RC 20560 (9/27/96)
Computer Science/Mathematics 7 pages

Research Report

Evaluating the Registration of 3D Points to 3D Lines with Application to the Pose Estimation in a Projective Image

André Guéziec and Janez Funda

IBM Research Division
T.J. Watson Research Center
Yorktown Heights, New York

IBM T.J.Watson Research Center,
P.O.Box 704, Yorktown Heights, NY 10598,
gueziec@watson.ibm.com

Research Division
Almaden · T.J. Watson · Tokyo · Zurich

Evaluating the Registration of 3D Points to 3D Lines with Application to the Pose Estimation in a Projective Image

André P. Guéziec and Janez Funda[1]
IBM T.J.Watson Research Center

Abstract

We present three iterative methods for finding an optimal rotation and translation, or 3-D displacement, that will bring a set of 3-D points at the closest possible location to a set of 3-D lines, the correspondence between points and lines being known. The main application is the pose estimation of a 3-D model given a 2-D projection, orthographic or perspective.

Each method attempts to minimize a sum of positive measures of the Euclidean distance between a point and its corresponding line. The first method is a non-linear Levenberg Marquardt minimizer. The second method uses a series of linearizations. The last method is a statistically robust minimizer.

We generate random test data consisting of a bundle of lines. We select a point on each line, perturb the set of points with a known displacement and add noise to selected points. We then compare the registration results with the three methods. The robust method is less sensitive to noise than the other two. One advantage of the linear method is that it operates on as few as three points and lines.

We study the application of these methods to the registration of 3-D medical data to projection data.

1 Introduction

We present three methods, "LM", "Linear" and "Robust" for registering 3-D points to 3-D lines, that is, finding an optimal rotation and translation that will minimize a measure of the distance between each point and a corresponding line. It is assumed that the number of points and lines is the same and the correspondence between points and lines is known. The minimum number of pairs of points is three except for the LM method for which it is six.

This work was instigated by the problem of estimating the pose of 3-D anatomical surface models and images in one or several projective 2-D images. Another application is the superimposition of 3-D models on 2-D images to achieve realistic texture mapping to the models. These applications imply that the images have been calibrated, such that a projective line can be drawn starting at each image point, connecting the point to a perspective center, or to infinity- in the case of orthographic, or parallel, projection. In the case of perspective projection, the focal length and image center must be known.

The pose estimation uses the following two steps: In the first step, points are collected in 2-D together with their corresponding 3-D point; In the second step, the 3-D points are registered to the 3-D projective lines originating at the 2-D points using one of the methods presented in this paper. Note that the lines need not have a point in common for the method to work. In fact, the method is adapted to finding a solution for data collected in several projective views simultaneously, and to coping with parallel lines that occur in orthographic projection. The process of collecting points can be done either manually or automatically. Separating the point matching process from the numerical calculations of pose estimation has several advantages: We can validate each process separately; We can compensate manually for the spurious matches that are hypothesized automatically.

Each registration method presented attempts to minimize a sum of positive measures of the Euclidean distance between a point and its corresponding line. The first two methods, LM or non-linear Levenberg Marquardt minimizer, and Linear or linearization followed by constrained minimization use the sum of squared distances. Robust, or the statistically robust minimizer weights each squared distance in the sum with a function that favors distances similar to the median of the distances. All three methods are iterative minimizers of the objective function. In order to represent the 3-D rotation, we choose the Cayley parametrization, because of the simplicity to derive a linearization and the easy determination of the rotation parameters. All three methods are fast. They execute in less than a second on a RS/6000 model 58K, when applied to fewer than 100 points, which is much more than the number of useful corresponding points that could be extracted in practice from 2-D and 3-D data. Due to its very affordable complexity, the Robust method seems capable of executing in less than a second for any practical data set.

We generate test data of a set of points and a bundle of lines. The maximum angle between the lines is comparable to the angle that we would observe when imaging an object using x-ray equipment. We corrupt some of the points with white noise and increase progressively the number of corrupted points as well as the magnitude of the noise. The LM and Linear methods typically tolerate noise that does not exceed a tenth of the diameter of the cloud of data points, as well as noise of higher magnitude provided that less than 5 or 10 % of the points are corrupted. The Robust method tolerates

[1] IBM T.J.Watson Research Center, P.O.Box 704, Yorktown Heights, NY 10598, gueziec@watson.ibm.com

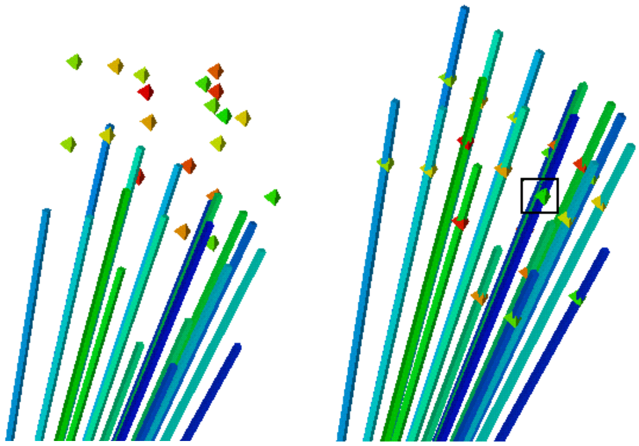

Figure 1: Configuration of points, represented as tetrahedra of different intensities, and lines, shown as thin parallelepipeds, before (Left) and after (Right) registration. Although one of the points, surrounded with a black rectangle, was perturbed with noise, the remaining points were successfully registered.

