

Techniques for Fast and Accurate Intra-Surgical Registration

David A. Simon, Martial Hebert, Takeo Kanade

das@ri.cmu.edu, hebert@ri.cmu.edu, tk@ri.cmu.edu

Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
Phone: (412) 268-6880

Abstract

The goal of intra-surgical registration is to establish a common reference frame between pre-surgical and intra-surgical 3-D data sets that correspond to the same anatomy. This paper presents two novel techniques which have application to this problem: high-speed pose tracking, and intra-surgical data selection. In the first part of this paper, we describe an approach for tracking the pose of arbitrarily-shaped rigid objects at rates up to 10Hz. Static accuracies on the order of 1mm in translation and 1 degree in rotation have been achieved. We have demonstrated the technique on a human face using a high-speed VLSI range sensor; however, the technique is independent of the sensor used or the anatomy tracked. In the second part of this paper, we describe a general purpose approach for selecting near-optimal, intra-surgical registration data. Due to high costs associated with the acquisition of intra-surgical data, it is desirable to minimize the amount of data acquired, while ensuring that registration accuracy requirements are met. We synthesize near-optimal intra-surgical data sets, based upon an analysis of differential surface properties of pre-surgical data. We demonstrate, using data from a human femur, that discrete point data sets selected using our method provide superior pose refinement accuracy to those selected by human experts.

Keywords: 3-D Registration, Pose Estimation, High-Speed Pose Tracking, Geometric Constraint-Based Data Selection

1 Introduction

A growing number of surgical procedures require the establishment of a common reference frame between pre-surgical volumetric data and the corresponding patient anatomy. This requires the solution of the 3-D registration problem. Once a common reference frame is determined, pre-surgical data can be used in tasks such as: guiding robotic tool movements [15][30], guiding or constraining a surgeon's tool movements [5][16][22][24], superimposing graphical overlays of internal anatomy upon a surgeon's view of the patient [6], or guiding the position of radiosurgical equipment [27].

Current approaches to 3-D registration in medicine require manual specification of corresponding points in pre- and intra-surgical data sets [5]. Establishing correspondence is simplified by patient-attached fiducial markers, the locations of which can be extracted from both data sets. Recent approaches to 3-D registration attempt to eliminate the need for fiducials and manual specification of correspondences by using features which are intrinsic to the data. For example, researchers have attempted to match intrinsic features such as: bounding contours to surfaces [17], ridge lines to ridge lines [8], surfaces to surfaces [12], and discrete points to surfaces [6][16][22][28].

When the patient anatomy being registered is fixed in space, 3-D registration need be performed only once to establish a common reference frame. However, a recent trend in medical technology has been to reduce the need for invasive and uncomfortable fixturing devices (e.g. stereotactic neurosurgical frames). Without

the use of rigid fixation, the relevant anatomy is free to move. Thus, it becomes necessary to periodically register the relevant anatomy. The first part of this paper describes a technique capable of 3-D registration at rates of 5 to 10 iterations per second. While other researchers have addressed this problem [6][31], to our knowledge none have demonstrated sub-second performance without the use of fiducial markers.

We have demonstrated the high-speed registration capability by tracking the pose (position and orientation) of human faces. While we have not yet applied the technique to a clinical problem, we feel that there is great potential for this method to be used in medical applications. Three requirements for the use of this method are: 1) the ability to construct a polygonal mesh representation of the bounding surface of the relevant anatomy from pre-surgical data, 2) the ability to acquire, at high speed, 3-D data of the bounding surface of the anatomy during surgery, and 3) the ability to determine an approximate pose estimate for initialization. We have demonstrated the technique with a prototype high-speed VLSI range sensor developed at CMU [7]. As other high-speed 3-D sensors capable of intra-surgical use become available (e.g. GE's interventional MRI, high-speed X-ray imagers, real-time range from focus sensors [21]) it should be possible to adapt our technique to such sensors.

The second topic addressed in this paper is the selection and acquisition of intra-surgical data used in 3-D registration. Sensors which have been used for intra-surgical data acquisition include coordinate measuring devices [16][22][25], and X-ray imagers [17]. With these sensors, there are variable costs associated with the acquisition of new data. Each additional unit of data acquired (i.e. 3-D data point, X-ray view) expends time, and in the case of X-rays, increases the radiation to which the patient is exposed. Therefore, it is desirable to minimize the amount of intra-surgical data required to perform 3-D registration, without sacrificing accuracy of the resulting pose estimate.

Towards this goal, we have developed a technique for analyzing the *geometric constraint* between two data sets. We have demonstrated that there is a strong correlation between this geometric constraint, and the accuracy which results from registering the data sets. To automatically generate data sets which result in good registration accuracy, we have developed a synthesis technique which maximizes a measure of geometric constraint. We have empirically shown that the registration accuracy which results from synthesized data sets is comparable to that resulting from the best manually selected data.

Recently, two research groups have described systems which employ surface-based registration techniques for an orthopaedic application. Lavalley et al. [16] and Nolte et al. [22] both describe systems for planning and executing the insertion of screws into the pedicle component of human vertebrae. Both employ registration techniques similar to that described in this paper. In particular, they register 3-D surfaces that were derived from pre-surgical CT images, to discrete point data from a coordinate measuring device. In addition, Nolte et al. perform an excellent validation of the errors resulting from registration by comparing their surface-based results to a high accuracy approximation of ground truth. It is our hope that the data selection technique presented in this paper will ultimately be useful for improving the registration accuracies in applications such as these.

2 Discussion

2.1 Surface-Based Registration

A framework for surface-based, intra-surgical registration is outlined in Figure 1. The goal is to determine the best possible alignment between a surface extracted from pre-surgical volumetric data, and a set of data collected during surgery. The intra-surgical data considered in the following discussion are discrete 3-D points, such as those collected by a coordinate measuring device or range sensor. The registration technique, however, can easily be generalized to other types of intra-surgical data.

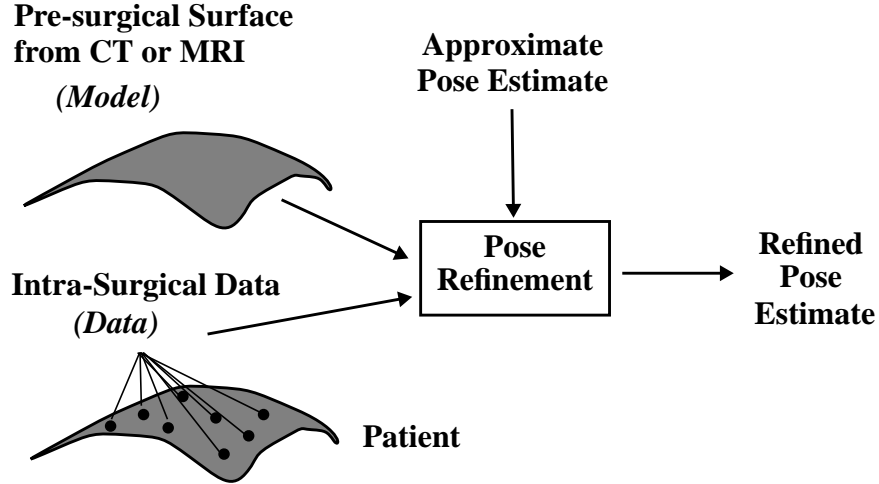


Figure 1: Surface-Based Registration

In Figure 1, we assume that an approximate pose estimate is available during surgery via coarse anatomical landmark correspondence. This estimate will be used as a starting point in the registration process. Thus, we will refer to the mechanism for performing registration as *pose refinement*.

