}_\mathtt{i}; \mathtt{R}, \mathbf{t})$:

$$\sum_i \frac{[\hat{\boldsymbol{\eta}}_i^T(\mathbf{p}_i + \mathtt{R}^\mathtt{T}\,\mathbf{t} - \mathtt{R}^\mathtt{T}\,\mathbf{q_i})]^2}{\hat{\boldsymbol{\eta}}_i^T\,\mathtt{R}^\mathtt{T}\,\mathtt{S_i}\,\mathtt{R}\,\hat{\boldsymbol{\eta}}_\mathtt{i}}. \tag{19}$$

This linearization of the constraint surface is only used to update estimates of $\mathtt{R}$ and $\mathbf{t}$. These revised estimates are then used to reestimate the closest model surface points $\mathbf{p}_i$ via Procedure D_M^2, which are in turn used in a new linearization of the constraint surface.

Before showing how to reestimate $\mathtt{R}$ and $\mathbf{t}$ from (19), several aspects of the approximation are important to consider.

- It is immediate from the definition that the approximation error is second-order. With this in mind we may identify and consider three sources of significant error caused by the approximation: large data point to model surface distances, large changes in $\mathtt{R}$ and $\mathbf{t}$, and high curvature in the model surface. The first, caused mostly by sensor measurement outliers, is controlled through robust estimation [16, 31, 34, 38]. The second is controlled by taking small steps in the minimization of Equation 19 before switching back to updating the closest points $\mathbf{p}_i$ using Procedure D_M^2. The third, which is only a significant concern for models with a substantial number of regions of high curvature, may be controlled by some combination of (a) small step sizes, (b) down-grading the influence of high-curvature regions through weighting, and (c) solving registration in a coarse-to-fine manner used smoothed versions of the model at coarse levels.

- The numerator of each term in (19) is simply the perpendicular distance from the inversely transformed data point (i.e. transformed back into the model coordinate system) to the planar surface determined by $\mathbf{p}_i$ and $\hat{\boldsymbol{\eta}}_i$. If the covariance matrices are isotropic, so that $\mathtt{S_i} = \sigma_\mathtt{i}^2\mathtt{I}$ then the denominator of each term reduces to just σ_i^2. In the further simplification that $\sigma_i = \sigma$ for all data points, the algorithm simplifies to the method proposed by Chen and Medioni [7].[4] This provides a new derivation of these "normal distance ICP" algorithms in terms of an underlying objective function to be minimized.

- To understand the expression in the denominator when the noise is not isotropic, first write $\mathtt{S_i}$ in terms of its spectral decomposition:

$$\mathtt{S_i} = \begin{bmatrix}\boldsymbol{\Gamma}_1\ \boldsymbol{\Gamma}_2\ \boldsymbol{\Gamma}_3\end{bmatrix}\ \mathrm{diag}(\sigma_1^2, \sigma_2^2, \sigma_3^2)\ \begin{bmatrix}\boldsymbol{\Gamma}_1^T\\ \boldsymbol{\Gamma}_2^T\\ \boldsymbol{\Gamma}_3\end{bmatrix},$$

where the $\boldsymbol{\Gamma}_j$'s are the unit eigenvectors (component directions) of $\mathtt{S_i}$ and

[4]The only difference is that in the original description, two range data sets were being registered and the model surface was estimated from one of the data sets.

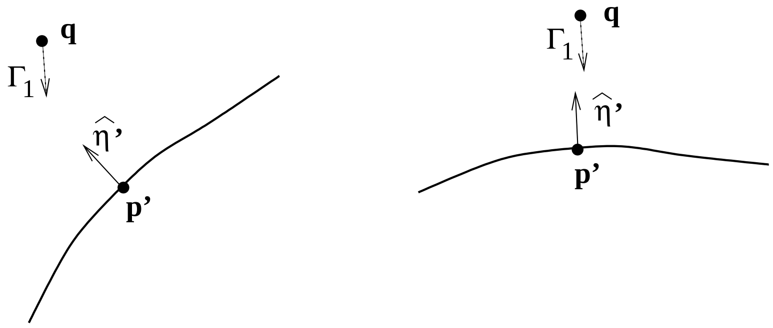

Figure 4: When the angle between the model surface normal rotated into data coordinates and the principle error direction is larger (left) the denominator of (19) and (20) will be smaller. Conversely, when the angle is smaller (right) the denominator term will be larger.

$\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq 0$. Then, it is easily seen that

$$\hat{\boldsymbol{\eta}}_i^T \, \mathtt{R}^\mathtt{T} \, \mathtt{S_i} \, \mathtt{R} \, \hat{\boldsymbol{\eta}}_\mathtt{i} = \sum_{\mathtt{j=1}}^{3} (\sigma_\mathtt{j} \boldsymbol{\Gamma}_\mathtt{j}^\mathtt{T} \, \mathtt{R} \, \hat{\boldsymbol{\eta}}_\mathtt{i})^2$$

In other words, the denominator is computed by projecting the rotated normal onto each component direction and scaling by the component variance. As $\sigma_2/\sigma_1 \to 0$, the denominator reduces to just the projection onto the primary error direction. The denominator is therefore larger for rotated normals nearly aligned with the primary error direction and smaller for rotated normals nearly perpendicular to this direction. See Figure 4.