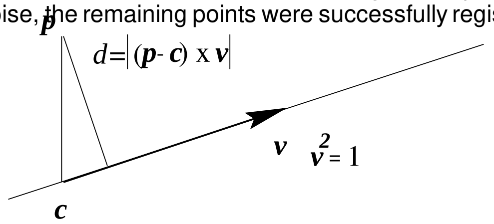

Figure 2: Euclidian distance d between a 3-D line defined with a vertex $\mathbf{c}$, a unit direction $\mathbf{v}$ and a 3-D point $\mathbf{p}$.

up to 40 % of outliers at all noise levels.

2 Previous Work

Kumar and Hanson [1] analyze different formulations for the pose estimation problem and provide a sensitivity analysis to the calibration parameters. They assume that the projection model is the perspective model and derive equations for line tokens correspondences. They provide several robust methods to weight out outliers. In addition, there is a rich literature on the problem of pose estimation using points or line tokens or more complex representation of shape. Consult for instance [2] and its bibliography.

This paper brings the following contributions: We compare the nonlinear minimization to the iterative linearization. We provide precise timing experiments. We applied our research to radiographic applications and describe a related experiment. We assume that the imagery is calibrated, but we do not rely on a perspective projection. Parallel projections or projections for which there are several centers of perspective, such as the scout views generated by CT scanners, are treated using the same formalism. The equations do not change if several views are used simultaneously to determine the pose, assuming the views are co-registered.

3 Mathematical Formulation

Our approach consists in minimizing the sum of squared distances between a set of m 3-D vertices $\{\mathbf{p_i}\}$ and corresponding m 3-D lines $\{(\mathbf{c_i}, \mathbf{v_i})\}$, passing through a given center of perspective $\mathbf{c_i}$ and having the direction of the unit vector $\mathbf{v_i}$, $\|\mathbf{v_i}\| = \mathbf{1}$. In the following, $\mathbf{Q}$ represents a 3-D rotation matrix and $\mathbf{t}$ represents a 3-D translation vector. $\mathbf{Q}$ and $\mathbf{t}$ are the unknown and all the other symbols have known values or expressions. The distance d_i between the transformed 3-D point $\mathbf{Qp_i} + \mathbf{t}$ and the ith. 3-D line $(\mathbf{c_i}, \mathbf{v_i})$ is obtained by taking the norm of the cross product between the vector joining $\mathbf{c_i}$ and $\mathbf{Qp_i} + \mathbf{t}$ and the unit vector $\mathbf{v_i}$ that specifies the direction of the line.

The problem can be written as follows:

Problem 1

$$\min_{\mathbf{Q},\mathbf{t}} \sum_{i=1}^{m} d_i^2 = \min_{\mathbf{Q},\mathbf{t}} \sum_{i=1}^{m} \|\mathbf{V_i}(\mathbf{c_i} - (\mathbf{Qp_i} + \mathbf{t}))\|^2, \tag{1}$$

where $\mathbf{V_i}$ denotes the skew symmetric matrix $\mathbf{V_i} = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix}$, implementing the vector cross product with $(v_1, v_2, v_3)^t$. Solutions exist for any number of data points and lines provided that there is at least one point. However, it is known that a minimum of three points are required to provide a unique solution [3].

One parametrization of the rotation that is convenient for our formulation is the one by Cayley [4], which we explain next. $\mathbf{U}$ is a skew symmetric matrix obtained from the vector $\mathbf{u}$, i.e.,

$$\mathbf{U} = \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix},$$

then the 3x3 matrix $\mathbf{Q} = (\mathbf{I} - \mathbf{U})(\mathbf{I} + \mathbf{U})^{-1}$ is a rotational matrix. This particular parametrization has the following advantages: It is straightforward to linearize the rotation, resulting in $\mathbf{Q} \sim \mathbf{I} - \mathbf{2U}$; When the rotation is small, the parameter values correspond to the magnitude of rotation along the Cartesian axes.