Most approaches to geometric pose refinement attempt to minimize a least square error metric such as:

$$\min_{R, T} \sum_i \|M_i - (RD_i + T)\|^2 \quad (1)$$

where D_i represent points in the *Data* set, M_i represent points in the *Model* set, and R and T are a rotation and translation respectively which minimize the expression. In this paper, the D_i correspond to the discrete 3-D points collected during surgery, while the M_i correspond to 3-D points on the pre-surgical surface.

In fiducial-based approaches to registration, the correspondences between the D_i and M_i are assumed to be known; each (D_i, M_i) pair corresponds to the same fiducial. Given this correspondence, there are several techniques for finding R and T which minimize the least square error in (1) [3][9][11].

In fiducial-less approaches to registration, the (D_i, M_i) correspondences are unknown a priori. An approach for estimating pose despite these unknown correspondences was introduced in a paper by Besl and McKay [2]. Below is an overview of the iterative closest point (ICP) algorithm presented in that paper:

1. For each point D_i in the *Data* set, compute the *closest* point (Euclidean distance) M_i which lies on the surface of the *Model* set.
2. Using the correspondences from step 1, find R and T which minimize the equation (1) via the method described in [3].
3. Apply the incremental transformation from step 2 to all points in the *Data*.
4. If the relative changes in R and T are less than a threshold, ϵ , terminate, else goto 1.

The ICP algorithm works quite well, especially when an approximate pose estimate is available for initialization. In general there is no guarantee that ICP will converge to the global minimum, however we have found convergence to be very good in practice. Techniques exist for finding the global minimum when non-global convergence is a problem [2]. When outliers are present in the data (i.e. points in the *Data* set

for which there is no correspondence in the Model), additional processing may be necessary [18][32]. We have employed outlier detection similar to that described in [32] for our high-speed registration work.

The ICP algorithm has provided us with a basic pose refinement capability which we have used in our work on high-speed registration, as well as in the validation of our intra-surgical data selection techniques. One benefit of the ICP algorithm is that the approach is independent of data representation. The only data representation requirement is that it be possible to calculate closest points between the two data sets. Thus, it should be straight-forward to modify our implementation of ICP to handle other data types (e.g. registering bounding contours to surfaces).

2.2 Speed Enhancements to Surface-Based Registration

Due to the simplicity of the ICP algorithm, it is well suited to high-speed implementation. In particular, unlike some other pose refinement methods [10], time consuming gradient calculations are not required. For this reason, we have been able to use ICP as the core component of a system for pose tracking of arbitrarily-shaped 3-D surfaces at rates up to 10 Hz. To perform pose tracking at high speeds, it was necessary to add several speed enhancements to the basic ICP algorithm. Each of these enhancements: kd-trees, closest point caching, fast surface point computation, and acceleration are described in the following sections.

2.2.1 Kd-trees

The most computationally expensive step in the ICP algorithm is finding the closest point sets. In general if there are N_D points in the Data set and N_M geometric entities (i.e.: points, lines, triangles) in the Model set, then the complexity of a single closest point query is $O(N_D N_M)$. However, as suggested in [2] and demonstrated in [32], this complexity can be reduced to $O(N_D \log N_M)$ by the use of a k-dimensional binary tree, or simply kd-tree [1]. The use of kd-trees for closest point computation allows us to decide at each node of a binary tree on which side of a hyperplane the closest point will lie. Thus, large regions of the search space can be pruned at each level in the search. We have implemented a closest point algorithm based on the kd-tree [4]. As demonstrated below, the use of kd-trees was the most significant factor in improving the speed of ICP execution.

2.2.2 Closest Point Caching

A second small speed improvement was realized by caching closest points. Referring to the Model set as M and the Data set as D , points in M and D which are proximal at iteration k , are highly likely to be proximal at iteration $k+1$. Thus, rather than finding the single closest point in M for a given point $D_i[k]$, we can find n closest points in M and cache these points together with the point $D_i[k]$. Note that there is little overhead involved in finding n closest points when n is a small number like 5. On the next iteration, since the point $D_i[k+1]$ is likely to be close to the point $D_i[k]$, it is also likely that the closest point in M to $D_i[k+1]$ will be one of the points cached on the previous iteration. It is possible to determine conclusively whether the closest point is contained in the cached set by performing a simple test. This test compares the magnitude of the previous incremental transformation to the distance between the closest cached point and the n th closest cached point (where n is the number of cached points). A variation on this test can also determine whether the closest point at iteration $k+1$ is the *same* as the closest point at iteration k . The overall result of caching is that closest points can often be found without requiring a full search of the kd-tree. Rather, only the points in the cached set must be tested.

A similar caching technique can be applied to *spatially* (rather than *temporally*) adjacent points. If two data points $D_i[k]$ and $D_{i+1}[k]$ are spatially proximal, then it is likely that their corresponding closest points $M_i[k]$ and $M_{i+1}[k]$ will also be spatially proximal. An analogous caching technique can be applied to this situation, however we have not yet implemented caching for spatially adjacent points.

2.2.3 Fast Surface Point Computation

Since the Model set is a triangular mesh surface, computation of the closest point requires an additional step. The output of the kd-tree based closest point algorithm will return the *vertex*, V_i , which is closest to the Data point D_i , as shown in Figure 2. Given V_i , the closest Model point M_i will lie within, or on the border of, one of the triangles to which the vertex belongs¹. In order to find M_i , it is necessary to project D_i into the planes defined by each of these triangles. The resulting projected points will either lie inside or outside of a given triangle. For each triangle, if the projected point lies inside the triangle, call this point C_k , where k is the triangle index. For projected points which lie outside of the triangle, C_k is defined as the closest point between the border of the triangle and the projected point. Finally, M_i is found as the point C_k^* which is closest to D_i among all C_k . In order to perform these computations quickly, once D_i is projected into each of the planes, all computations are performed in 2-D rather than 3-D. Thus, during initialization each triangle must be stored in both its 2-D and 3-D representations.

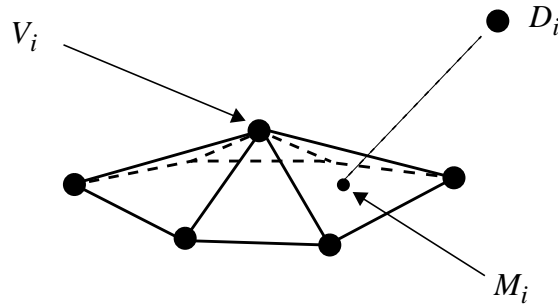


Figure 2: Closest Surface Point Computation

2.2.4 Acceleration

A final speed improvement was realized using a modified version of the *accelerated* ICP algorithm described in [2]. The accelerated ICP algorithm adds the following step to the basic algorithm (after step 2):

- 2b. If the incremental transformations (\mathbf{R} , \mathbf{T}) at iterations $k-1$, $k-2$, and $k-3$ are *well aligned*, extrapolate the current incremental transformation.

The well aligned condition above tests that the solution has been moving in an approximately constant direction. Extrapolation is performed by scaling the current incremental transformation. The scale factor is a function of the least square error and the magnitude of the incremental transformations at the previous three iterations.