- Singularities in $\mathtt{S_i}$ are handled naturally because $\mathtt{S_i}$ need not be inverted.

4.2.1 Updating $\mathtt{R}$ and $\mathbf{t}$ from (19)

Updating $\mathtt{R}$ and $\mathbf{t}$ is most convenient in the model coordinate system. Using the definitions $\mathbf{q}_i' = \mathtt{R}^\mathtt{T}(\mathbf{q}_\mathtt{i} - \mathbf{t})$ and $\mathtt{S}_\mathtt{i}' = \mathtt{R}^\mathtt{T}\mathtt{S_i}\mathtt{R}$ as in Section 3.1, the incremental translation and rotation to be estimated are $\boldsymbol{\Delta}\mathbf{t}$ and $\Delta\mathtt{R}$ in

$$\sum_i \frac{[\hat{\boldsymbol{\eta}}_i^T (\mathbf{p}_i + \boldsymbol{\Delta}\mathbf{t} - \Delta\mathtt{R}^\mathtt{T}\mathbf{q}_\mathtt{i}')]^2}{\hat{\boldsymbol{\eta}}_i^T \, \Delta\mathtt{R}^\mathtt{T} \, \mathtt{S}_\mathtt{i}' \, \Delta\mathtt{R} \, \hat{\boldsymbol{\eta}}_\mathtt{i}}. \tag{20}$$

Using these, the new estimates of rotation and translation are

$$\mathtt{R}\,\Delta\mathtt{R} \qquad \text{and} \qquad \mathtt{R}\,\Delta\mathtt{R})\,\boldsymbol{\Delta}\mathbf{t} + \mathbf{t}.$$

The difficulty in estimating $\boldsymbol{\Delta}\mathbf{t}$ and $\Delta\mathtt{R}$ from (20) is the non-linearity caused by the appearance of $\Delta\mathtt{R}$ in the denominator. One solution therefore is to approximate the denominator with $\hat{\boldsymbol{\eta}}_i^T \, \mathtt{S}_\mathtt{i}' \, \hat{\boldsymbol{\eta}}$, linearizing the estimation problem.

The approximation introduces a small error in the rotation of $\hat{\boldsymbol{\eta}}_i$ prior to projecting onto the components of $\mathtt{S_i}$ (see discussion above and Fig. 4). Making this approximation, the denominator effectively becomes a weight,

$$w_i = 1/\hat{\boldsymbol{\eta}}_i^T \, \mathtt{S}'_{\mathtt{i}} \, \hat{\boldsymbol{\eta}} \tag{21}$$

and the summation becomes

$$\sum_i w_i [\hat{\boldsymbol{\eta}}_i^T (\mathbf{p}_i + \boldsymbol{\Delta}\mathbf{t} + \Delta\mathtt{R}\mathbf{q}'_{\mathtt{i}})]^2. \tag{22}$$

Intuitively, this weight corrects the normal distance to be closer to distance along the principle error direction (Fig. 4). $\Delta\mathtt{R}$ and $\boldsymbol{\Delta}\mathbf{t}$ are calculated updated from (20) using well-known methods. Further refinement of these incremental estimates may be obtained by recomputing the weights and reestimating $\boldsymbol{\Delta}\mathbf{t}$ and $\Delta\mathtt{R}$, similar to what's done in several motion estimation algorithms [42, 45].

A second approach to handling the non-linearity in (20) is to use gradient descent or a Levenberg-Marquardt procedure. Parameterizing $\Delta\mathtt{R}$ using a small angle approximation, the gradient of (20) is straightforward and efficiently computable, especially when evaluated at $\boldsymbol{\Delta}\mathbf{t} = \mathbf{0}$ and $\Delta\mathtt{R} = \mathtt{I}$. Once the gradient direction is determined, (20) may be evaluated at several steps along the gradient vector to locate the minimum.

This suggests an overall algorithm with two main phases. The first phase alternates (a) Procedure D_M^2 to update the closest model points $\mathbf{p}_i$ with (b) the weighted linearization (perhaps run for several reweighting steps) to update estimates of $\mathtt{R}$ and $\mathbf{t}$. After the first phase converges, the second phase replaces the weighted linearization with the gradient based procedure. This is then run in conjunction with Procedure D_M^2 until final convergence. This final convergence should only require one or two steps of alternating Procedure D_M^2 (for each data point) with the gradient procedure.

