Proceedings of the 2004 IEEE
International Conference on Robotics & Automation
New Orleans, LA • April 2004

# Spatial Motion Constraints in Medical Robot Using Virtual Fixtures Generated by Anatomy

Ming Li and Russell H. Taylor

*Department of Computer Science*
*NSF Engineering Research Center for Computer Integrated Surgical Systems and Technology*
*The Johns Hopkins University*
*Baltimore, Maryland 21218, USA*

{liming & rht}@cs.jhu.edu

***Abstract*** **- In Ear, Nose and Throat (ENT) surgery, the operating volume is very limited. This is especially true in sinus surgery, when the instrument passes through the nasal and sinus cavity to reach the pathological area. The nasal and sinus bones impose geometric constraints on the work volume. During the surgery, the surgeon needs to control the motion of the instrument tip to accomplish some delicate procedure; meanwhile he/she needs to avoid hitting anatomic constraints. In this paper, we present a method to assist the path following task in a constrained area. The system reads the user's force input and combines it with the planned tip-trajectory to create the tip motion constraints; meanwhile it generates the tool-shaft boundary constraints based on a 3-D geometric model. We map instrument tip motion and boundary information to joint displacements via robot kinematics, and then use a constrained quadratic optimization algorithm to compute the optimal set of corresponding joint displacements. In the preliminary study, we show that robot guidance using cooperative control and virtual fixtures derived from complex geometry can assist users in skilled manipulation tasks, while maintaining desirable properties such as collision avoidance and safety.**

***Index Terms - collaborative manipulation; virtual fixture; optimization robot control; geometry constraint.***

## I. INTRODUCTION

In sinus surgery, medical instruments or an endoscope camera are inserted through the nose into a sinus cavity. The surgeon must precisely manipulate these instruments based on visual feedback from the endoscope and on information from preoperative 3D images such as CT. Although surgical navigation systems can track instruments relative to preoperative data [1], the difficulty of precise surgical manipulation with endoscopic instruments makes endoscopic sinus surgery a natural candidate application domain for cooperatively controlled surgical robots.

The goal of human-machine collaborative systems (HMCS) research is to create mechanisms that selectively provide cooperative assistance to a human user (for us, a surgeon), while allowing the user to retain ultimate control of the procedure. Taylor *et al.* [2] developed an augmentation system for fine manipulation, and Kumar *et al.* [3] applied it to microsurgical tasks such as inserting a needle into a 100 micron retinal vessel. Virtual fixtures in a collaborative system provide cooperative control of the manipulator by "stiffening" a hand-held guidance mechanism against certain directions of motion or forbidden regions of the workspace. Recent research on motion constraints using Kumar's system [4, 5] has focused on simple techniques for "guidance virtual fixtures". This prior work focused on 2D geometric guidance motion of the tool tip or camera and assumed that the tool or camera itself did not have any other environmental constraints.

The focus of this paper is automatic generation of spatial motion constraints associated with complex 3D anatomy, based on preoperative medical images. In endoscopic sinus surgery, the endoscope and other instruments have some degree of translational and rotational freedom but their motion is constrained by anatomic structures. During surgery, the instruments or the camera should avoid collisions or excessive force on delicate anatomy while still moving in the desired manner to accomplish the intended task. Constrained robot control has been discussed previously in both tele-manipulation and cooperative manipulation contexts. Funda, Taylor, *et al.* [6] formulated desired motions as sets of task goals in any number of coordinate frames relevant to the task, optionally subject to additional linear constraints in each of the task frames for redundant and deficient robots. The geometric complexity of medical workspace constraints in ENT makes this approach attractive for our current research. In preliminary work [7] we applied the method of [6] to surgical environments in which motion constraints on a simple path-following task are automatically derived from registered pre-operative models created from 3D images. In this paper, we generalize these techniques to more general cooperative control cases in which virtual fixtures are automatically generated from registered preoperative medical images. First, we describe the system and algorithm for generating spatial motion constraints derived from geometry. We then describe implementation and experiments.