Besl and McKay calculate a single acceleration scale factor for both translation and rotation. We achieved better results by decoupling the acceleration of translation and rotation. There are two reasons for doing this. First, in Besl's approach, the well aligned condition above is tested once for both rotation and translation. Thus, for example, if rotation was well aligned but translation was not, no acceleration would be performed. However, an acceleration on rotation alone seems desirable in this situation. A second reason for decoupling is related to the scale factor used in extrapolation. Besl and McKay used the same scale factor to extrapolate both rotation and translation components. This scale factor is designed to extrapolate the solution as much as possible in a single step without overshoot. In the coupled version, the size of the scale factor is governed by the component (translation or rotation) which would cause the solution to over-

1. This is not strictly true, as there are pathological cases for which M_i will lie in a totally different triangle. In our experience, we found that we can ignore such cases.

shoot first. The other component could usually be accelerated further. By decoupling, translation and rotation are independently accelerated as much as possible without overshoot.

2.2.5 Enhancement Results

Four speed enhancements have been described: closest point computation via kd-trees, closest point caching, fast surface point computation, and decoupled acceleration. The results of applying each of these enhancements to a single registration problem are summarized in Table 1. In this problem, D was a point set containing 2432 points and M was a triangular mesh containing 4860 facets. The initial pose error was a rotation of roughly 10 degrees about each axis, and a translation of roughly 10% of the object size along each axis. The ICP termination threshold, ϵ , was small.²

Type	Time	%Time	Iterations	Rot-Acc	Trans-Acc
none	908.8	100.0	122	0	0
a	261.2	28.7	35	11	11
kd	62.2	6.8	122	0	0
kd/a	18.0	2.0	35	11	11
kd/a/d	13.1	1.4	25	13	7
kd/a/d/c	11.9	1.3	25	13	7
kd/a/d/c/2d	8.3	0.9	25	13	7

Table 1: Enhancement Comparisons

In the table, *Type* indicates the enhancements used:

- none: no speed enhancements
- a: coupled acceleration
- kd: kd-tree based closest point computation
- d: decoupled acceleration
- c: closest point caching
- 2d fast surface point computation

Time is the total ICP execution time in seconds. *%Time* is the percentage of time relative to the slowest time. *Iterations* is the number of ICP iterations. *Rot-Acc* and *Trans-Acc* are the number of iterations for which rotation and translation have been accelerated respectively.

The speed improvements shown in Table 1 give an idea of the relative utility of each of the described enhancements. The actual relative utility is a function of the underlying data, the initial pose, and the termination threshold. Acceleration and kd-tree search are always the two most important enhancements. The relative utility of kd-tree search increases with the number of points in the data set. Caching is useful when the termination threshold is small, since the number of cache hits will be large during fine-tuning.

2. The magnitude of ϵ determines the amount of “fine-tuning” performed by the ICP algorithm. Smaller values of ϵ result in pose estimates closer to the local minima.

We believe that additional speed improvements are possible via a multi-processor implementation of the ICP algorithm. The closest point computation is easily parallelized, and doing so should result in speed improvements roughly proportional to the number of processors. We plan to address this issue in future work.

2.3 The Tracking Algorithm

An outline of the tracking algorithm is shown in Figure 3. Each box in the diagram represents a processing step, and the processing sequence is indicated by the large-headed arrows. Inputs to a processing step are indicated by the quantities to the left of each box, while outputs are indicated by the quantities to the right.

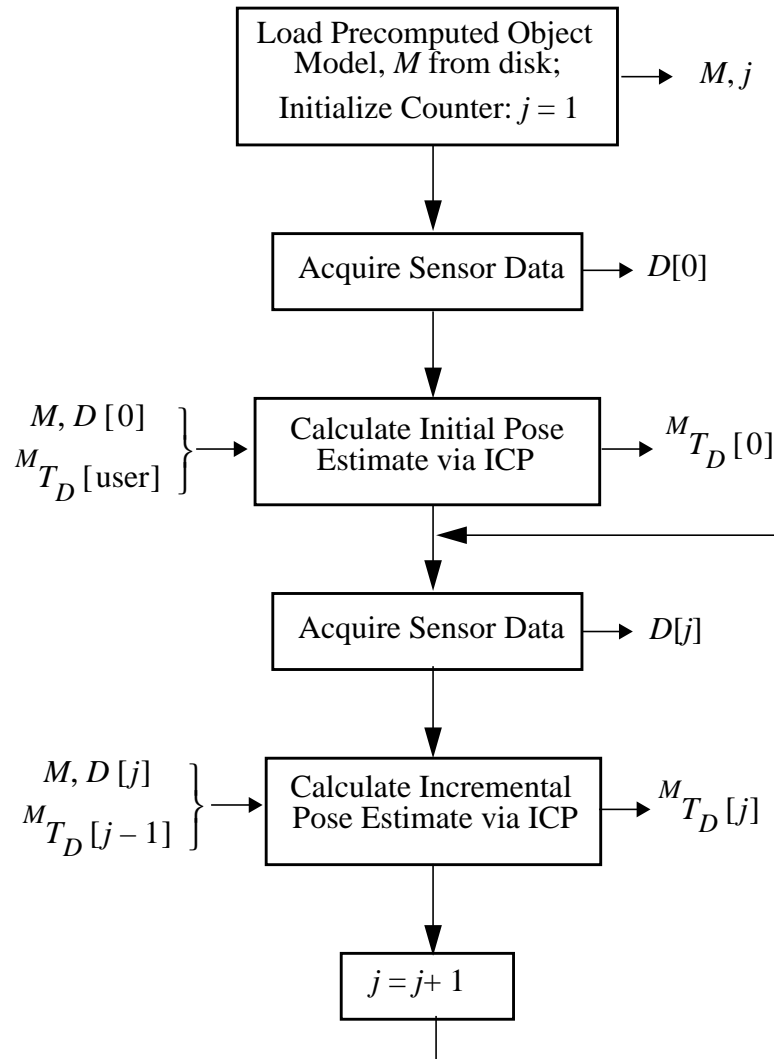


Figure 3: Tracking Algorithm

During initialization, a precomputed triangular mesh model, M , is loaded into memory, and a kd-tree is built from M . After a range image, $D[0]$, is acquired from the sensor, an initial transformation between the Model and the initial object pose can be calculated. This transformation, $M_{T_D}[0]$, can be found in several seconds using the ICP algorithm with a starting transformation provided by the user³. In practice using the face data discussed below, we have found that initial pose errors as large as 15 degrees of rotation about

each axis, and 50% of the surface size in translation will typically converge to the global minimum. Once ${}^M T_D [0]$ has been calculated, it is used to transform the Model to the initial object position. Thus, all future pose estimates are measured with respect to this initial starting pose.

After initialization, the algorithm enters the tracking loop. Within the loop, data are acquired by the sensor, and the object pose is estimated via the ICP algorithm in roughly 0.1 - 0.3 sec. These high speeds are possible for two reasons. First, the difference in object position between iterations j and $j-1$ is typically small. For example, translational velocities of 10cm per second and rotational velocities of 20 degrees per second lead to incremental object pose discrepancies of roughly 2cm and 4 degrees. Thus, since the ICP algorithm uses ${}^M T_D [j-1]$ as the starting point when finding ${}^M T_D [j]$, registration can be performed in a small number of iterations, typically 3-10. Second, the resolution of the range data used in our experiments was usually 16x16. Thus, the number of closest point computations required (256) was significantly less than that required when using the full sensor resolution (32x32).

During each data acquisition cycle, two preprocessing steps are performed on the range data. First, it is necessary to eliminate noisy data. For the CMU VLSI range sensor, noisy data is associated with poor reflection from the object of the projected light. Thus, noisy range data can be eliminated by thresholding reflected intensity values. Second, it is necessary to determine which range data points lie on the surface of the object to be tracked. Since our experiments were performed in an uncluttered environment, range data on the object surface can be distinguished by thresholding the Z component of the range data. While this simple operation works well for our experiments, a more sophisticated approach would be required in a cluttered environment.