4.2.2 Initialization

Initialization is an important consideration both when Procedure D_M^2 is first invoked at the start of C-ICP2 and in restarting Procedure D_M^2 after each refinement to $\mathtt{R}, \mathbf{t}$. Given an initial transformation estimate, a simple initialization method is to form a line from each inverse transformed data point $\mathbf{q}'_i$ and principle error direction $\boldsymbol{\Gamma}'_{i,1}$, and intersect this line with the model surface. Reinitialization of Procedure D_M^2 following refinement of the transformation estimate is also straightforward. For non-singular covariance matrices, the previous matched model point $\mathbf{p}_i$ serves as an appropriate $\mathbf{p}_{i,0}$. For singular matrices, if $\mathbf{q}'_i - \mathbf{p}_i$ is not in the column space of $\mathtt{S}'_{\mathtt{i}}$ (with $\mathbf{q}'_i$ and $\mathtt{S}'_{\mathtt{i}}$ recomputed from the new pose), then again the line through $\mathbf{q}'_i$ in direction $\boldsymbol{\Gamma}'_{i,1}$ may be intersected with the model surface.

4.2.3 Robust Estimation

Robustness to outliers is crucial in registration because measurement errors and measurements from background surfaces are unavoidable. When reasonable initial transform estimates are available, the best choice of robust technique is an M-estimator (see discussion in [38]), which has the dual advantages of downgrading or completely eliminating the influence of points with large Mahalanobis distances, while sacrificing little of the statistical efficiency of square error norms for points with small residuals. The well-known susceptibility of M-estimators to "leverage points", which causes their low breakdown point, is not an issue here because of the initial pose estimate.

The M-estimator is used in refining the pose estimate, so that the linearized pose estimation equation becomes

$$\sum_i \rho(\sqrt{w_i}[\hat{\boldsymbol{\eta}}_i^T(\mathbf{p}_i + \boldsymbol{\Delta}\mathbf{t} + \Delta\mathrm{R}\mathbf{q}_i')]/\sigma). \tag{23}$$

We have effectively used both Cauchy [20] and Beaton-Tukey biweight [2] ρ functions. Several aspects of this objective function are important to consider

- The "weight" term w_i (21) is included in the argument of $\rho(\cdot)$ because it is part of the distance computation.
- The introduction of the scale term σ may appear strange at first because Mahalanobis distances are normalized. Data-to-model error distances are not only caused by measurement errors, however. They are also caused by registration errors and discrepancies between the model and the actual object. Initially these errors will be large, but as the algorithm converges they will be greatly reduced. Hence, the scale parameter σ must be (robustly [20]) reestimated one or more times during C-ICP2. It must be fixed before the algorithm is allowed to converge.
- The solution to (23) is based on iteratively reweighted least-squares (IRLS) [20]. The robust weight function $w_\rho(u) = \rho'(u)/u$, where $u = \sqrt{w_i}[\hat{\boldsymbol{\eta}}_i^T(\mathbf{p}_i + \mathbf{q}_i')]/\sigma$, is computed for each point based on the current pose and match and is then used to scale each term in (20). These robust weights are also used in the gradient phase of Algorithm 2.

With the addition of robust weighting, the description of C-ICP2 is now complete.

5 Algorithms for Unidirectional Errors

Two new registration algorithms may be derived in the special case that the error covariance matrices are concentrated along a single direction. Unidirectional errors of this sort are typical of triangulation-based range sensors. Structured light sensors [1, 21, 36] work by recording black and white patterns of light projected from a light source and off scene surfaces. By projecting a variety of patterns, a bit vector may be formed at each pixel which encodes the position of the scene surface along the backprojection from the pixel. Measurement error (as opposed to system calibration error), therefore, is predominantly along this backprojection. Using more traditional stereo sensors, where depth calculation depends on point, edge or line locations detected in one or more images, the ratio of the depth to the intercamera distance predominates in determining the error distribution. When this ratio is large, which is typical of most sensors, the error distribution is dominated by the depth direction [5, 30].

Two new algorithms are described here. The first, a specialization of C-ICP2, is a general method for handling unidirectional errors. This second, which is substantially different and much faster, depends on having the range data represented as a range image with an associated projection function $\mathcal{P}$. These two algorithms are simpler to implement, faster and, for predominantly unidirectional errors, more accurate than current ICP algorithms.

5.1 C-ICP3

C-ICP3 introduces two simplifications over C-ICP2. The first and most important is to replace *Procedure* D_M^2 with a much simpler technique. For data point $\mathbf{q}_i$ let $\mathbf{\Gamma}_i$ be the error direction, which means that $\mathtt{S_i} = \sigma_\mathtt{i}^2 \mathbf{\Gamma}_\mathtt{i} \mathbf{\Gamma}_\mathtt{i}^\mathtt{T}$. Transforming $\mathbf{\Gamma}_i$ $\mathbf{q}_i$ and from data to model coordinates yields $\mathbf{\Gamma}'_i = \mathtt{R}^\mathtt{T} \mathbf{\Gamma}_\mathtt{i}$ and $\mathbf{q}'_i = \mathtt{R}^\mathtt{T} \mathbf{q}_\mathtt{i}$, respectively. Then, the model point minimizing the Mahalanobis distance to $\mathbf{q}'_i$ must be along the line through $\mathbf{q}'_i$ in direction $\mathbf{\Gamma}'_i$:

$$\mathbf{p}(u) = \mathbf{q}'_i + u \mathbf{\Gamma}'.$$

The minimization in *Procedure* D_M^2 therefore reduces to the problem of finding the smallest $|u|$ such that $\mathbf{p}(u) = 0$. Denoting this point u_i, the matching point is $\mathbf{p}_i = \mathbf{p}(u_i)$.