4 Non-Linear Minimization

Suppose that we consider the expression (1) as the sum of squares of m real-valued functions $f_i(\mathbf{x})$ of $n = 6$ variables, $\mathbf{x} = (u_1, u_2, u_3, t_x, t_y, t_z)^t$, with:

$$f_i(\mathbf{x}) = \mathbf{d_i} = \|\mathbf{V_i}(\mathbf{c_i} - (\mathbf{Qp_i} + \mathbf{t}))\| \tag{2}$$

We can rewrite the problem 1–equation (1)–as:

$$\min_{\mathbf{x}} \sum_{i=1}^{m} \|f_i(\mathbf{x})\|^2 = \min_{\mathbf{x}} \|\mathbf{f}(\mathbf{x})\|^2, \tag{3}$$

where $\mathbf{f}$ is the vector $(f_1, \ldots, f_m)^t$. To minimize (3), we can use the Levenberg Marquard method, which consists of switching gradually from Newton's minimization method, to the steepest descent method. Since details on these methods are available elsewhere [5], we briefly recall the principles and concentrate on providing suitable input to library routines that implement these methods.

Newton's method supposes that a quadratic approximation of $\frac{\|\mathbf{f}(\mathbf{x}+\mathbf{y})\|^2}{2}$ is available for a small $\|\mathbf{y}\|$:

$$\frac{\|\mathbf{f}(\mathbf{x}+\mathbf{y})\|^2}{2} \simeq \frac{\|\mathbf{f}(\mathbf{x})\|^2}{2} + \mathbf{g}^t\mathbf{y} + \mathbf{y}^t\mathbf{H}\mathbf{y}$$

where $\mathbf{g}$ denotes the gradient of $\frac{\|\mathbf{f}\|^2}{2}$ and $\mathbf{H}$ the Hessian, or symmetric matrix of second-order partial derivatives. By application of the chain rule, we get $\mathbf{g} = \mathbf{J}^t\mathbf{f}$, where $\mathbf{J}$ is the gradient of $\mathbf{f}$, which is a m by 6 matrix. Further differentiation leads to $\mathbf{H} \simeq \mathbf{J}^t\mathbf{J}$, provided that the error residual $\|\mathbf{f}\|$ is small with respect to $\|\mathbf{J}\|$. Newton's method consists of computing the value of $\mathbf{y}$ that will minimize the quadratic term, which corresponds to solving $\mathbf{J}^t\mathbf{J}\mathbf{y} + \mathbf{J}^t\mathbf{f} = \mathbf{0}$ For numerical reasons, it is preferable to minimize the following, which is less sensitive to roundoff errors:

$$\min_{\mathbf{y}} \|\mathbf{J}\mathbf{y} + \mathbf{f}\|$$

The Levenberg Marquardt method consists of solving:

$$(\mathbf{J}^t\mathbf{J} + \lambda \mathbf{I})\mathbf{y} + \mathbf{J}^t\mathbf{f} = \mathbf{0}$$

For a small λ it corresponds to Newton's method, and for a large λ, $\mathbf{y}$ is determined by the formula $\mathbf{y} = -\frac{\mathbf{J}^t\mathbf{f}}{\lambda}$, which corresponds to the steepest descent method, along the direction of the gradient. In practice, λ is set to a very small value at the beginning and gradually increased as the solution vector is approached.

It is necessary to provide the Levenberg Marquardt implementation with procedures for computing both $\mathbf{f}$ and $\mathbf{J}$ for any input parameters $\mathbf{x}$. $\mathbf{f}$ is easily computed by first deriving the rotation $\mathbf{Q}$ from (u_1, u_2, u_3) and by using the formula (2). The gradient requires more work: the derivatives with respect to the translation parameters are the simplest:

$$\frac{\partial f_i}{\partial \mathbf{t}} = \frac{\mathbf{V}_i^t \mathbf{V}_i(\mathbf{Q}\mathbf{p}_i - \mathbf{c}_i + \mathbf{t})}{f_i}$$

In order to compute the derivative of f_i with respect to $\mathbf{u}$, considerable simplification is obtained by computing the derivatives of the vector $\mathbf{Q}\mathbf{p}_i$ with respect to $\mathbf{u}$, which is a 3 by 3 matrix: $\frac{\partial \mathbf{Q}\mathbf{p}_i}{\partial \mathbf{u}}$. We can write:

$$\frac{\partial f_i}{\partial \mathbf{u}} = (\frac{\partial \mathbf{Q}\mathbf{p}_i}{\partial \mathbf{u}})^t \frac{\mathbf{V}_i^t \mathbf{V}_i(\mathbf{Q}\mathbf{p}_i - \mathbf{c}_i + \mathbf{t})}{f_i}$$

The method requires that the 6 by 6 matrix $\mathbf{J}^t\mathbf{J}$ be of full rank. It is thus necessary to provide at least 6 corresponding points and lines. We will see that the Linear method exposed next does not have this requirement.

5 Linearization and Constrained Minimization

A second method, which we refer to as the Linear method, consists in linearizing the rotation. The minimization must be subjected to the constraint that the rotation will be small, for the linearized equation to be valid.

