Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems
September 28 - October 2, 2004, Sendai, Japan

Scale-Invariant Registration of Monocular Stereo Images to 3D Surface Models

Darius Burschka, Ming Li, Russell Taylor and Gregory D. Hager
Department of Computer Science
Johns Hopkins University, Baltimore, USA
e-mail:{burschka|liming|rht|hager}@cs.jhu.edu

***Abstract*— We present an approach for scale recovery from monocular stereo images of an endoscopic camera with simultaneous registration to dense 3D surface models. We assume the camera motion to be unknown or at least uncertain. An example application is the registration of endoscope images to pre-operative CT scans that allows instrument navigation during surgical procedures. The application field is not restricted to the medical field. It can be extended to registration of monocular video images to laser-based surface reconstructions in, e.g., mobile navigation area or to autonomous aircraft navigation from topological surveys. A novel way for depth estimation from arbitrary camera motion is presented.**

In this paper, we focus on the robust initialization of the system and on the scale recovery for the reconstructed 3D point clouds with accurate registration to the candidate surfaces extracted from the CT data. We provide experimental validation of the algorithm with data obtained from our experiments with a phantom skull.

I. MOTIVATION

In endonasal surgery and other medical minimally invasive procedures, an endoscopic camera is used to provide information about the current instrument location to the surgeon. Since some sinus surgical procedures are close to optic nerves, the eyes, and the brain, surgeons require the best possible information during the surgery to guide the surgical instruments inside of the nasal cavities (Fig. 1).

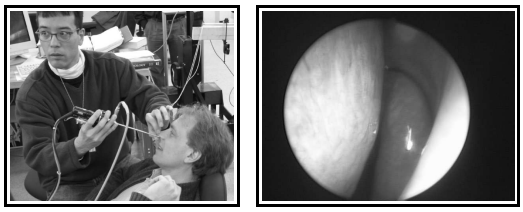

Fig. 1. Endoscopic inspection of the nasal Sinus cavities depicting the limited information provided to the surgeon.

The necessary information is supplied primarily through the endoscope requiring from the surgeon a good knowledge of anatomy.

Computer Integrated Surgery techniques has been employed in the endonasal approach. Image-guided surgery system (IGS), that integrates pre-operative medical images with the endoscope information, supplies valuable information beyond what the endoscope alone can provide.

In IGS, data from a preoperative CT scan is downloaded into the computer in the operating room and localizers are attached to surgical instruments. Once the patient's head position is registered, the software provides the surgeon with the 3D location of the tip of the instrument visualized relative to the pre-operative CT-scan. The current frameless registration method for IGS is based on anatomic fiducial points or based on contour mapping [6]. The accuracy of registration is crucial for any image-guided system. If registration errors occur, the localization accuracy suffers. Therefore, the operating surgeon must realign the registration at several points in the operative field throughout the procedure.

In our *JHU-Steady Hand Surgical Robot System* [9], we incorporate the surgeon in the loop. The surgeon holds the instrument and moves it the same way s/he does it in traditional surgery. The robot reads surgeon's input and provides appropriate assistance to him, such as: to avoid collisions between instruments and important anatomical structures, and to guide the surgeon to the target accurately. With the assistance of the robot, the surgeon is released from dealing with trivial issues such as the fine control of the motion of the instruments. S/he can concentrate on the surgical region of interest [7]. In this kind of a robotic assistant system, the registration accuracy and the real-time aspects are crucial. In our system, we use a new registration method, which can validate the pre-surgical registration and update the system in real time without distracting the surgeon with the additional task of landmark localization. In this way, the autonomy of the navigation system is significantly increased.

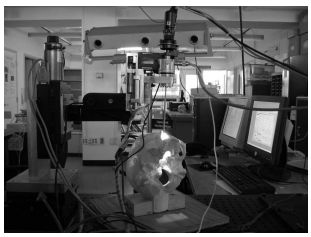

Fig. 2. Our experimental system with the *Johns Hopkins Steady Hand Robot* and an OPTOTRAK system for accuracy validation.

Our approach addresses the problem of registration of the endoscopic images to the pre-operative CT-scan without the usage of external fiducials. The presented system recovers a scaled 3D structure of the inspected endonasal cavity from endoscopic images and registers it to the surface points of the CT scan. This

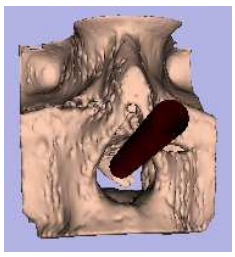

Fig. 3. The Endoscope relative to the CT scan.

way, the position of the instrument is localized in the frame of the CT scan that can now be used for virtual fixtures and path planning (Fig. 3) [4].