Fig. 1 conceptually illustrates the relationship between the instrument, 3D path and approach aperture to the workspace cavity in our experiments. The surgical instrument is a sharp-tipped pointer held either by a robot or freehand. In other cases it might be a surgical endoscope or a grasping instrument. We use the term "tip frame" to refer to a coordinate system whose origin is at the tip of the pointer and whose orientation is parallel to the tool holder of the robot. The "tool boundary frame" is a coordinate system whose origin corresponds to the point on the tool that is closest to the surrounding anatomy and whose orientation is again parallel to the tool holder.

0-7803-8232-3/04/$17.00 ©2004 IEEE

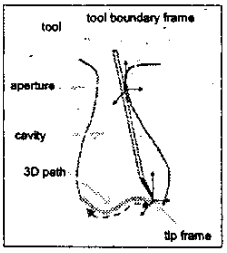


Fig. 1 Relationship between the tool, cavity, approach aperture, and 3D path in our sample task

## II. VIRTUAL FIXTURE GENERATION SYSTEM

Fig. 2 shows the system of virtual fixture generation for our task. In the pre-operative stage, from medical images (e.g., CT image, MRI), the surgeon defines the tool-tip path by selecting positions in the image slice. 3D geometry model also is created from the pre-operative images. After the robot is calibrated and registered, the planned tool-tip path and 3D model are transformed into the robot coordinate system. This paper focuses on a method to generate virtual fixtures derived from complicated geometry, which is the shadow part of Fig. 2.

In our collaborative system, the surgeon is in the control loop; s/he is able to control the progress of the tool along the constrained path. The system reads the surgeon's input and combines it with the planned tool-tip trajectory and the current tool-tip position to create the spatial motion for the tool-tip. Meanwhile, tool-shaft boundary motion constraints are generated from the registered 3D geometry model and the current tool position. With some other constraints, such as joint limitations of the robot, all these constraints are fed into the constrained optimization control algorithm to obtain the robot joint velocities

## III. CONSTRAINED CONTROL ALGORITHM OVERVIEW

It is important to be able to place absolute bounds on the motion of the instrument in the constrained working environment. Within these bounds, the controller should try to place the instrument tip as close to the desired position as possible. The basic control loop may be summarized as follows:

***Step 0:*** We assume that the robot is holding the surgical instrument, and that a model of the patient's anatomy has been obtained and registered to the coordinate system of the robot.

***Step 1:*** Describe a desired incremental motion of the surgical instrument, based upon (a) surgeon inputs, such as may be obtained from a joystick or hands-on cooperative force control; (b) an *a priori* surgical task description; (c) real time sensor feedback, such as might be obtained from a vision sensor. This description may include both an objective function describing desired outcomes (e.g., move as close as possible to a target) and motion constraints (e.g., avoid collisions, do not exceed robot joint limits, do not permit position errors to exceed specified limits, restrict tip motion to remain within a desired envelope, etc.). The desired

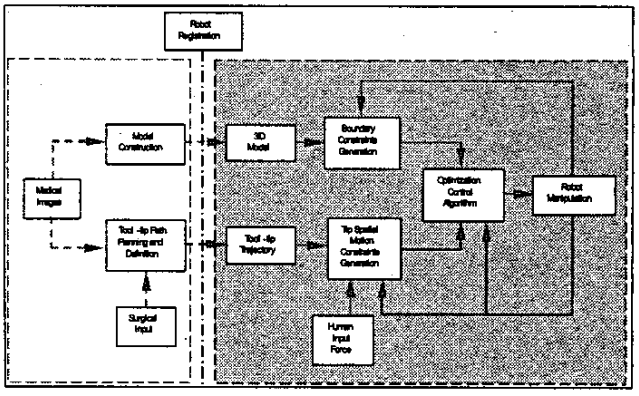


Fig. 2 Virtual fixture generation system

incremental motion is described as the solution to a constrained optimization problem.

***Step 2:*** Use the robot and task kinematic equations to produce a new linearized optimization problem, in which instrument motion variables and other task variables have been projected onto incremental joint variables. This problem has the general form:

$$\begin{aligned} &\arg\min \|W \cdot \Delta x - f\| \\ &H \cdot \Delta x \geq h \\ &\Delta x = J \cdot \Delta q \end{aligned} \tag{1}$$

where $\Delta q$ is the desired incremental motion of the joint variables and $\Delta x$ is an arbitrary vector of task variables. Different components of the optimization function may be assigned different relative weights, so that the errors of critical motion elements are close to zero, while errors in other non-critical motions simply stay as low as possible within tolerances allowed by the constraint set.