After linearizing the rotation $\mathbf{Q}$ with $\mathbf{I} - 2\mathbf{U}$, and constraining $\|\mathbf{u}\|$ to be smaller than r, real value, (such as .1, in order to guarantee that the approximation holds true up to at least 2 decimals in each rotation entry) the expression (1) becomes:

Problem 2

$$\min_{\mathbf{U},\mathbf{t}} \sum_i \|\mathbf{V}_i(\mathbf{c}_i - (\mathbf{I} - 2\mathbf{U})\mathbf{p}_i - \mathbf{t})\|^2 + \lambda(r^2 - \|\mathbf{u}\|^2) \quad (4)$$

We transform the expression (4) and introduce the skew symmetric matrices $\mathbf{P}_i$ (the relationship between $\mathbf{P}_i$ and $\mathbf{p}_i$ is the same as the one between $\mathbf{U}$ and $\mathbf{u}$):

$$\min_{\mathbf{u},\mathbf{t}} \sum_i \|\mathbf{V}_i(\mathbf{p}_i - \mathbf{c}_i + 2\mathbf{P}_i\mathbf{u} + \mathbf{t})\|^2 + \lambda(r^2 - \|\mathbf{u}\|^2), \quad (5)$$

which we can rewrite in matrix form:

$$\min \|\mathbf{A}\mathbf{x} - \mathbf{b}\| \text{ subject to } \|\mathbf{B}\mathbf{x}\| < r, \quad (6)$$

with $\mathbf{A} = \begin{bmatrix} 2\mathbf{V}_1\mathbf{P}_1 & \mathbf{V}_1 \\ \dots & \dots \\ 2\mathbf{V}_n\mathbf{P}_n & \mathbf{V}_n \end{bmatrix}$, 3nx6 matrix, $\mathbf{x}^t = (\mathbf{u}^t, \mathbf{t}^t)$, $\mathbf{b} = [\mathbf{V}_i(\mathbf{c}_i - \mathbf{p}_i)]$, 3nx1 matrix, and finally $\mathbf{B} = [\mathbf{I} \quad \mathbf{0}]$, 3x6 matrix.

This method is iterative: once $\mathbf{u}, \mathbf{t}$ have been estimated, we compute $\mathbf{Q}$ from $\mathbf{u}$, set the new 3-D points $\mathbf{p}_i := \mathbf{Q}\mathbf{p}_i + \mathbf{t}$ and solve equation (6) until the increments of $\mathbf{u}, \mathbf{t}$ are below a certain small value or until a maximum number of iterations has been exceeded. Contrary to the LM method, any number of points greater than two can be fed to equation (6). An answer could normally be obtained with 1 point only, but we have chosen to only consider problems with more equations than unknowns. The method works very well for three points and thus offers a good alternative to closed form solutions [**?**].

6 Use of a Robust M-Estimator:

In this section, we describe the third method which we refer to as the Robust method. Robust means that the method will be resistant to outliers[6]. We start with the expression minimized in (5) with the difference that each individual error term– 3-D distance point-line–is weighted by–specifically, becomes the argument of– a function ρ, such that measurements that correspond to an error exceeding the median of the errors by some factor will have a constant contribution. This formulation was introduced by Tukey, and ρ is called the Tukey weighting function.

Problem 3

$$\min_{\mathbf{u},\mathbf{t}} \Delta = \sum_i \rho(\frac{\|\mathbf{V}_i(\mathbf{p}_i - \mathbf{c}_i + 2\mathbf{P}_i\mathbf{u} + \mathbf{t})\|}{s}) \quad (7)$$

where s is a scale factor. Let us note $e_i = \|\mathbf{V}_i(\mathbf{p}_i - \mathbf{c}_i + 2\mathbf{P}_i\mathbf{u} + \mathbf{t})\|$ the error. The constant s can be defined as: $s = \frac{\text{median}(e_i)}{.6745}$. It is not necessary to know the exact expression of ρ. We will rather use the derivative of ρ, Ψ, which has the following expression [2, 1]:

$$\Psi(u) = \begin{cases} u(1 - u^2/a^2)^2 & \text{if} \|u\| \le a \\ 0 & \text{otherwise;} \end{cases}$$

Following [1], we set $a = 2$. Note that the choice of a as well as the choice of s influence the proportion of data points that will be considered outliers. Contrary to before, we cannot solve the last equation (7) directly and we need to differentiate it with respect to $\mathbf{u}, \mathbf{t}$. The first step is to differentiate e_i with respect to $\mathbf{u}$ and $\mathbf{t}$.