The second simplification is in the denominator of the objective function of the incremental rotation and translation update equation, (20). Using $\mathtt{S_i} = \sigma_\mathtt{i}^2 \mathbf{\Gamma}_\mathtt{i} \mathbf{\Gamma}_\mathtt{i}^\mathtt{T}$ yields

$$\hat{\boldsymbol{\eta}}_i^T \, \Delta\mathtt{R}^\mathtt{T} \, \mathtt{S}'_\mathtt{i} \, \Delta\mathtt{R} \, \hat{\boldsymbol{\eta}}_\mathtt{i} = (\sigma_\mathtt{i} {\mathbf{\Gamma}'_\mathtt{i}}^\mathtt{T} \, \Delta\mathtt{R} \, \hat{\boldsymbol{\eta}}_\mathtt{i})^2.$$

Beyond this, the calculation of $\Delta\mathtt{R}$ and $\mathbf{\Delta t}$ is the same as in *C-ICP2*.

The combination of these two simplifications results in what will be referred to as algorithm *C-ICP3*.

5.2 C-ICP4

When the data set is formed into a range image and measurement errors are concentrated along the lines of sight of the pixel, a more dramatic simplification of C-ICP2 is possible. The idea behind this arises from reversing the thinking about the matching process of *Procedure* D_M^2.

Consider a point $\mathbf{p}$ on the model surface, and think about finding the closest data point to $\mathbf{p}$ based on the current transformation estimate. Define $\mathbf{p}' = R\mathbf{p} + \mathbf{t}$. Then, because errors are along the lines of sight, the closest point to $\mathbf{p}'$ (in a Mahalanobis distance sense) is the point $\mathbf{q}(u, v)$ such that (u, v) is the closest range image pixel to $\mathcal{P}(\mathbf{p}')$, the projection of $\mathbf{p}'$ onto the image plane (Fig. 5). Now, if $\Gamma(u, v)$ is the line of sight of pixel (u, v) (and the error

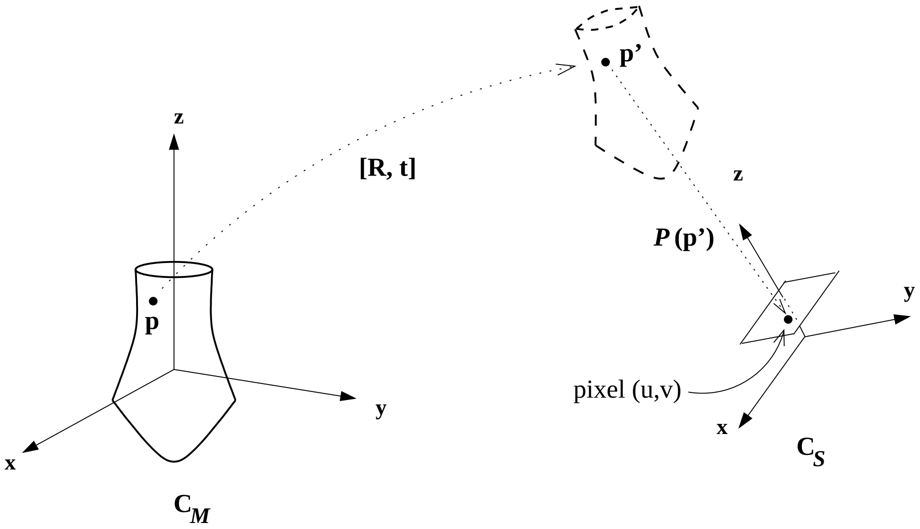

Figure 5: Matching in Algorithm C-ICP4 maps model point $\mathbf{p}$ into data coordinate using the current transformation estimate and then projects this onto the image plane. The matching data point is the point $\mathbf{q}(u,v)$ stored at the nearest pixel, (u,v).

direction for $\mathbf{q}(u,v)$), then the line through $\mathbf{q}(u,v)$ in direction $\Gamma(u,v)$ will not necessarily pass through $\mathbf{p}'$. Therefore it will not satisfy the singularity constraints of the error distribution. If the model point is considered as a model surface patch, however, with transformed normal $\boldsymbol{\eta}'$, then this line will pass through the patch and close to $\mathbf{p}'$. Furthermore, as shown above, if the patch is planar then $\mathbf{p}'$ suffices as the match to $\mathbf{q}(u,v)$, and there is no need to find the actual closest model surface point. Therefore, $\mathbf{p}'$ can serve as the closest model point to $\mathbf{q}(u,v)$. The approximation error is second-order. The significance of this is that unlike any of our previous algorithms and unlike any current ICP algorithm, there is no search involved in this closest point matching process![5]