***Step 3:*** Use known numerical methods [8] to compute incremental joint motions $\Delta q$, and use these results to move the robot.

***Step 4:*** Go back to Step 1.

## IV. SYSTEM IMPLEMENTATION

### A. Tip Spatial Motion Constraint Generation

Our method for implementing guidance virtual fixtures for tool-tip spatial motion is described in an earlier paper by Marayong, Li, *et al.* [5]. For completeness, we will summarize key details here. We will model the tool tip as a Cartesian "robot" whose position $x(t) \in SE(3)$ and velocity $v(t) \in R^3$ are related to the robot's joint positions $q(t)$ and the relations

$$\begin{aligned} x(t) &= Kins(q(t)) \\ v(t) &= J(q(t))\dot{q}(t) \\ J_{ij}(t) &= \frac{\partial Kins(q)_i}{\partial q_j} \end{aligned} \tag{2}$$

The tip's motion is to be controlled by applying forces and torques on the handle of the instrument, transformed to produce forces $f \in R^3$ resolved into the coordinate system of the tool tip. We define a reference direction of motion $D = D(t)$, and span and kernel projection operators

$$\begin{aligned} Span(D) &\equiv [D] = D\left(D^T D\right)^+ D^T \\ Ker(D) &\equiv \langle D \rangle = I - [D] \end{aligned} \tag{3}$$

We then define a function $u = u(x,S)$ computing a signed distance from the tool tip to the task motion target (e.g., an anatomical feature or desired tool path). The new preferred direction is then defined as

$$D_c(x) = (1 - k_d)[D]f + k_d \|f\| \langle D \rangle u \tag{4}$$

where $k_d$ is a blending coefficient with $0 \le k_d \le 1$, which governs how quickly the tool is moved toward the reference direction. We then define an admittance control law

$$v_{tip-desired} = k\left([D_c] + k_r \langle D_c \rangle\right) \tag{5}$$

where $v_{tip-desired}$ is the desired tip velocity, $k$ is the admittance gain and $k_r$ $(0 \le k_r \le 1)$ is an admittance ratio that attenuates the non-preferred component of the force input.

Our desired 3D Curve $D$ is generated by B-spline interpolation. As mentioned, the tool-tip trajectory is planned by picking positions on the pre-operative medical image slices. After registration, all these points are transformed into the robot coordinate system. We then interpolate a B-spline curve to fit these sample points. At each control loop, the robot encoders read out the current tool-tip position. We search the closest point on the B-spline to the current tool-tip position and compute the tangent direction of the B-spline at that point. If $t_x$, $t_y$ and $t_z$ are the components of the tangent to the curve, and $n_x$, $n_y$ and $n_z$ are the components of the vector from current position to the closest point on the curve, then we have $D = \begin{pmatrix} t_x & t_y & t_z \end{pmatrix}^T$ and $u = \begin{pmatrix} n_x & n_y & n_z \end{pmatrix}^T$.

*B. Boundary Constraint Generation*

We use 3D triangulated surface models of anatomy to develop real-time constraints on tool motion. Production of such anatomic models from 3D medical images and registration of the models to robotic workspaces are well-established techniques [9].

However, the models are geometrically complex; generating constraints in real time can be a challenge. In the current work, we model the surgical tool as one or more "fat" line segments representing the tip and tool shaft. Boundary constraints are generated from closest-point pairs $P_b$ and $P_t$ on the geometry boundary and the tool respectively.

In our work, we use a covariance tree data structure [10] to search for the closest point on the surface to the tool. A covariance tree is a variant of a k-dimensional binary tree (k-D tree). The traditional k-D tree structure partitions space recursively along principal coordinate axes. In our covariance tree, each sub-space is defined in the orthogonal coordinate system of the eigenvectors centered at the center of mass of the point set, and is recursively partitioned along this local coordinate frame. We rotate each local reference frame so that the *x-axis* is parallel to the eigenvector with the largest eigenvalue and the *z-axis* is parallel to the eigenvector with the smallest eigenvalue. An important advantage of covariance trees is that the bounding boxes tend to be much tighter than those found in conventional k-D trees and tend to align with surfaces, thus producing a more efficient search.