Using ${}^M T_D [j-1]$ as the starting point for incremental pose estimation works well when object motion is erratic and unpredictable. In some situations, however, object motion may be smooth, continuous and thus easier to predict. For such motions, improved results are possible using an extrapolation scheme such as a Kalman filter. While we have not implemented a Kalman filter for this purpose, we have implemented both first and second order extrapolation. Since the extrapolated pose is often closer to the true pose than ${}^M T_D [j-1]$, the cycle time reduced.

2.4 Tracking Results

The primary hardware component of the pose tracker is the CMU high-speed VLSI range sensor [7]. The sensor consists of two main components: the sensor head and light-stripe generator as seen in Figure 4. The current version of this sensor can acquire 32x32 cell range images in as little as 1 millisecond. Range data is acquired at 10 bits of resolution and is accurate to 0.1% along the direction of range measurement. As currently configured, the sensor workspace is roughly a cube 15cm on a side.

Using a precision positioning device to provide ground truth, we characterized the *static* accuracy of the pose tracker. All accuracy tests were performed using the same physical object: a small bust of a human face. In these tests, the average translation error was found to be 0.9mm (about 1% of the object size), while the average rotation error was 1.4 degrees. Until recently, it had not been possible to measure the system's dynamic accuracy since our positioning device was not capable of generating accurately calibrated dynamic trajectories. The recent acquisition of a high-speed, fiducial-based pose tracker [25] will allow us to characterize dynamic accuracy. We have, however, been able to characterize the dynamic *repeatability* of the tracker. Object trajectories with maximum velocities of 100mm/sec and 22 degrees/sec were reliably tracked with repeatability of roughly 1 degree in rotation, and 1mm in translation. Additional system performance results can be found in [28].

3. A fully automated initialization which does not require user input would be possible by applying one of the techniques for solving the global pose estimation problem discussed in [2].

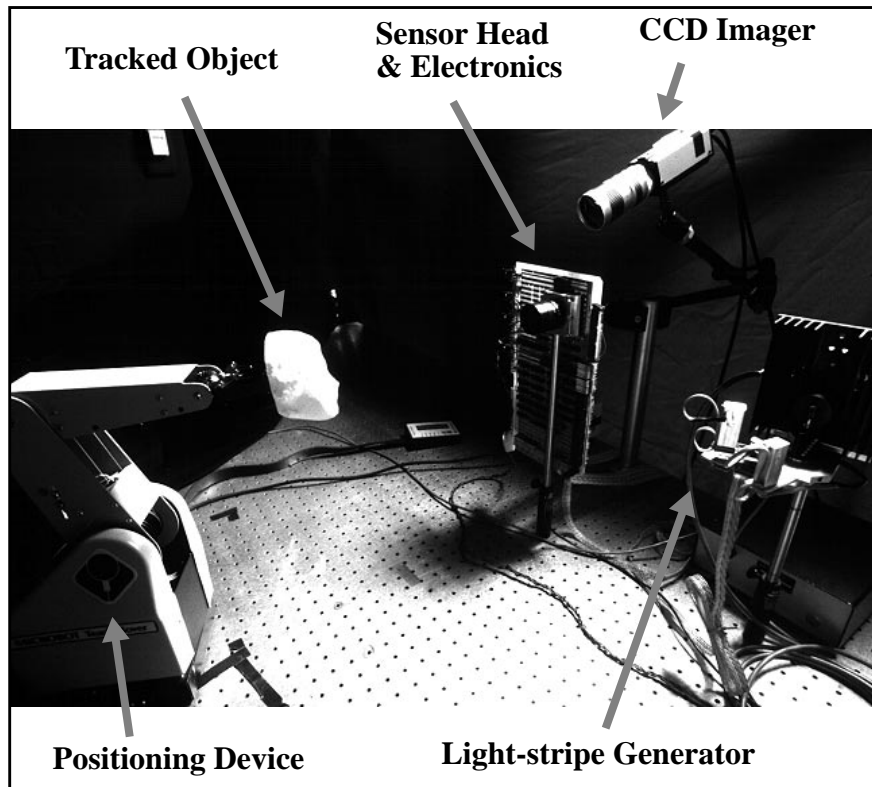


Figure 4: Pose Tracking Apparatus

We have demonstrated the high-speed tracker by estimating the pose of real human faces. The polygonal mesh surface model of one of these faces is seen in Figure 5. The data in this figure was acquired using a conventional light-stripe range finder [26]. While the face tracking demonstrations were not performed under clinical conditions, we plan to evaluate our tracking techniques on a head-tracking problem in neurosurgery.

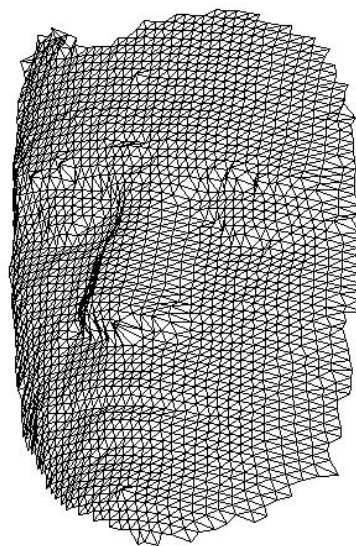


Figure 5: Surface Mesh of the First Author's Face

The high-speed tracking method which we have demonstrated is independent of the particular sensor used to acquire intra-surgical data. The sensor must be able to acquire 3-D data on the surface of the relevant anatomy at rates and densities sufficient for the application. The tracking method is also independent of the particular anatomical region which is being tracked. The anatomy to be tracked must be rigid, and its surface must be visible to the selected sensor during surgery.

This concludes the first part of the paper on high-speed pose tracking. The second part of the paper discusses the problem of the selective acquisition of intra-surgical pose refinement data.

2.5 Pose Refinement Data Selection

In general, there is a strong relation between the accuracy resulting from surface-based pose refinement, and the quantity of intra-surgical data used. Large quantities of high-quality intra-surgical data tend to result in better accuracy. Unfortunately, there are often high costs associated with the acquisition of large quantities of intra-surgical data. Two such costs are the *time* needed to acquire the data, and the patient's *exposure to radiation*. Minimizing acquisition time is particularly important due to the high monetary costs of operating room use, and the risk of patient infection which increases with time. The fundamental trade-off between data quantity and pose refinement accuracy motivates our work in the selection of intra-surgical data. The goal of this work is to generate and execute a plan for intra-surgical data acquisition such that the amount of data acquired is minimized, while ensuring that pose refinement accuracy requirements are met.