This idea of reversing the roles of model and data to dramatically simplify the matching process leads to Algorithm C-ICP4, summarized in Fig. 6. In C-ICP4, the model is discretized into a set of planar patches represented by points and normals: $\{(\mathbf{p}_j, \hat{\boldsymbol{\eta}}_j)\}$. This computation is done off-line as a preprocessing step. On-line, given a range image, $\mathbf{Z}(u,v)$, registration proceeds with the usual iterated steps of matching and pose refinement. Matching uses the non-iterative process described above. Uniqueness in the matching may be enforced by only allowing the closest model patch for each data point — this is generally not a problem. Pose refinement requires the solution to a slightly modified version of

[5]Algorithm C-ICP3 has search in finding the intersection of the error constraint line with the model surface.

Off-line compilation of model:

Form a set of planar patches $\{(\mathbf{p}_j, \hat{\boldsymbol{\eta}}_j)\}$, from a regular sampling of the model surface. Avoid sampling in regions of high curvature.

On-line: registration

(1) Given a range image $\mathbf{Z} : (u, v) \rightarrow \mathbf{q}(u, v)$.
and an initial transformation estimate $\mathtt{R}_0$, $\mathbf{t}_0$.
(2) $k = 0$;
(3) **repeat**
(4) **for** each model patch j {
(5) $\mathbf{p}'_j = \mathtt{R}_\mathtt{k}\,\mathbf{p}_\mathtt{j} + \mathbf{t}_\mathtt{k}$;
(6) $(u, v)^T = \mathcal{P}(\mathbf{p}'_j)$;
(7) $\mathbf{q}_j = \mathbf{Z}(u, v)$;
(8) }
(9) Compute pose increments $\Delta\mathtt{R}_\mathtt{k}, \Delta\mathbf{t}_\mathtt{k}$ from
$$\sum_i \rho(\sqrt{w_i}[(\hat{\boldsymbol{\eta}}'_i)^T(\mathbf{p}'_i + \boldsymbol{\Delta}\mathbf{t} + \Delta\mathtt{R}\mathbf{q}_\mathtt{i}]/\sigma)$$
as in Algorithm C-ICP2.
(10) Update the pose estimates: $\mathbf{t}_{k+1} \leftarrow \Delta\mathtt{R}_\mathtt{k}(\mathbf{t}_\mathtt{k} + \Delta\mathbf{t}_\mathtt{k})$ and $\mathtt{R}_{\mathtt{k}+1} = \Delta\mathtt{R}_\mathtt{k}\,\mathtt{R}_\mathtt{k}$
(11) $k++$;
(12) **until** ($\Delta\mathtt{R}_{\mathtt{k}+1} \rightarrow \mathtt{I}_{3\times3}$ and $\Delta\mathbf{t}_{k+1} \rightarrow \mathbf{0}$)

Figure 6: Outline of Algorithm C-ICP4, a dramatically simplified registration algorithm applicable when the range data are stored in image format and measurement errors are predominantly along the lines of sight (backprojection lines) of the pixels.

(23). With trivial adjustments, this uses the linearization and gradient descent update techniques from Section 4.2.1 and the robust methods from Section 4.2.3.

This completes the basic description of Algorithm C-ICP4. More details and variations are considered in what follows.

5.2.1 Discretization of the Model

Model discretization requires predicting the model to data transformation, transforming the model into this predicted viewpoint, and then sampling the model surface uniformly based on the viewpoint. In effect, this creates a simulated range image from the model surface, but with surface normals in addition to points. Together, the model points and normals create a "patch set" $\{(\mathbf{p}_j, \hat{\boldsymbol{\eta}}_j)\}$. If several substantially different viewpoints are possible, then multiple viewpoint-dependent patch sets may be created. It is important to note, however, that within the same "aspect" registration does not depend significantly on the choice of model viewpoint. The only differences should be the position of the samples — all other differences are eliminated when the patch set is transformed into

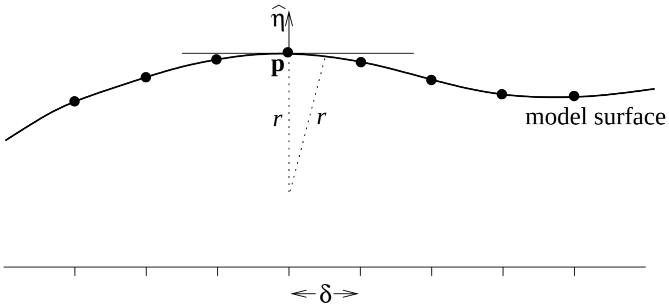

Figure 7: For sample spacing δ on the surface, the approximation error using a linear patch at model ponit $\mathbf{p}$ with radius of curvature r is at most $r - \sqrt{r^2 - (\delta/2)^2} \approx \delta^2/(2r)^2$.

the sensor coordinate system — and these differences, like differences in pixel positions within the sensor are irrelevant.