The tree is built recursively as follows. Given a set of points $\{s_i\}$ sampled from the surface, we find the centroid and moments

$$\begin{aligned} \bar{s} &= \frac{1}{N}\sum_{i=1}^{N} s_i \\ M &= \sum \tilde{s}_i \cdot \tilde{s}_i^T \quad \text{where} \quad \tilde{s}_i = s_i - \bar{s} \end{aligned} \tag{6}$$

We then compute the eigenvalue decomposition of $M$ and determine a rotation $R$ with columns $R_x$ and $R_z$ corresponding to the largest and smallest eigenvalues, as discussed above. We then rotate the $\tilde{s}_i$ to compute $\check{s}_i = R \cdot \tilde{s}_i$. Finally, we compute the bounding box of the $\check{s}_i$ and partition the $\check{s}_i$ about the cutting plane $x = 0$. Each of these sets is thus recursively subdivided until only a small number of points remain in each box.

In our case, we must modify this procedure because our surface model is composed of small triangles, not sample points. To do this, we fit a circumsphere about each triangle. We then construct a covariance tree of the centers of each circumsphere. At each node (i.e. for each bounding box) of the tree, we note the maximum radius $r_{\max}$ of all circumspheres associated with that node. We compute the corners of an expanded bounding box

$$(\check{s}_{\min}, \check{s}_{\max}) = (-r_{\max} + \min_i \check{s}_i, r_{\max} + \max_i \check{s}_i) \tag{7}$$

where the min & max operations are performed element-wise.

Tree searching proceeds as follows. Given a line segment $(a,b)$ between two points $a$ and $b$, we wish to find all possible surface patches that may be closer than some threshold distance $e_{thresh}$ to the line segment. At each level of the tree, we first transform the line segment to the local coordinates $(\check{a},\check{b}) = \left(R^{-1} \cdot (\bar{a} - \bar{s}), R^{-1} \cdot (\bar{b} - \bar{s})\right)$. If we are at a terminal node of the tree, then for each triangle $t$ in the node, we compute the point pair $(\check{c}_t, \check{d}_t)$ corresponding to the closest approach between the line segment and triangle, where $\check{c}_t$ is on the triangle and $\check{d}_t$ is on $(\check{a},\check{b})$. We retain any such pairs with $\|\check{c}_t - \check{d}_t\| < e_{thresh}$. For a non-terminal node, we compute the point of closest approach $\check{c}_{node}$ and a corresponding point

$\bar{d}_{node}$ on the line segment. If the distance $\left\|\bar{c}_{node} - \check{d}_{node}\right\| < e_{thresh}$, we recursively search the left and right sub-trees for points of close approach to $(\bar{a}, \check{b})$. When the search is complete, we transform all point pairs back into the world coordinate system. If there are no point pairs closer than $e_{thresh}$, we generate no boundary constraints for the current time step of the control.

One difficulty with this approach is that it tends to produce many point pairs from the same patch of anatomy that are almost equivalent, thus producing an excessive number of motion control constraints. Therefore, in practice we modify our search as follows: For any non-terminal node with $\left\|\bar{c}_{node} - \bar{d}_{node}\right\| < e_{thresh}$ that is very flat (i.e., with $\bar{z}_{\max} - \bar{z}_{\min}$ smaller than a specified value) and for which the line segment does not penetrate the bounding box, we simply return the point pair $(\bar{c}_{node}, \check{d}_{node})$ without further recursion.

### C. *Control Algorithm Implementation*

We have experimented with both velocity and incremental step position control. In the discussion below, we will use the latter. Thus, at each time step, the goal is to compute incremental joint motions $\Delta q$, which then are fed to low-level position servos.

We compute desired tool tip velocity using the admittance law described above, and convert this to an incremental 6-DOF tool motion $\Delta P_{tip-des} = \left(\left(v_{tip-desired}\Delta t\right) \| (0,0,0)\right)^T$, where $\Delta t$ is the sample interval. We identify 3 classes of requirements: in the tip frame, tool boundary frame, and in joint space. For each, we define an objective function $\zeta$ to be minimized and a set of linearized constraints.