$$\begin{aligned} \frac{\partial e_i}{\partial \mathbf{u}} &= \frac{\partial \sqrt{(\mathbf{p}_i - \mathbf{c}_i + 2\mathbf{P}_i\mathbf{u} + \mathbf{t})^t \mathbf{V}_i{}^t \mathbf{V}_i(\mathbf{p}_i - \mathbf{c}_i + 2\mathbf{P}_i\mathbf{u} + \mathbf{t})}}{\partial \mathbf{u}} \\ &= \frac{2\mathbf{P}_i^t \mathbf{V}_i^t \mathbf{V}_i(\mathbf{p}_i - \mathbf{c}_i + 2\mathbf{P}_i\mathbf{u} + \mathbf{t})}{2e_i} \\ \frac{\partial e_i}{\partial \mathbf{t}} &= \frac{\mathbf{V}_i^t \mathbf{V}_i(\mathbf{p}_i - \mathbf{c}_i + 2\mathbf{P}_i\mathbf{u} + \mathbf{t})}{2e_i} \end{aligned}$$

At this point, we introduce the projection matrix: $\mathbf{W}_i = \mathbf{V}_i^t\mathbf{V}_i = \mathbf{I} - \mathbf{v}_i\mathbf{v}_i^t$.

The derivative of Δ with respect to the vector $\mathbf{u}$ yields, once the constant terms have been removed from the summation:

$$\begin{aligned} \frac{\partial \Delta}{\partial \mathbf{u}} &= 0 = \sum_i \Psi(\frac{e_i}{s}) \frac{\mathbf{P}_i^t \mathbf{W}_i(\mathbf{p}_i - \mathbf{c}_i + 2\mathbf{P}_i\mathbf{u} + \mathbf{t})}{e_i} \\ \frac{\partial \Delta}{\partial \mathbf{t}} &= 0 = \sum_i \Psi(\frac{e_i}{s}) \frac{\mathbf{W}_i(\mathbf{p}_i - \mathbf{c}_i + 2\mathbf{P}_i\mathbf{u} + \mathbf{t})}{e_i} \end{aligned}$$

which we can rewrite in matrix form:

$$\mathbf{C}\mathbf{x} = \mathbf{d} \tag{8}$$

with $\mathbf{C} = \sum_i \mathbf{\Psi}(\frac{e_i}{s}) \begin{bmatrix} -2\mathbf{P}_i\mathbf{W}_i\mathbf{P}_i & -\mathbf{P}_i\mathbf{W}_i \\ 2\mathbf{W}_i\mathbf{P}_i & \mathbf{W}_i \end{bmatrix}$, `6x6` matrix, $\mathbf{x}^t = (\mathbf{u}^t, \mathbf{t}^t)$,
$\mathbf{d} = \sum_i \mathbf{\Psi}(\frac{e_i}{s}) \begin{bmatrix} -\mathbf{P}_i\mathbf{W}_i(\mathbf{c}_i - \mathbf{p}_i) \\ \mathbf{W}_i(\mathbf{c}_i - \mathbf{p}_i) \end{bmatrix}$, `6x1` vector. The method can be applied to any number (≥ 1) of points. However, in the case of three points, an exact match for all points is possible and we have observed that the Linear method is more accurate, since the normal least squares equations (8) are used rather than the direct minimization.

In practice, the Linear Method -equation (6)- has been implemented with a explicit constraint of $\|\mathbf{u}\|$ being smaller than r, while the Robust Method -equation (8)- has been implemented without constraint. The technique for implementing that constraint is Generalized Singular Value Decomposition, which is explained in Golub and Van Loan [7].

7 Experimental Comparison of the Methods

To compare the LM, Linear and Robust methods, we generate a set of lines and points that simulates a typical configuration that would occur when solving a pose estimation problem specialized to radiographic imagery.

The first step is to generate the base configuration of points and lines. We first choose m random points $\{p_i\}_{i=1,\ldots,m}$ inside the unit cube: $0 \leq x, y, z \leq 1$. We assume a perspective projection. The following step is to choose a center of perspective $\mathbf{c}$, where all lines will intersect. We place $\mathbf{c}$ along the line of equation $x = y = z$. We cast a ray between $\mathbf{c}$ and each $\mathbf{p}_i$, obtaining a direction $\mathbf{v}_i$. For this experiment, the point $(-2, -2, -2)$ was chosen, which corresponds to a maximum angle for the bundle of lines of approximately 20 degrees. In order for the experiment to be independent of our particular choice for the origin and axes, we rotate the entire configuration by a rotation of random axis and of arbitrary angle (we choose 30 degrees). We then rotate the points $\{p_i\}$ only by a random rotation of 30 degrees, which is within the range of expected rotations for the pose estimation problem. We translate the points by a random vector that has the same magnitude as the points– the random vector is drawn inside the unit cube. This completes the construction of the base configuration. A base configuration of 20 points and lines is shown in Fig. 1.

The next step is to take the base configuration, register the points to the lines and compare the transformation obtained with the one that was effectively applied. In order to make that comparison, we apply both transformations to landmark points and compute the distance between the two transformed points. We choose the vertices of the unit cube as landmark points $(1, 0, 0)$, $(0, 1, 0)$ and $(0, 0, 1)$. The maximum distance, or maximum error, is chosen to measure the quality of registration.