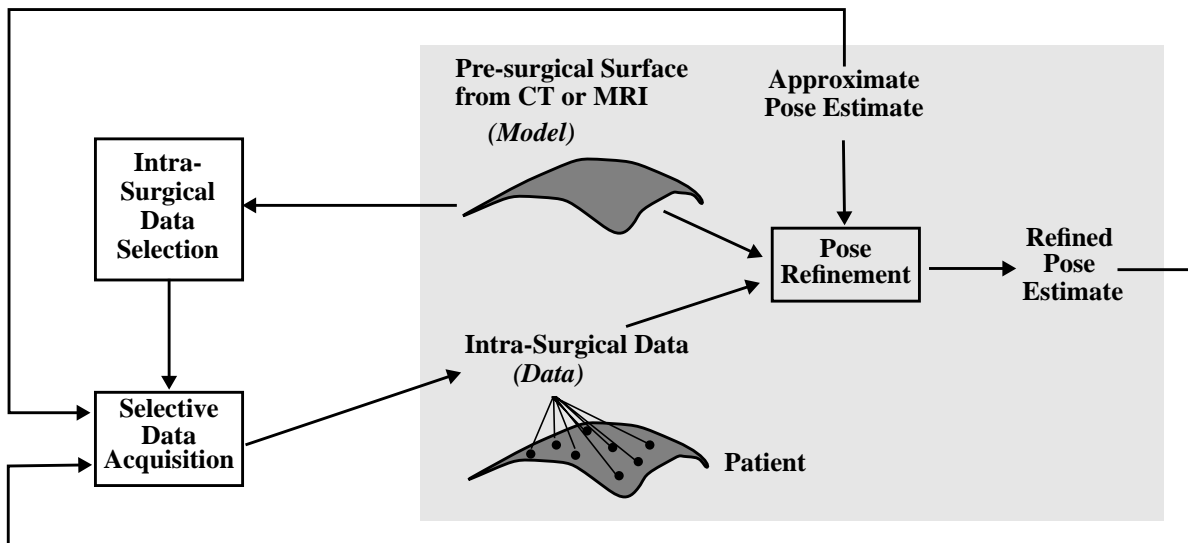


Figure 6: Data Selection and Acquisition

A method for selecting and acquiring intra-surgical data is outlined in Figure 6. The shaded portion of the figure is identical to Figure 1. The first step in the approach is *Intra-Surgical Data Selection*. This step uses the pre-surgical surface model as input, and outputs a set of desired intra-surgical data, specified in the reference frame of the pre-surgical data. The criterion for data selection is the maximization of *geometric constraint* between pre-surgical and intra-surgical data.

The next step is *Selective Data Acquisition*. This step requires as input the desired intra-surgical data, and a current estimate of the object's pose. The action performed by this step is the acquisition of the desired data using a sensor of choice. Due to uncertainty in the pose of the patient, it is impossible to collect the desired data precisely. (If we could collect this data precisely, then we would already know the pose which

we are seeking!) Therefore, it is necessary to have an estimate of the patient’s pose to aid the acquisition process. Initially, an approximate pose estimate is available to guide the acquisition process via coarse anatomical landmark correspondence. After some intra-surgical data have been acquired, an incremental pose refinement can be calculated and fed back to the data acquisition module to aid in the collection of subsequent data. The order of data acquisition can be planned such that the earliest data collected are the least sensitive to precise localization of the patient, while data collected during the later stages require a fairly good estimate of patient pose.

The actual acquisition process could be performed either by a device such as a robot, or manually by a surgeon. When a robot is used, the pose estimate can be fed back directly to the robot’s controller. When a surgeon acquires the data manually, pose feedback can be in the form of a 3-D graphical display of the relevant anatomy, with the desired data overlaid on upon this display. While we have not yet used this technique clinically, we plan to do so shortly in the area of orthopaedic surgery.

2.5.1 Geometric Constraint: Motivating Examples

To gain a better appreciation for the role of geometric constraint in pose refinement, consider the situation in Figure 7. Assume that the goal is to estimate the pose of a slotted cylinder using a coordinate measuring device to collect data on its surface. If data were only collected at points indicated by an “X”, it would be impossible to determine orientation about the central axis, or translation along it. Additional data collected in the disk-shaped end regions and within the slot would allow determination of translation and rotation respectively. The problem with the data configuration shown in Figure 7 is that there are “freedoms” in the geometric constraint between the surface of the cylinder, and the discrete points shown. Such freedoms result in multiple solutions in R and T to (1).

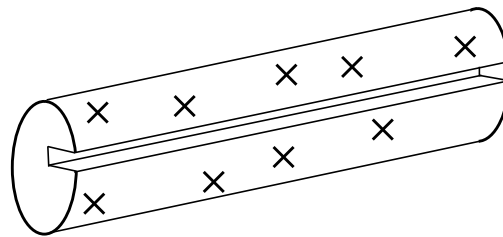


Figure 7: Localization of a Slotted Cylinder

As a second example of the role of geometric constraint in pose refinement, imagine that we are trying to localize a cube (disregarding the symmetries) using data sampled from each of its faces. Figure 8 shows three sampling configurations on a cube: C1 has 25 points per face for a total of 150 points, while C2 and C3 have 4 points per face for a total of 24 points. If we were to perform pose refinement using each of these sampling configurations assuming noisy data, which one would we expect to result in the best accuracy?

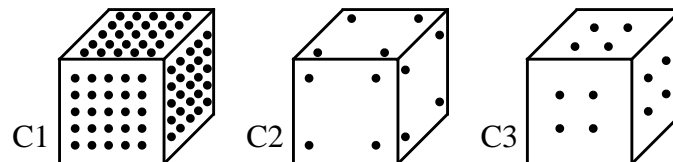


Figure 8: Localization of 3 Cubes

In order to answer this question, we performed a simple experiment. For each sampling configuration, we performed pose refinement 100 times from random initial poses. Zero mean gaussian noise was added to

each discrete point in the data set. Figure 9 shows the resulting pose refinement errors from (1), normalized by the number of points, and plotted relative to the error for configuration C1. As might be expected, the configuration C1 results in the best pose refinement accuracy. This agrees with the intuition that larger quantities of data will result in better pose refinement accuracies. The results for each of the 24 point configurations is less intuitive, and an explanation will be delayed until the next section. Note that despite the same number of data points, configuration C2 clearly provides better accuracy than configuration C3.

Normalized Pose Error

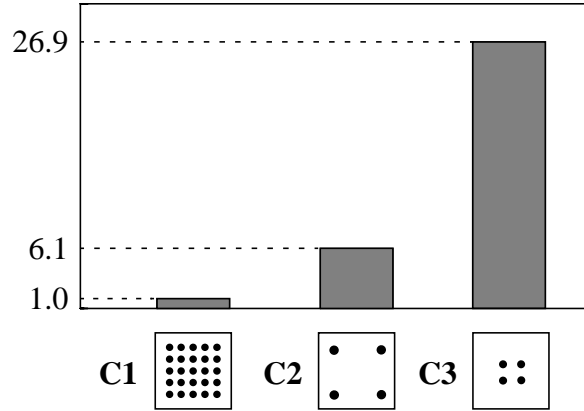


Figure 9: Cube Pose Refinement Errors

2.5.2 Geometric Constraint Analysis

We begin the description of geometric constraint analysis by posing the following question: given a discrete point lying on a surface, how does the distance between the point and the surface vary as the point is perturbed by a small amount about its resting position?

The distance between a point \mathbf{x} , and a surface is defined as the length of the shortest line between the point and the surface. In general, there is no closed form analytical expression for this distance given an arbitrary surface; however, the following local approximation has been proposed [29]:

$$D(\mathbf{x}) = \frac{F(\mathbf{x})}{\|\nabla F(\mathbf{x})\|} \quad (2)$$

where $F(\mathbf{x}) = 0$ is the implicit equation of the surface, $\|\nabla F(\mathbf{x})\|$ is the magnitude of the gradient to the surface, \mathbf{x} is a point which may or may not lie on the surface, and $D(\mathbf{x})$ is the approximate distance. It can be shown that $D(\mathbf{x})$ is a first order approximation to the true point-to-surface distance.