Several further issues in the discretization of the model are significant.

- Model surface sampling density should be controlled by tradeoffs between the desired level of precision in registration and the computation time. Exact formulas for the precision are difficult to obtain, aside from the obvious and unattainable lower bound variance σ_s^2/n on the transformed position of any model surface point. (Here, σ_s^2 is the variance in the data and n is the number of matches.) Hence, sampling density should be determined experimentally based on the model, the sensor, and application constraints.

- There should be no smoothing of the model surface points and normals since this smoothing can create bias in registration. If the possibility of local minima is a concern, then a coarse-to-fine hierarchy of patch sets may be created with smoothing of the model at coarse levels prior to sampling.

- Even though linear patches (points and normals) are used, the approximation error is extremely low. Suppose δ is the spacing between model patches on the model surface and suppose δ is the spacing between data points when backprojected onto the actual surface. Then, a simple analysis shows that the maximum error in discretization and in matching is approximately $\delta^2/(2r)^2$, where r is the local radius of curvature on the model surface (Fig. 7). This means higher-order approximations are generally unnecessary, even for high-precision registration. It does suggest, however, that points of extremely high curvature on the model such as sharp edges and blending regions, be avoided in the patch set of C-ICP4. This is desirable anyway, because these are often regions of large sensor error and places where model and actual objects are likely to disagree.

5.2.2 Convergence

A formal convergence proof is difficult to obtain for Algorithm C-ICP4. The problems are two-fold: changes in the transformation estimate will produce changes in matching of patches to data points, and because the patches are linear approximations these changes are not in a strict sense "smooth". The above arguments about the approximation error, however, may also be applied to show that the effects of any changes in matching are extremely small. Furthermore, as the incremental changes $\Delta \mathtt{R}_\mathtt{k}$ and $\Delta \mathbf{t}_k$ become small — inducing changes in $\mathbf{p}'_j$ significantly less than δ — there are fewer and fewer changes to the matches. As a result, with proper initialization, convergence is not a problem in a practical sense.

6 Experimental Results

[Author's note: these are just plans. The experiments should be done in a week or so.]

State that these algorithms (C-ICP4, in particular) are being used in practice, but the real test must come through simulation because ground truth is unknown. (Could put in GR&R tests for precision, though?) Compare C-ICP2 and C-ICP4 to a more traditional point-based algorithm. Assign $\sigma_3 = \sigma_2$ and vary the ratio $\sigma_2/sigma_1$. Consider the effects of variation in surface orientation, width of the surface, number of points, and magnitude of σ_1. Use a noise model that varies σ_1 in depth.

7 Summary and Conclusions

This paper has addressed the problem of registering a three-dimensional model with a range data set for applications where precision and accuracy are crucial. The main innovation is the formulation of registration as a statistical optimization problem using as an objective function the summed, squared Mahalanobis distances between data points and the transformed model surface. This builds measurement uncertainty in the form of error covariance matrices into the registration problem formulation. Two algorithms, C-ICP1 and C-ICP2, which generalize current point-based registration algorithms, arose in solving the registration optimization problem. Two further algorithms, C-ICP3 and C-ICP4, were derived for the special case of covariance matrices dominated by a single error direction. This is especially relevant to registration against range data obtained using triangulation-based sensors. The measurement errors for data from these sensors tends to be concentrated along the backprojection lines for each pixel.

Algorithm C-ICP4 is the most important in practice. It was derived for triangulation-based range sensors where the data are (or may be) stored as a dense range image. It reverses the role of model and data in the matching process, and in doing so makes matching particularly simple, requiring no iterative closest point search. As a result it is simpler, faster and more accurate than current registration algorithms.

The main limitation of the algorithms derived here is that the covariance matrices are assumed to be known in advance. This is a problem when measurement error depends on surface orientation because this orientation is not known unless it is estimated from the data or taken from the registered model. One solution is to run an algorithm such as C-ICP4 to convergence and then use surface orientations for the registered model to reinitialize the covariance matrices. Once this is complete the more general algorithm C-ICP2 may be used to refine the transformation estimate. Practically, this seems unlikely to be a major concern, (perhaps reference back to experimental results) although this intuition will need to be confirmed experimentally,

Finally, the covariance-base registration algorithms may be extended trivially to handle multiple range data sets. If the transformation between these data sets is known in advance, for example by rotating the object using a high-precision rotary stage, the multiple data sets provide simultaneous constraints on the position of the object within the stage. If the transformation between data sets is not known, registration of the model against each data set determines the calibration transformations between them.

Appendix A: Special Cases of Registration

This appendix shows that multivariate locations and linear regression are special cases of registration estimation. This means that known results about bias in these estimation problems carry over to registration. The derivations build on the problem formulation in Equation 5 and the closest point and distance equations for linear models developed in Section 3.1.