The last step is to add noise to selected points. We add zero mean Gaussian noise to the points. The maximum standard deviation of the noise varies from 0 to 0.33, which is roughly the third of the diameter of the cloud of points. This amount of noise was rarely observed in practice. We add noise to a selected number of points, up to 50 %.

In Figs. 3, 4, and 5 we show the maximum registration error with each method. The figures show that the LM and Linear methods typically tolerate noise that does not exceed a tenth of the diameter of the cloud of data points. Noise of higher magnitude can be coped with provided that less than 5% of the points are corrupted.

While in theory, the Robust method should still work with 50 % of spurious points, in practice, such a large number of outliers is only tolerated with small noise levels, and we have observed successful registration with up to approximately 40 % of outliers corrupted with noise of magnitude a third of the diameter of the cloud of data points. Due to the iterative evaluation process for the rotation and translation, significant errors can remain at early stages of the registration, making it difficult to isolate outliers based solely upon the residual error. Valid data points are more likely to be misinterpreted as outliers, resulting in an incorrect registration.

Number Points / Method	LM	Linear	Robust
100 points (CPU time)	790	690	28
20 points (CPU time)	200	42	9.6

Table 1: Average execution times in milliseconds for the three registration methods applied to data sets that comprise 100 points (top) and 20 points (bottom).

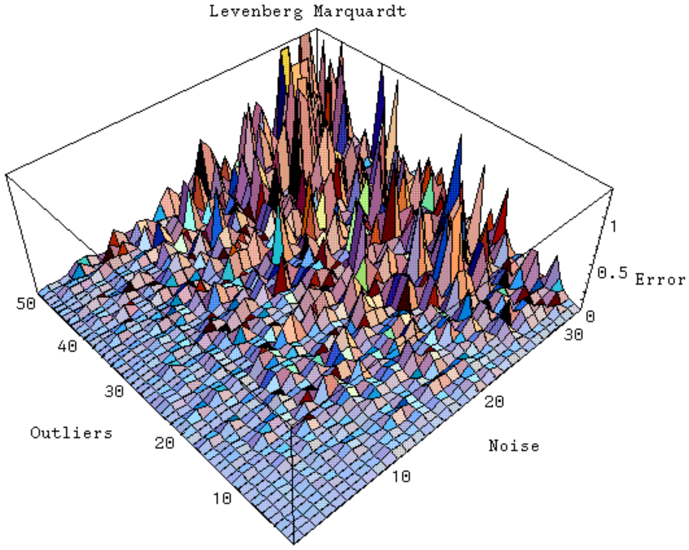

Figure 3: Maximum registration Error plotted vs Noise magnitude and percentage of Outliers for the LM (Levenberg Marquardt) method.

The following table 1 provides us with the average execution times (using an IBM RS6000) for the three different methods on data sets that comprise 100 points and 20 points respectively. All three methods execute in less than a second for datasets of fewer than 100 points, which is much more points than expected in practice, since in a practical situation, it is difficult to extract from image and model data a large number of 3-D points and lines that unambiguously correspond to each other. Due to an advantageous complexity for the numerical computation of the 6 parameters, the Robust method executes significantly faster than the other two and is only linearly dependent on the number of points. The Linear method uses a version of the Singular Value Decomposition, that has a component cubic in the number of data points, which is why the computing time grows faster than with the other methods. The LM method uses a linear system that is also 6 by 6, but has a significant overhead of computing the gradient of each of the functions. It thus starts with the highest cost, but the cost does not increase dramatically with the number of points m.

The Robust method has two parts: The first part, whose complexity is linear in the number of data points, consists in determining the median of error measurements and forming a linear system; The system, which is of size 6 by 6, is solved in the second part, in constant time. The Robust method will execute in less than a second for virtually any data set. The method, applied to 1000 points, provided registrations in an average of .25 seconds.

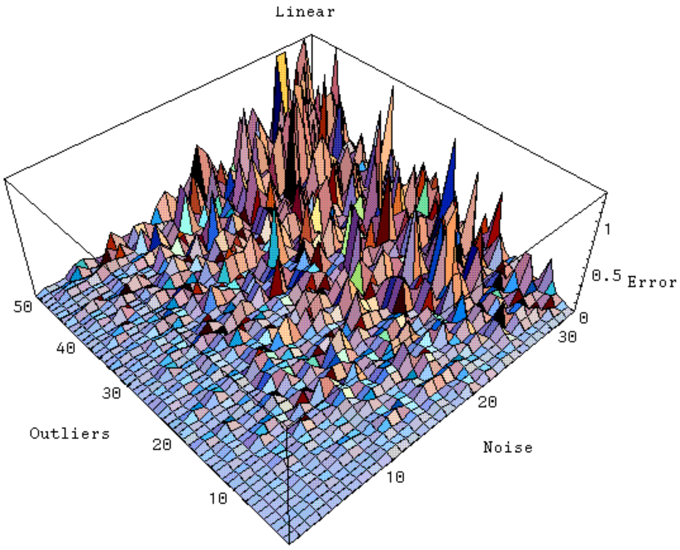

Figure 4: Maximum registration error for the Linear method.