Assume that there exists a point \mathbf{x}_s , which lies on the surface. This point can be perturbed with respect to the surface by applying a differential transformation T to the point. T can be represented by a homogeneous transformation, which is a function of 6 parameters $(t_x, t_y, t_z, \omega_x, \omega_y, \omega_z)$, where $(\omega_x, \omega_y, \omega_z)$ are rotations about the X, Y, and Z axes respectively, and (t_x, t_y, t_z) are translations along the newly rotated X, Y, and Z axes. Define:

$$\mathbf{t} = [t_x \ t_y \ t_z \ \omega_x \ \omega_y \ \omega_z]^T \quad (3)$$

as the 6-vector of parameters. As a first step in the analysis, consider the gradient of D with respect to each of the parameters of t . It can be shown that the resulting 6-vector $V(x_s)$ is defined as:

$$V(x_s) = \frac{\partial}{\partial t} D(T(x_s)) = \begin{bmatrix} \mathbf{n} \\ \mathbf{x}_s \times \mathbf{n} \end{bmatrix} \quad (4)$$

where \mathbf{n} is the unit normal to the surface evaluated at the point \mathbf{x}_s . $V(x_s)$ relates a small transformation specified by a vector dt to a corresponding change in distance between the point and the surface. In other words:

$$D(T(x_s)) = V^T(x_s) dt \quad (5)$$

Until this point, we have considered how the distance between a single point and a surface changes as a function of an arbitrary, small rigid transformation. The goal is to analyze the constraint imposed by a *collection* of points upon the underlying surface. Squaring (5), results in:

$$\begin{aligned} D^2(T(x_s)) &= dt^T V(x_s) V^T(x_s) dt \\ &= dt^T M(x_s) dt \end{aligned} \quad (6)$$

where $M(x_s)$ is a symmetric, positive semi-definite 6x6 matrix. Summing the quantity in (6) over points within a region of the surface R , results in the sum of squared distance errors:

$$\begin{aligned} E_R(T(x_s)) &= dt^T \left[\sum_{\mathbf{x}_s \in R} M(x_s) \right] dt \\ &= dt^T \Psi_R dt \end{aligned} \quad (7)$$

where Ψ_R is the sum of the $M(x_s)$ matrices evaluated at each point in the region R . The matrix Ψ_R is a scatter matrix which contains information about the distribution of the original $V(x_s)$ vectors over the region R . Performing principal component analysis [13], Ψ_R is transformed into an expression which is more easily interpreted.

$$\begin{aligned} E_R(T(x_s)) &= dt^T Q \Lambda Q^T dt \\ &= \sum_{i=1}^6 \lambda_i \left(dt^T \mathbf{q}_i \right)^2 \end{aligned} \quad (8)$$

where E_R is the least square error over the region R , Λ is a diagonal 6x6 matrix containing the eigenvalues of Ψ_R , Q is a 6x6 matrix whose columns are the eigenvectors of Ψ_R , $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \lambda_4 \geq \lambda_5 \geq \lambda_6$ are the eigenvalues of Ψ_R , and \mathbf{q}_i are the corresponding eigenvectors. Each eigenvector \mathbf{q}_i can be interpreted as a differential transformation represented as a 6-vector; the first three elements are the translation components, and the last three elements are the rotation components. We should note that this result is similar to one presented in [19].

From (8) it can be seen that the eigenvector \mathbf{q}_1 corresponding to the largest eigenvalue, represents the *direction of maximum constraint*. Perturbing the points in the region R in the direction of \mathbf{q}_1 will result in the largest possible change in E_R from among all possible directions of perturbation. Similarly, the differential transformation represented by the eigenvector \mathbf{q}_6 corresponds to the *direction of maximum freedom*. Perturbing the points in this direction will result in the smallest possible change in E_R from among all possible directions of perturbation. In general, an eigenvalue λ_i is proportional to the rate of change of E_R induced by a differential transformation in the direction \mathbf{q}_i .

A special situation occurs when some of the λ_i are close to or equal to zero. For each such eigenvalue, a singularity exists such that perturbing the points in the direction of the corresponding eigenvector will re-

sult in no change in E_R . Clearly, such singularities are very bad when performing pose refinement since it becomes impossible to localize the pose in the direction(s) of the singularity. The slotted cylinder example of Figure 7 has two zero eigenvalues, one corresponding to a rotation about the central axis and the other corresponding to a translation along it.

We now have a basis for understanding the pose refinement accuracy results from the cubes of Figure 9. For each of the cube sampling configurations (C1, C2 and C3), we performed the constraint analysis presented above. Figure 10 plots the magnitude of the smallest eigenvalue for each configuration versus the pose refinement errors plotted in Figure 9. The trend from this plot is clear; large magnitudes of the minimum eigenvalue result in better pose refinement accuracies. This agrees with the intuition from (8); for data configurations with small minimum eigenvalues there are perturbations about the global minimum which result in only small changes in error. For the cubes of Figure 10, a small rotation has a much larger effect on the error in (1) for the points in configuration C2 than for those in configuration C3. This allows the cube to be localized more accurately using the points in configuration C2.

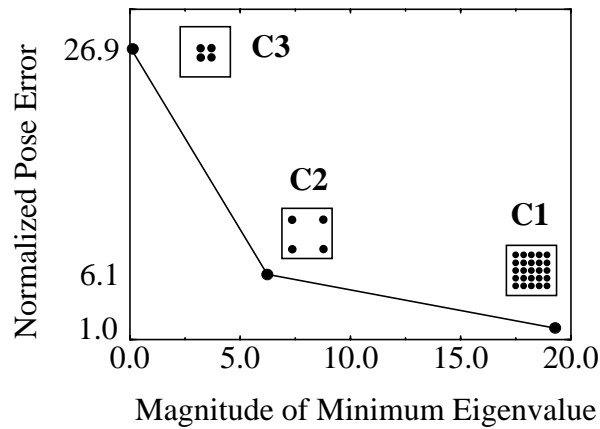


Figure 10: Accuracy vs. Minimum

In Figure 10, the magnitude of the minimum eigenvalue can be thought of as a criterion measure which evaluates the “goodness” of a particular sampling configuration. In general, there are a variety of possible criterion measures which could be used for this purpose. We are currently investigating several such measures, including a measure of isotropy proposed in [14]:

$$6 \cdot \frac{\sqrt[6]{\lambda_1 \cdot \lambda_2 \cdot \lambda_3 \cdot \lambda_4 \cdot \lambda_5 \cdot \lambda_6}}{\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 + \lambda_6} \quad (9)$$

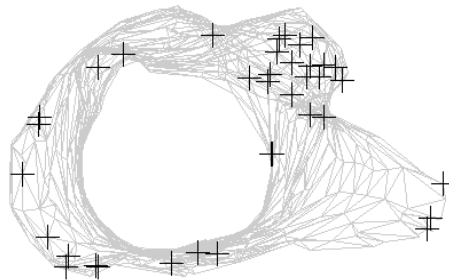
and the following measure suggested by Nahvi [20] discussion of the implications of criterion measure selection for a related problem can be found in [14].

2.5.3 Geometric Constraint Synthesis Results

The constraint analysis method of the previous section provides a criterion for performing intra-surgical data selection as outlined in Figure 6. The goal of geometric constraint synthesis is to generate data configurations for a given surface which maximize the smallest eigenvalue (or other criterion) resulting from constraint analysis. We have developed a technique for performing constraint synthesis for fixed-size, discrete point data sets. While the synthesized data configurations are not *provably* optimal, we have verified empirically that the resulting pose refinement accuracy is similar to the best data sets generated by local human experts. Since this work is research in progress, the results presented in this section should be viewed as preliminary.