***Tool tip motion:*** We require that an incremental tool tip motion be as close as possible to some desired value. We express this as:

$$\begin{aligned} \zeta_{tip} &= \left\|\Delta P_{tip} - \Delta P_{tip-des}\right\|^2 \\ \Delta P_{tip-des}{}^T \cdot \Delta P_{tip} &\geq 1 - \varepsilon \end{aligned} \tag{8}$$

where $\varepsilon$ is a small positive number (0.01 in our experiments). We relate tool frame motion to joint motion via the Jacobean relationship $\Delta P_{tip} = J_{tip}(q)\Delta q$. We rewrite (8) as

$$\begin{aligned} \zeta_{tip} &= \left\|W_{tip} \cdot \left(J_{tip}(q)\Delta q - \Delta P_{tip-des}\right)\right\|^2 \\ H_{tip-des} J_{tip}(q)\Delta q &\geq h_{tip} \end{aligned} \tag{9}$$

where $H_{tip-des} = \Delta P_{tip-des}{}^T$, $h_{tip} = 1 - \varepsilon$ and $W_{tip} = diag\{w_{tip}\}$ denotes a diagonal matrix of weighting factors specifying the relative importance of each component of $\Delta P_{tip}$. Since we want to track the path tightly, we set $w_{tip}$ to a fairly high value (1 in the current experiments).

***Boundary constraints:*** Since the instrument is inserted into a cavity, we want to ensure that the instrument itself will not collide with the cavity boundary as a result of the motion. For each potential contact point pair we get a constraint of the general form

$$n_b{}^T \cdot \left(P_k + \Delta P_k - P_b\right) \geq \varepsilon \tag{10}$$

where $P_k$ is the position of the potential collision point on the tool and $P_b$ is the position of the potential contact point on the surface. $P_k$ and $P_b$ are the closest point pair we generated by the method described in last section. $n_b$ is the normal of the contact point on the surface, and $\varepsilon$ is a small positive number (0.01 in our experiments). Constraint (10) indicates that the angle between $\left(P_k + \Delta P_k - P_b\right)$ and $n_b{}^T$ is less than $90^\circ$. We can also define an objective function $\zeta_k = \left\|W_k \Delta P_k\right\|$ expressing the desirability of minimizing extraneous motion of the tool near the boundary, and can again rewrite these formulae in terms of $\Delta q$:

$$\begin{aligned} \zeta_k &= \left\|W_k \Delta P_k\right\| \\ H_k J_k(q)\Delta q &\geq h_k \end{aligned} \tag{11}$$

Currently, we use very low values (0.001) for $w_k$ and rely mainly on the inequality constraints. An alternative would have been to leave the $\zeta_k$ term out of the optimization altogether. The number of boundary constraints is dynamically changed because it depends on how many closest-point pairs we generate based on the relative position of the tool and the geometry constraint.

***Joint limits:*** Finally, we want to ensure that none of the joint limits are exceeded as a result of the motion. This requirement can be stated as, $q_{\min} - q \leq \Delta q \leq q_{\max} - q$, where $q$ is the vector of the current values of the joint variables, and $q_{\min}$ and $q_{\max}$ denote the vectors of lower and upper bounds on the joint variables respectively. We also want to minimize the total motion of the joints. This can be rewritten in the form

$$\begin{aligned} \zeta_{joints} &= \left\|W_{joints}\Delta q\right\|^2 \\ H_{joints}\Delta q &\geq h_{joints} \end{aligned} \tag{12}$$

where

$$H_{joints} = \begin{bmatrix} I \\ -I \end{bmatrix}, \quad h_{joints} = \begin{bmatrix} q_{\min} - q \\ -(q_{\max} - q) \end{bmatrix} \tag{13}$$

Again, we set $w_{joints}$ to 0.001 and simply enforce the inequality constraints.