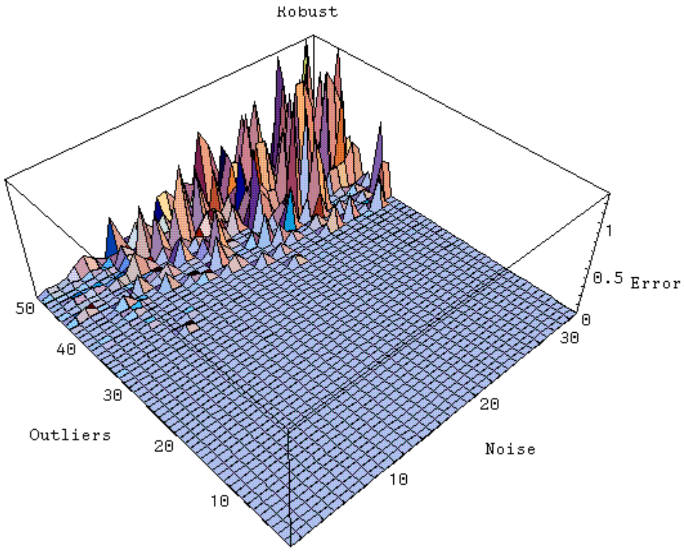

Figure 5: Maximum registration error for the Robust method.

8 Registration of a Surface Model to its Projections

The registration of a 3-D object to a 2-D projection, or *pose recovery*, consists of finding the 3-D position and orientation, or *pose* of the camera that would have produced the given projection. Alternatively, the problem can be seen as, given a predefined pose for the camera, finding the pose of the object that would have produced the projection. The availability of several projections can help to constrain the pose further, and to obtain a more accurate estimate. We assume here that the internal calibration parameters of the camera are known. For instance, for a perspective projection, the focal length, image center and scale parameters are known. If there is distorsion, it has been compensated for.

In Medical Imaging, a related goal is to register 3-D volumetric data, such as a CT-scan, to a 2-D radiographic image. Such a registration can be used for instance to correlate a surgical plan based upon the 3-D CT-scan data with the accurate location of a patient on the operating table. An on-going joint study between IBM and Integrated Surgical Systems, CA, aims to determine whether a radiography based localization could be used instead of implanting metallic fiducial markers [8].

The Robust method was used for testing the registration of a surface model to two projections. We built a surface model of a human femur starting from a CT scan of a dry bone. The image of the bone was separated from the image of a fiducial system that had been attached to it. An isosurface was then extracted, containing 100K vertices. The surface was then approximated with a triangulated surface containing 1500 vertices.

In order to simulate radiographs taken through the femur model, we then generated two perspective profiles of the femur model with different camera orientations, differing by 40 degrees, which we call *target profiles*. The size of the model, which was extracted from a CT-scan, reflected the size of a typical human femur. To model the camera, we selected a focal length of 1000 mm, which is typical of a plain film radiograph. The profiles were obtained by first constructing the apparent contour of the surface as a collection of closed 3-D non planar curves, and then projecting the apparent contours on the focal planes of the cameras. Recall that the apparent contour is the locus of all the points of the surface such that the tangent plane and viewing direction are parallel. On a triangulated surface with certain regularity properties, the apparent contour can be defined as a set of closed and oriented curves.

The pose was then perturbed with a series of random rotations and translations. For each perturbation, we were able to relax the pose of the surface to its original location. The algorithm we used was the following: Using the current pose, we generated two profiles, one for each view, which we call *transient profiles*. These transient profiles were superimposed onto the target profiles. For each point of the target profiles, we computed the closest point on the transient profile, thus hypothesizing correspondences. The correspondences found were then filtered to retain the ones that globally contributed to the same 2-D transformation.

As the target profiles were perfectly calibrated, we could cast a ray in space from the center of perspective to the profile, and as the transient profiles had been projected from the surface, we could save the position on the apparent contour of the corresponding surface point. Hence, for each transient profile, we reduced the problem to registering 3-D surface points to 3-D lines. The points and lines were fed to the Robust algorithm, and the transformation found was applied to the surface.

The final solution, shown in Fig. 6 and Fig. 7, was obtained by iterating on the above process until the transformation found by the Robust algorithm corresponded to the identity transformation.