To demonstrate the capabilities of constraint synthesis, we ran the algorithm on a surface model generated from CT images of the proximal end of a human femur. Figure 11 shows a synthesized data set containing 37 points superposed on the surface model of the femur. Generation time for this set was about 30 minutes on a Sparc-10 workstation. In order to evaluate the synthesized data set, we compared it to 21 manually selected data sets of the same size. These sets were selected by 7 people, including one surgeon with training as an engineer, three experts in 3-D computer vision, and three graduate students in robotics. We performed 1000 iterations of pose refinement using each of the synthesized data sets. Starting poses were determined randomly, with maximum initial pose errors of 30mm in translation, and 15 degrees in rotation. On each iteration, zero-mean, gaussian noise with a standard deviation of 1.0mm was added to each point in the data sets. Figure 12 shows the results of this experiment. The graph shows the magnitude of the minimum eigenvalue from constraint analysis for each of the 22 data sets, versus the pose refinement error (1), averaged over 1000 iterations. Each point in the graph represents a different data set.

Bottom View:



Side View:

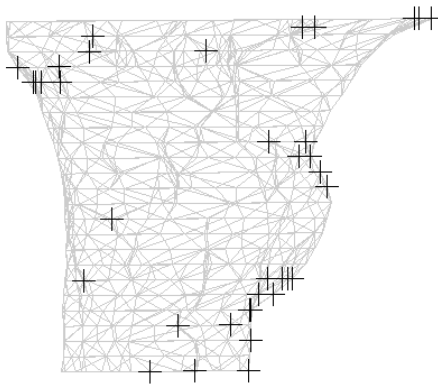


Figure 11: Synthesized Data Set for Femur

Several observations should be made from Figure 12. First, the magnitude of the minimum eigenvalue is related to the variance in pose refinement error. Data sets with small minimum eigenvalues may have large pose refinement errors, while those with larger minimum eigenvalues all have small errors. This suggests a relation between the magnitude of the minimum eigenvalue for a given data set, and an upper bound on the resulting pose refinement error. Second, the data set with the largest minimum eigenvalue and smallest pose refinement error was the one synthesized automatically by our algorithm. The average transformation errors for the synthesized data set are 0.4mm in translation and 0.2 degrees in rotation. Third, note that calculating the magnitude of the minimum eigenvalue is a very fast operation (roughly 1ms on a Sparc-10), while calculating the average pose refinement error is much slower (about 5 minutes on a Sparc-10). It is the low cost of calculating the minimum eigenvalue which allows us to synthesize near-optimal data sets.

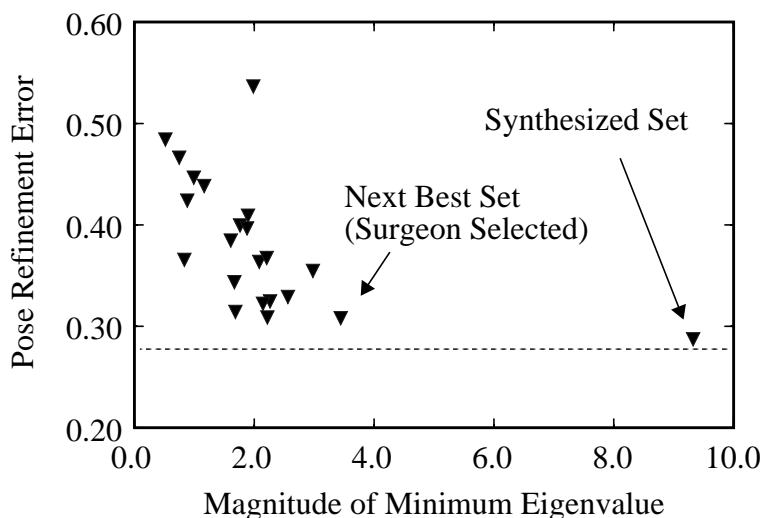


Figure 12: Accuracy vs. Minimum Eigenvalue:

The results presented in Figures 11 & 12 implicitly assume that acquisition of the indicated intra-surgical data is both clinically and technically feasible. In general, certain regions of the underlying structure (e.g. femur) may be inaccessible to intra-surgical data acquisition. Furthermore, diseased or damaged regions of the anatomy may result in inaccurate pre- and intra-surgical data. For example, it is very difficult to build accurate surface models of an arthritic femoral head from pre-surgical CT data. Such model inaccuracies will ultimately result in registration errors. To deal with this problem, clinical application of constraint synthesis will require the demarcation of accessible and un-diseased anatomical regions. The synthesis process can then be constrained to generate data which lie within these regions.

We have repeated the experiment described above on a few other surfaces with similar results. In particular, when synthesizing data points for the cube using 24 discrete points, the technique reliably finds the provably optimal configuration: a distribution similar to configuration C2, with points located as close to the corners as possible. We are currently in the process of a more thorough investigation of the synthesized data sets. In particular, we are performing a series of experiments designed to validate our results using physical (non-simulated) registration experiments. Using fiducial-based registration to provide ground truth, we are investigating the accuracy which results from surface-based registration with and without the use of the synthesized “optimal” data.

The constraint synthesis algorithm finds near-optimal configurations for *fixed* amounts of data; the number of data points is an input to the algorithm. The overall goal of data selection is not only to determine good configurations, but also to minimize the amount of data required. Towards this goal we are currently investigating methods for selecting minimally sized data sets. As input, our proposed method uses bounds on the acceptable accuracy, and estimates of the uncertainty in: the intra-surgical sensor data, the pre-surgical surface model, and the initial pose. The success of this method will depend on our ability to develop realistic noise models for the sensor and pre-surgical surface.

Accurate surface-based registration relies upon the ability to build precise surface models from pre-surgical data. While techniques for generating surface models from CT data have been available for several years, the resulting models are typically used for visualization applications in which model accuracy is not crucial. We are currently investigating accuracy issues in surface model generation by analytically comparing generated surface models to a known ground truth [23]. In addition, we are studying the sensitivity of surface-based pose refinement to noise in the sensor and pre-surgical surface models.

We believe that the methods presented in this part of the paper will become useful tools in the area of intra-surgical pose refinement. Geometric constraint analysis will be useful not only to provide a criterion for data selection, but also to allow the study and evaluation of manually created data sets. Geometric constraint synthesis will be useful for automatically generating near-optimal data sets without input from a human expert. In addition to applying constraint synthesis on a per-patient basis, it could also be useful to study data generation for entire classes of objects (i.e. all femurs).

2.6 Conclusions

In this paper, we have presented two novel techniques with application to medical robotics and computer assisted surgery. First, we have demonstrated a high-speed pose tracking capability with application to intra-surgical use. The technique is independent of the particular sensor used, and of the anatomical region to which it is applied. While this technique has not yet been demonstrated in a clinical application, we feel that it has significant promise for clinical use. Second, we have presented a method for selecting near-optimal, intra-surgical pose refinement data. The goal of the selection process is to minimize the amount of data acquired for pose refinement, while maintaining good accuracy.

There are several directions in which the work is proceeding. With the high-speed tracker, we are currently working on a multi-processor implementation which would parallelize the closest point computation. The goal is to increase both the rate at which pose tracking can be performed, as well as the amount of data which can be processed at high speeds. We are also investigating an extension to the technique which would allow tracking of articulated objects, such as human hands. Finally, we are planning to evaluate the high-speed pose tracker on head-tracking for neurosurgery.