***Putting it together:*** We combine all the task constraints and objective functions, and then obtain the overall optimization problem, which is:

$$\zeta = \min_{\Delta q} \left\| \begin{bmatrix} W_{tip} & & \\ & W_k & \\ & & W_{joints} \end{bmatrix} \cdot \left( \begin{bmatrix} J_{tip}(q) \\ J_k(q) \\ I \end{bmatrix} \Delta q - \begin{bmatrix} \Delta P_{tip-des} \\ 0 \\ 0 \end{bmatrix} \right) \right\|^2 \tag{14}$$

subject to

$$\begin{bmatrix} H_{tip} & & \\ & H_k & \\ & & H_{joints} \end{bmatrix} \cdot \begin{bmatrix} J_{tip}(q) \\ J_k(q) \\ I \end{bmatrix} (\Delta q) \geq \begin{bmatrix} h_{tip} \\ h_k \\ h_{joints} \end{bmatrix} \quad (15)$$

which can be solved numerically using the method of Lawson and Hanson [8] for the set of joint displacements $\Delta q$, satisfying the constraint (15) and minimizing the error norm of (14).

## V. Experiment and Results

We have evaluated our approach using the experimental setup shown in Fig. 3. A thin wire attached inside the nasal cavity of a plastic skull served as the target path, and the benchmark task was to trace this path with the tip of a bent pointer without colliding with the walls of the cavity. Five small radioopaque fiducials were implanted in the skull, which was then CT-scanned. The positions of the fiducials in the CT image were determined by standard image processing methods. We used a Northern Digital Optotrak® 3D tracking system to perform registration of CT coordinates to robot coordinates and (coincidently) to the Optotrak coordinates by standard methods. Essentially, an Optotrak probe was used to locate each fiducial while observing additional Optotrak markers on the robot and then the appropriate transformations were computed using the method of [12].

We used 3D-SLICER [11] for anatomy modeling and interactive display. 3D-SLICER's built-in segmentation functionality was used to threshold, segment and render the skull CT dataset to create a triangulated model of the skull surface. For the current experiment, we only used the nose and sinus portion of the resulting skull model. The 3D-SLICER skull surface model and the model of the sinus portion are shown in Fig. 4. There are about 99,000 vertices and 182,000 triangles in this surface model, all of which were transformed to robot coordinates after registration.

Our current implementation uses the JHU Steady-hand robot [2]. Steady-hand robotic mechanisms are coupled with computation to enhance human capabilities at highly skilled, interactive tasks. It is a 7 DOF remote-center-of-motion (RCM) manipulator with high position resolution. A force sensor is integrated at the end-effector and the operator holds a surgical pointer mounted to the force sensor.

In our path following experiment, we defined the target path with respect to CT space by tracing the wire with the tip of an Optotrak pointer. We gathered multiple sample points along the path and then interpolated a 5-th degree B-spline curve to fit these sample points.

The current tool-tip position with respect to the robot can be determined from the robot joint encoders and the forward kinematics of the Steady-hand robot. First, we manually choose a point on the curve close to the current tip position and guide the tool tip to the curve. Then the system reads the user's force input and guides the user to follow the B-spline curve. The sampling frequency of position and force data used in our algorithm is 30Hz, although a PID controller for

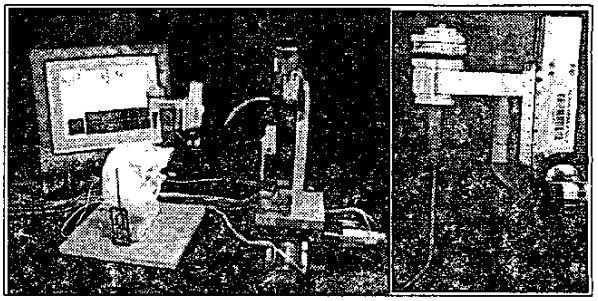

Fig. 3 Experimental apparatus showing the JHU Steady-hand robot and skull phantom (left) and close-up of pointer tool (right)