The multivariate location problem is obtained from registration by choosing the model

$$f(\mathbf{p}) = \mathbf{p} = \mathbf{0}.$$

In doing this, rotation becomes irrelevant and Equations 4 and 5 reduce to

$$\sum_i (\mathbf{t} - \mathbf{q}_i)^T \; \mathtt{S_i}^{-1} \; (\mathbf{t} - \mathbf{q}_\mathtt{i}),$$

which is equivalent to (1).

The linear regression problem is obtained from registration for linear models. For linear models, the objective function (5) reduces to

$$\sum_i \frac{[\hat{\boldsymbol{\eta}}^T(\mathbf{p}_0 + \mathtt{R}^\mathtt{T}\,\mathbf{t} - \mathtt{R}^\mathtt{T}\,\mathbf{q}_\mathtt{i})]^2}{\hat{\boldsymbol{\eta}}^T\,\mathtt{R}^\mathtt{T}\,\mathtt{S}_\mathtt{i}\,\mathtt{R}\,\hat{\boldsymbol{\eta}}}. \tag{24}$$

This may be simplified much further by choosing the plane $z = 0$, which makes $\mathbf{p}_0 = \mathbf{0}$ and $\hat{\boldsymbol{\eta}} = (0, 0, 1)^T$. In doing this, restrict attention to $\Re^2$, write

$$\mathtt{R} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix},$$

and define $\gamma = (\cos\theta, -\sin\theta) \cdot \mathbf{t}$. Making all these substitutions in (24) and simplifying yields the optimization equation

$$\sum_i \frac{(x_i \sin\theta - y_i \cos\theta + \gamma)^2}{[-\sin\theta, \cos\theta]\, \mathtt{S_i}\, [-\sin\theta, \cos\theta]^{\mathtt{T}}}. \tag{25}$$

With $\mathtt{S_i} = \mathrm{diag}(0, \sigma^2)$, this is equivalent to

$$\sum_i (x_i m - y_i + b)^2$$

where $m = \tan\theta$, $b = \gamma / \cos\theta$, and the scale term has been dropped. With $\mathtt{S_i} = \mathrm{diag}(\sigma^2, \sigma^2)$, this is equivalent to

$$\sum_i (x_i \sin\theta - y_i \cos\theta + \gamma)^2.$$

These are the ordinary and orthogonal regression objective functions as discussed in the introduction (2).

References

[1] J. Batlle, E. Mouaddib, and J. Salvi. Recent progress in coded structured light as a technique to solve the correspondence problem: A survey. *Pattern Recognition*, 31(7):963–982, 1998.

[2] A. E. Beaton and J. W. Tukey. The fitting of power series, meaning polynomials, illustrated on band-spectroscopic data. *Technometrics*, 16:147–185, 1974.

[3] R. Bergevin, M. Soucy, H. Gagnon, and D. Laurendeau. Towards a general multiview registration technique. *IEEE Trans. on PAMI*, 18(5):540–547, 1996.

[4] P. Besl and N. McKay. A method for registration of 3-d shapes. *IEEE Trans. on PAMI*, 14(2):239–256, 1992.

[5] S. Blostein and T. Huang. Error analysis in stereo determination of 3-d point positions. *IEEE Trans. on PAMI*, 9(6):752–766, 1987.

[6] G. Champleboux, S. Lavallee, R. Szeliski, and L. Brunie. From accurate range imaging sensor calibration to accurate model-based 3-d object localization. In *Proc. CVPR*, pages 83–89, 1992.

[7] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. *Image and Vision Computing*, 10(3):145–155, 1992.

[8] C. Chua and R. Jarvis. 3d free-form surface registration and object recognition. *Int. J. of Computer Vision*, 17(1):77–99, 1996.

[9] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *SIGGRAPH'96*, pages 303–312, 1996.

[10] G. J. Ettinger. *Hierarchical Three-Dimensional Medical Image Registration.* PhD thesis, MIT, 1997.

[11] O. Faugeras. *Three-Dimensional Computer Vision.* MIT Press, 1993.

[12] O. Faugeras and M. Hebert. The representation, recognition, and locating of 3-d objects. *Int. J. of Robotics Research*, 5(3):27–52, 1986.

[13] J. Feldmar, J. Declerck, G. Malandain, and N. Ayache. Extension of the ICP algorithm to nonrigid intensity-based registration of 3d volumes. *Computer Vision and Image Understanding*, 66(2):193–206, May 1997.

[14] P. Fua and Y. Leclerc. Registration without correspondences. In *Proc. CVPR*, pages 121–128, 1994.

[15] W. Grimson, T. Lozano-Perez, W. Wells, G. Ettinger, and S. White. An automatic registration method for frameless stereotaxy, image, guided surgery and enhanced reality visualization. In *Proc. CVPR*, pages 430–436, 1994.

[16] F. R. Hampel, P. J. Rousseeuw, E. Ronchetti, and W. A. Stahel. *Robust Statistics: The Approach Based on Influence Functions*. John Wiley & Sons, 1986.