In intra-surgical data selection, there are several directions of ongoing work. First, intra-surgical data selection currently requires manual specification of data size. We are extending this method to generate minimally-sized data sets automatically. Second, intra-surgical data selection currently generates discrete point 3-D data sets. We are extending the method to generate bounding contours such as those that could be derived from radiographs or CT images. Third, in intra-surgical registration, it is not always necessary for pose accuracy to be isotropic; accuracy in certain directions may be more critical than in others. Given such non-isotropic accuracy requirements, we would like to generate data sets which make the best possible use of limited data. Finally, we are planning to apply the data selection method to a clinical problem in the area of orthopaedic surgery.

Acknowledgments

The authors would like to thank Robert O'Toole for his insightful comments on this paper. Andy Gruss provided help using the CMU VLSI range sensor. Dr. Anthony DiGioia applied his clinical expertise to the selection of discrete point data to which we compared our synthesis results.

3 References

- [1] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the Association for Computing Machinery*, 18(9):509–517, September 1975.
- [2] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, February 1992.
- [3] O. D. Faugeras and M. Hebert. The representation, recognition, and locating of 3-d objects. *The International Journal of Robotics Research*, 5(3):27–52, Fall 1986.
- [4] J. H. Friedman, J. L. Bentley, and R. A. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software*, 3(3):209–226, 1977.

- [5] R. L. Galloway and R. J. Maciunas. Stereotactic Neurosurgery. *Critical Reviews in Biomedical Engineering*, 18(3):181–205, 1990.
- [6] W. E. L. Grimson, T. Lozano-Perez, W. M. Wells, G. J. Ettinger, S. J. White, and R. Kikinis. Automated registration for enhanced reality visualization in surgery. In *Proceedings of the First International Symposium on Medical Robotics and Computer Assisted Surgery*, pages 82–89, Pittsburgh, PA, September 1994.
- [7] A. Gruss, S. Tada, and T. Kanade. A VLSI smart sensor for fast range imaging. In *International Conference on Intelligent Robots and Systems (IROS '92)*, pages 349–58, Raleigh, NC, July 1992. IEEE.
- [8] A. Gueziec and N. Ayache. Smoothing and matching of 3-d space curves. In *Proceedings of the Conference on Visualization in Biomedical Computing 1992*, pages 259–273, Chapel Hill, NC, October 1992. SPIE.
- [9] R. M. Haralick and H. Joo. 2d-3d pose estimation. In *Proceedings of the 9th International Conference on Pattern Recognition*, pages 385–391, Rome, Italy, November 1988. IAPR, IEEE.
- [10] D. L. Hauser and D. L. Taylor. Optimal alignment of geometric models for comparison. In *Proceedings of the First Conference on Visualization in Biomedical Computing*, pages 426–431, Atlanta, May 1990. IEEE.
- [11] B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642, April 1987.
- [12] H. Jiang, R. A. Robb, and K. S. Holton. A new approach to 3-d registration of multimodality medical images by surface matching. In *Visualization in Biomedical Computing - SPIE Vol 1808*, pages 196–213. SPIE, 1992.
- [13] M. G. Kendall and A. Stuart. *Canonical Variables*, chapter 43, pages 320–369. C Griffin, London, 4th edition, 1977.
- [14] J. Kim and P. K. Khosla. Dexterity measures for design and control of manipulators. In *Proceedings of the International Workshop on Intelligent Robots and Systems*, pages 758–763, Osaka, Japan, November 1991. IEEE/RSJ.
- [15] Y. S. Kwoh, J. Hou, E. A. Jonckheere, and S. Hayati. A robot with improved absolute positioning accuracy for ct guided stereotactic brain surgery. *IEEE Transactions on Biomedical Engineering*, 35(2):153–160, February 1988.
- [16] S. Lavallee, P. Sautot, J. Troccas, P. Cinquin, and P. Merloz. Computer assisted spine surgery: a technique for accurate transpedicular screw fixation using ct data and a 3-d optical localizer. In *Proceedings of the First International Symposium on Medical Robotics and Computer Assisted Surgery*, pages 315–322, Pittsburgh, PA, September 1994.
- [17] S. Lavallee, R. Szeliski, and L. Brunie. Matching 3-d smooth surfaces with their 2-d projections using 3-d distance maps. In *SPIE Proceedings: Geometric Methods in Computer Vision*, pages 322–336, San Diego, July 1991.
- [18] T. Masuda and N. Yokoya. A robust method for registration and segmentation of multiple range images. In *Proceedings of the 2nd CAD-Based Vision Workshop*, pages 106–113, February 1994.
- [19] C. Menq, H. Yau, and G. Lai. Automated precision measurement of surface profile in cad-directed inspection. *IEEE Transactions on Robotics and Automation*, 8(2):268–278, April 1992.
- [20] A. Navhi. *Identification and Control of the Redundant Linear Drives of Two Anthropomorphic Robots*. Masters thesis, McGill University, Montreal, September 1994.
- [21] S. K. Nayar, M. Watanabe, and M. Noguchi. Real-time focus range sensor. Technical Report CUCS-028-94, Department of Computer Science, Columbia University, New York, N.Y., November 1994.

- [22] L. P. Nolte, L. J. Zamorano, J. Zhaowei, Q. Wang, F. Langlotz, E. Arm, and H. Visarius. A novel approach to computer assisted spine surgery. In *Proceedings of the First International Symposium on Medical Robotics and Computer Assisted Surgery*, pages 323–328, Pittsburgh, PA, September 1994.
- [23] R. V. O’Toole, D. A. Simon, B. Jaramaz, O. Ghattas, M. K. Blackwell, L. Kallivokas, F. Morgan, C. Visnic, A. M. DiGioia, and T. Kanade. Towards more capable and less invasive robotic surgery in orthopaedics. In *Proceedings of the First International Conference on Computer Vision, Virtual Reality and Robotics in Medicine*, pages 123–130, Nice, France, April 1995. Springer-Verlag.
- [24] K. Radermacher, H. W. Staudte, and G. Rau. Computer assisted orthopedic surgery by means of individual templates - aspects and analysis of potential applications. In *Proceedings of the First International Symposium on Medical Robotics and Computer Assisted Surgery*, pages 42–48, Pittsburgh, PA, September 1994.
- [25] R. Rohling, P. Munger, J. Hollerbach, and T. Peters. Comparison of relative accuracy between a mechanical and an optical position tracker for image-guided neurosurgery. In *Proceedings of the First International Symposium on Medical Robotics and Computer Assisted Surgery*, pages 277–282, Pittsburgh, PA, September 1994.
- [26] K. Sato and S. Inokuchi. Range-imaging system utilizing nematic liquid crystal mask. In *Proc. ICCV*, pages 657–661, London, UK, 1987.
- [27] A. Schweikard, R. Tombropoulos, J. R. Adler, and J. Latombe. Planning for image-guided radiosurgery. In *Applications of Computer Vision in Medical Image Processing*, pages 96–101. AAAI, March 1994.
- [28] D. A. Simon, M. Hebert, and T. Kanade. Real-time 3-d pose estimation using a high-speed range sensor. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 2235–2241, San Diego, CA, May 1994. IEEE.
- [29] G. Taubin. Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(11):1115–1138, November 1991.
- [30] R. H. Taylor, B. D. Mittelstadt, H. A. Paul, W. Hanson, P. Kazanzides, J. F. Zuhars, B. Williamson, B. L. Musits, E. Glassman, and W. L. Bargar. An image-directed robotic system for precise orthopaedic surgery. *IEEE Transactions on Robotics and Automation*, 10(3):261–275, June 1994.
- [31] M. Yamamoto. Direct estimation of range flow on deformable shape from a video rate range camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(1):82–89, January 1993.
- [32] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *The International Journal of Computer Vision*, 13(1):119–152, October 1994.