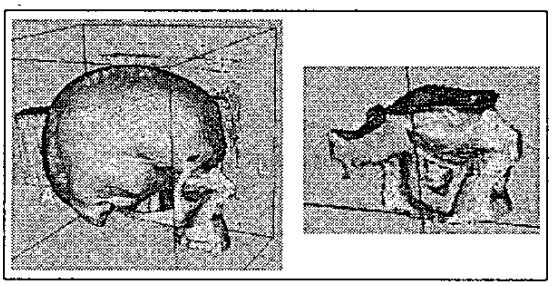

Fig. 4 3D-SLICER [11] surface models of test phantom. (left) whole skull surface model and, (right) the model of nasal cavity portion used to generate constraints

servoing the robot position runs on a dedicated motion control-board at a much higher rate. The tool-tip position is computed from the robot encoders and transformed back to CT space. As a comparison, we also gathered freehand data. A user held the tool with an Optotrak rigid body affixed, and moved the tool through the sinus cavity to follow the wire as close as possible. The tip position was also transformed into CT space. In both cases, the position of the tool is displayed on the 3D-SLICER interface. This provides the user with visual feedback of the task.

We choose the admittance ratio $k_r$ as 0 to enforce the tool-tip motion only along the preferred direction; control gain $k_d$ as 0.2. Our experimental results (TABLE I) show that an experienced user can perform the guidance task with twice accuracy and in 65% the time with robot assistance compared to freehand execution. The average error of 5 trials by robot guidance is 0.76mm and the average time is 17.24s; the average error for freehand is 1.82mm while the average time is 26.61s. During the path following task, the tool itself did not hit the bone. The robot calibration and system registration errors are the two main sources of tip motion error. In our experiment, the residual registration error measured across the 5 fiducials is 0.425mm. Although it is small, the backlash of the robot also contributes to the robot guidance tip motion error. Fig. 5 shows tool trajectories and the relative position of the tool with the nasal cavity model. Fig. 6 shows the complete error of the system. In our task, the size of $\Delta q$ in (1) is $7\times1$. $H \cdot J$ varies from $20\times7$ to $39\times7$, $W \cdot J$ varies from $13\times7$ to $37\times7$ according to the number of boundary

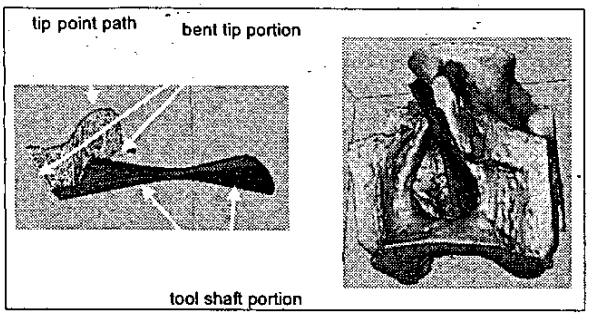


Fig. 5 Trajectories of the tool during the path following procedure. (left) the swept volume of the tool path, (right) the relative position between the tool and the nasal cavity.

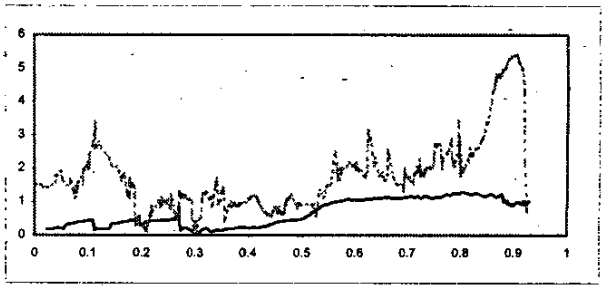


Fig. 6 Magnitude of position error using Robot guidance (solid line) and freehand (dashed line). x-axis(mm): the parameter in B-spline parameter domain, y-axis(mm): the magnitude of the position error

constraints. With our 2GHz Pentium IV PC, the average time in each control interval for the boundary search and optimization problem solution was 5.54ms.

## VI. Conclusion

ENT surgery presents a constrained working environment for both the surgeon and the mechanical devices designed to assist them. The control for the medical design of the devices must reflect these constraints. In this paper we outlined and implemented a guidance virtual fixture generated by complex 3D anatomy.

Robot guidance using cooperative control and virtual fixtures derived from complex geometry can assist users in skilled manipulation tasks, while maintaining desirable properties such as collision avoidance and safety. The experiments reported here used a simple task with constraints similar to those that might be encountered in ENT surgery. The results showed significant improvement in both accuracy and execution time, compared to free-hand instrument manipulation.