[17] P. Hébert. *From Points to Shape Recovery: Reliable Geometric Primitive Extraction*. PhD thesis, Université Laval, 1994.

[18] K. Higuchi, M. Hebert, and K. Ikeuchi. Building 3-d models from unregistered range images. *Graphical Models and Image Processing*, 57(4):315–333, 1995.

[19] A. Hilton, A. Stoddart, J. Illingworth, and T. Windeatt. Reliable surface reconstruction from multiple range images. In *Proc. 4th ECCV*, pages 117–126, 1996.

[20] P. W. Holland and R. E. Welsch. Robust regression using iteratively reweighted least-squares. *Commun. Statist.-Theor. Meth.*, A6:813–827, 1977.

[21] E. Horn and N. Kiryati. Toward optimal structured light patterns. *Image and Vision Computing*, 17(2):87–97, 1999.

[22] D. P. Huttenlocher and S. Ullman. Recognizing solid objects by alignment. *Int. J. of Computer Vision*, 5(2):195–212, 1990.

[23] M. Irani, P. Anandan, and S. Hsu. Mosaic based representations of video sequences and their applications. In *Proc. IEEE Int. Conf. on Computer Vision*, pages 605–611, 1995.

[24] A. Johnson and M. Hebert. Surface matching for object recognition in complex 3-dimensional scenes. *Image and Vision Computing*, 16(9-10):635–651, July 1998.

[25] A. Johnson and S. Kang. Registration and integration of textured 3-d data. *Image and Vision Computing*, 17(2):135–147, 1999.

[26] R. Krishnapuram and D. Casasent. Determination of three dimensional object location and orientation from range data. *IEEE Trans. on PAMI*, 11(11):1158–1167, 1989.

[27] S. Lavallee and R. Szeliski. Recovering the position and orientation of free-form objects from image contours using 3d distance maps. *IEEE Trans. on PAMI*, 17(4):378–390, 1995.

[28] D. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31(3):355–395, 1987.

[29] T. Masuda and N. Yokoya. A robust method for registration and segmentation of multiple range images. *Computer Vision and Image Understanding*, 61(3):295–307, 1995.

[30] L. Matthies and S. Shafer. Error modelling in stereo navigation. *IEEE J. of Robotics and Automation*, 3(3):239–248, 1987.

[31] P. Meer, D. Mintz, A. Rosenfeld, and D. Y. Kim. Robust regression methods for computer vision: A review. *Int. J. of Computer Vision*, 6:59–70, 1991.

[32] C.-H. Menq, H.-T. Yau, and G.-Y. Lai. Automated precision measurement of surface profile in CAD-directed inspection. *IEEE Trans. on Robotics and Automation*, 8(2):268–278, 1992.

[33] T. S. Newman and A. K. Jain. A survey of automated visual inspection. *Computer Vision and Image Understanding*, 61(2):231–262, 1995.

[34] P. J. Rousseeuw and A. M. Leroy. *Robust Regression and Outlier Detection*. John Wiley & Sons, 1987.

[35] M. Rutishauser, M. Stricker, and M. Trobina. Merging range images of arbitrarily shaped objects. In *Proc. CVPR*, pages 573–580, 1994.

[36] K. Sato and S. Inokuchi. Range-imaging system utilizing nematic liquid crystal mask. In *Proc. IEEE Int. Conf. on Computer Vision*, pages 657–661, 1987.

[37] H. Sawhney and R. Kumar. True multi-image alignment and its application to mosaicing and lens distortion correction. *IEEE Trans. on PAMI*, 21(3):235–243, 1999.

[38] C. V. Stewart. Robust parameter estimation in computer vision. *SIAM Reviews*, 41(3), September 1999.

[39] A. Stoddart, S. Lemke, A. Hilton, and T. Renn. Estimating pose uncertainty for surface registration. *Image and Vision Computing*, 16(2):111–120, 1998.

[40] R. Szeliski. Video mosaics for virtual environments. *IEEE CGA*, 16(2):22–30, 1996.

[41] R. Szeliski and S. Lavallee. Matching 3-d anatomical surfaces with non-rigid deformations using octree-splines. *Int. J. of Computer Vision*, 18(2):171–186, 1996.

[42] P. Torr and D. Murray. The development and comparison of robust methods for estimating the fundamental matrix. *Int. J. of Computer Vision*, 24(3):271–300, 1997.

[43] M. Trobina. Error model of a coded-light range sensor. Technical Report BIWI-TR-164, ETH, 1995.

[44] P. Viola and W. Wells III. Alignment by maximization of mutual information. In *Proc. IEEE Int. Conf. on Computer Vision*, pages 16–23, 1995.

[45] J. Weng, T. Huang, and N. Ahuja. Motion and structure from two perspective views: Algorithms, error analysis, and error estimation. *IEEE Trans. on PAMI*, 11(5):451–476, 1989.

[46] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *Int. J. of Computer Vision*, 13(2):119–152, 1994.