A total of sixty different rotations were applied with a random axis and angles ranging from ten to sixty degrees. The registration accuracy was measured by selecting three points on the surface, two on the condyles and one on the trochanter, in order to simulate

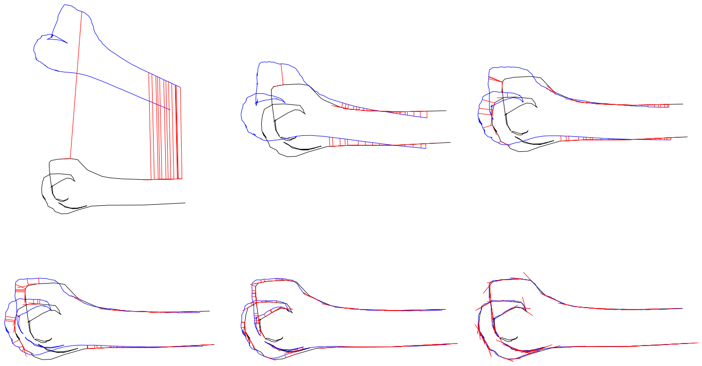

Figure 6: Snapshots of the surface to profile registration, observed from the camera that was used to generate the first profile.

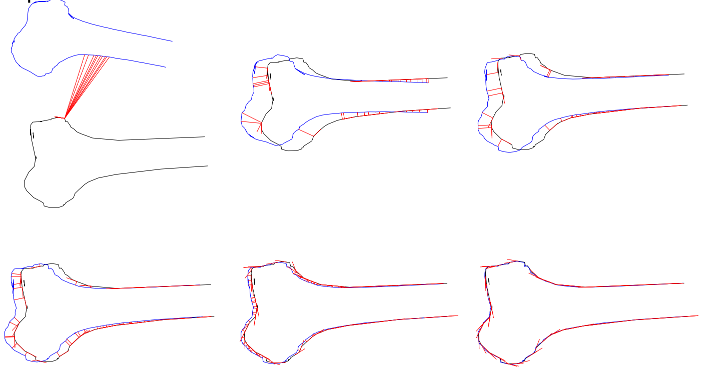

Figure 7: Snapshots of the registration taken from the second viewpoint.

the metallic pins used for clinical procedures. We then compared the original position of these "markers" to the final one obtained by applying the perturbation plus the transformation found by our algorithm. The largest distance between corresponding markers, reflecting the maximum registration error, was reported on a graph. As for most initial rotations, the maximum error was well below one mm, some misregistrations could be observed for perturbations exceeding 30 degrees of rotation, which are unlikely to be observed in practice.

One of the advantages of generating profiles and using correspondences between 3-D lines and apparent contours of the surface is that the registration can be performed fast: 13 seconds on a IBM RS6000 Workstation, representing the processing of both views and the creation of snapshots. Again, the overall registration process corresponds to solving many instances of the lines to points registration problem. Another advantage is that it is very easy to assess visually on the 2-D projection whether the registration is accurate, whereas a 3-D view can be useful at providing three dimensional spacial relationships. Note that the registration error is measured in 3-D using the final pose, lines and points.

9 Conclusion

We have presented three numerical methods for registering a set of 3-D points to a set of 3-D lines when the correspondence between the different points and lines is known in advance. The first method is a non-linear minimizer using the Levenberg Marquardt algorithm, which is a natural choice for solving such a problem. The second method that uses a linearization of the equations allows to constrain the solution to apply a small rotation at each iteration. It also works on as few as three points and lines. The last method is a statistically robust method.

We have generated test data with various noise levels and a varying proportion of points contaminated with noise. The robust method was the most successful in finding the accurate rotation and translation with up to 40 % of spurious points. Although all methods will require less than a second of CPU time on a workstation with most datasets, the robust method is by far the fastest, and scales at most linearly with the number of input points and lines.

These methods have been directly applied to the registration of Computed Tomography data to a surgical robot in [8].

REFERENCES

[1] Rakesh Kumar and Allen R. Hanson. Robust methods for estimating pose and a sensitivity analysis. *CVGIP-IU*, 60(3):313–342, 1994.

[2] Robert M. Haralick, Hyonam Joo, Chung-Nan Lee, Xinhua Zhuang, Vinay G. Vaidya, and Man Bae Kim. Pose estimation from corresponding point data. *IEEE Transactions on Systems, Man and Cybernetics*, 19(6):1426–1446, november/december 1989.

[3] Martin A. Fishler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Graphics and Image Processing*, 24(6):381–395, June 1981. ACM.

[4] Hermann Weyl. *The Classical Groups*. Princeton, 1939.

[5] P. Gill, W. Murray, and M. Wright. *Practical Optimization*. Academic Press, 1993.

[6] Peter J. Huber. *Robust Statistical Procedures*. SIAM, Eidgenössische Technische Hochschule Zürich, 1977.

[7] Gene H. Gulub and Charles F. Van Loan. *Matrix Computations*, chapter 12, pages 562–564. The Johns Hopkins University Press, 1993.

[8] A.P. Guéziec, P. Kazanzides, R.H. Taylor, and B. Williamson. Registration of computed tomography data to a surgical robot using fluoroscopy: A feasibility study. submitted to CVRMEDII-MRCASIII.