We are continuing to evaluate these methods and further integrating these components into a full frontal sinus surgery system. Meanwhile we are valuating other robot registration methods to improve the registration accuracy. In the future, we will introduce these types of spatial motion constraints to tele-manipulation, which is another popular surgical robot control mode. We will compare the performance using such virtual fixtures between the collaborative and tele-manipulative robot systems.

TABLE I Experimental results – robot assistance vs. freehand.

| Trial # | Robot guidance | | Freehand | |
|---|---|---|---|---|
| | Average Error (mm) | Average Time (s) | Average Error (mm) | Average Time (s) |
| 1 | 0.736 | 18.972 | 1.785 | 26.354 |
| 2 | 0.757 | 15.275 | 1.632 | 29.358 |
| 3 | 0.765 | 16.290 | 1.796 | 27.372 |
| 4 | 0.779 | 19.439 | 2.061 | 25.436 |
| 5 | 0.777 | 16.209 | 2.119 | 24.533 |
| avg | 0.763 | 17.237 | 1.819 | 26.611 |
| std | 0.395 | 1.848 | 1.126 | 1.863 |

## Acknowledgment

Partial funding of this research was provided by the National Science Foundation under grants EEC9731748 (CISST ERC), IIS9801684, IIS0099770, and IIS0205318. The authors also gratefully acknowledge the advice and collaboration of Dr. Masaru Ishii, M.D., Dr. Allison Okamura, Dr. Greg Hager, Dr. Peter Kazanzides, Greg Fischer, Anand Viswanathan, Panadda Marayong, and the rest of the CISST ERC "crew".

## References

[1] Taylor, R. and L. Joskowicz, *Computer-Integrated Surgery and Medical Robotics*, Standard Handbook of Biomedical Engineering & Design, M. Kutz, Editor. 2002, McGraw-Hill. p. 29.3-29.45.

[2] Taylor, R., P. Jensen, L. Whitcomb, A. Barnes, R. Kumar, D. Stoianovici, et al. *A Steady-Hand Robotic System for Microsurgical Augmentation.* International Journal of Robotics Research, 1999. 18(12).

[3] Kumar, R., T. Goradia, A. Barnes, P. Jensen, L. Whitcomb, D. Stoianovici, et al. *Performance of Robotic Augmentation in Microsurgery-Scale Motions.* 2nd Int. Symposium on Medical Image Computing and Computer-Assisted Surgery. 1999. Cambridge, England: Springer, September 19-22 p.1108-1115.

[4] Bettini, A., S. Lang, A. Okamura, G. Hager, *Vision Assisted Control for Manipulation Using Virtual Fixtures.* IEEE/RSJ International Conference on Intelligent Robots and Systems, 2001: p. 1171-1176.

[5] Marayong, P., M. Li, A. Okamura, G. Hager *Spatial Motion Constraints: Theory and Demonstrations for Robot Guidance Using Virtual Fixtures.* International Conference on Robotics and Automation. 2003. Taiwan, September 15-19 p.1954-1959.

[6] Funda, J., R. Taylor, B. Eldridge, S. Gomory, K. Gruben, *Constrained Cartesian motion control for teleoperated surgical robots.* IEEE Transactions on Robotics and Automation, 1996 p.453-465

[7] Li, M. and R. Taylor. *Optimal Robot Control for 3D Virtual Fixture in Contrained ENT Surgery.* Medical Image Computing and Computer-Assisted Interventions (MICCAI). 2003. Montreal, Nov 15-18 p.165-172.

[8] Lawson, C. and R. Hanson, *Solving Least Squares Problems.* 1974, Englewood Cliffs, NJ: Prentice-Hall.

[9] Taylor, R.H., et al., eds. *Computer-Integrated Surgery.* 1996, MIT Press: Cambridge, Mass.

[10] Williams, J., R. Taylor, and L. Wolff. *Augmented k-d techniques for accelerated registration and distance measurement of surfaces.* Computer Aided Surgery: Computer-Integrated Surgery of the Head and Spine. 1997. Linz, Austria, September p.P01-21.

[11] 3D Slicer web site,http//www.slicer.org,2003.

[12] Horn, B.K.P., *Relative Orientation.* International Journal of Computer Vision, 1990. 4: p. 59-78.