The presented task requires a *Simultaneous Localization and Mapping* capability in the given environment. The 3D structures need to be reconstructed in parallel to the localization from the monocular image stream. The problem of *Simultaneous Localization and Mapping*, also known as SLAM, has attracted immense attraction especially in the mobile robotics literature. SLAM addresses the problem of building a map of an environment from a sequence of landmark measurements obtained from a moving system. Since the motion especially of hand-operated devices is unknown, the mapping problem induces a localization problem to register the image frames relative to each other. The dominant approach to the SLAM problem was introduced in a seminal paper by Smith, Self, and Cheeseman [8]. This paper proposed the use of the extended Kalman filter (EKF) for incrementally estimating the posterior distribution over the robot pose along with the positions of the landmarks. Many popular SLAM implementations use laser range information as input to the process to simplify the estimation process to a pure localization and mapping since laser range finders estimate directly the 3D locations of the imaged points. We extended this approach to a vision-based system where the information from a monocular camera is used as input [1], [2].

The paper is structured as follows. In Section II, we describe the underlying processing that allows us to recover the 3D-structure and the motion from monocular image of the endoscopic camera and the way we align the data using a modified ICP approach. In Section III, we present some experimental results on the phantom skull. We conclude in Section IV with an evaluation of the presented approach and present our future research goals.

II. Approach

We presented already in [1] a system that localizes a monocular camera based on tracked landmarks in the image. This system uses a *Sum of Square Distances* (SSD) tracking algorithm [3] to establish correspondences between two images. The geometrical relation between the tracked features was known from a *teaching step*. We presented an extension to this approach that performs geometrical mapping of the features in parallel to the camera localization (SLAM) and, therefore, omits the necessity of a dedicated *teaching step* in [2] that was necessary to build the model representation a-priori. The problem in our SLAM extension is the recovery of the correct scale for the reconstruction. In case of a mobile robot, the correct scale is not always necessary. In applications, where the system is used as a replacement for the inertial unit, merely the orientation of the robot may be of interest. On the other hand, the scale can also be recovered using the odometry information. The situation is different in the presented case of endoscopic surgeries. Our goal was the development of a camera navigation system that could be used in freehand endoscope procedures without the assistance of the robot as well. The 3D information of the CT scan that defines the reference frame for our localization turns out to provide all the necessary information for the scale recovery.

In this section, we recapitulate the key steps of our vision-based reconstruction, followed by a detailed discussion of the scale recovery and alignment of the camera data to the CT surface data.

A. Localization and Mapping Step.

A known solution for recovery of the camera motion, in cases where eight point correspondences between the two images can be established, is the *eight-point-algorithm*. The recovered *Essential Matrix* contains the information about the translation direction T' and rotation R between the images. The translation information can be recovered just up to a scale because of the way, how this matrix is constructed [10].

In our case, the number of corresponding (detectable) points between two camera frames varies significantly during the sinus surgery. There are situations, when less than eight points can be matched. The above approach fails in these cases, therefore, we use it just to bootstrap the system and switch to our new localization method requiring merely 3 point correspondences afterward. We will sketch out the process below.

1) **Feature Extraction:** The algorithm described below assumes that *point features* are extracted from the images, which uniquely represent a specific area in them. Possible features are: intersections of contours resulting from edge filters or the areas themselves.

The problem in real endonasal images is the sparse density of points that actually can be used for a model reconstruction. Another problem that we needed to address here is the moving light source, which is attached to the endoscope (Fig. 2). This violates the brightness constancy assumption in most common stereo algorithms. We obtained preliminary results showing that the blood vessel structure provides sufficient information for tracking (Fig. 1). In real images, we compensate the brightness variations by running a gradient filter on the original images and doing an SSD search on the resulting gradient images.

Our current results are based on experiments with a phantom skull. This skull does not have any detectable texture. We added colored points on the surface that we segment in the hue space of the color representation. This way, we are able to identify and track the point features in image sequences using a simple color blob tracker despite the changing lighting conditions (Fig. 4).

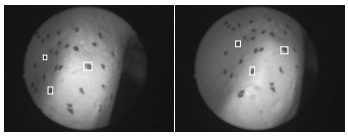

Fig. 4. Example of corresponding points on our phantom.

2) **Motion Estimation:** We assume that each 3D point P_i imaged in a unifocal camera frame $p_i = (u_i v_i 1)^T$ can be represented by its direction vector $n_i = p_i/||p_i||$ and the distance to the real point D_i to $P_i = D_i \cdot n_i$. Since in

typical applications the scale m of the reconstruction may be unknown, the system can also operate with a scaled version of the distance $\lambda_i = D_i/m$. In our approach, we calculate an estimate for the rotation $\tilde{\mathbf{R}}$ and the scaled translation $\mathbf{T}'^*$ between the points $\{P_i\}$ in the current frame and points $\{P_i^*\}$ in the next frame to

$$\begin{gathered} \bar{P} = \tfrac{1}{n}\sum_{i=1}^{n} P_i, \quad \bar{P^*} = \tfrac{1}{n}\sum_{i=1}^{n} P_i^*, \\ P_i' = P_i - \bar{P}, \quad P_i'^* = P_i^* - \bar{P^*}, \\ \tilde{\mathbf{M}} = \sum_{i=1}^{n} P_i'^* P_i'^{\mathbf{T}}, \quad [U\ D\ V^T] = \mathrm{svd}(\tilde{\mathbf{M}}), \\ \underline{\tilde{\mathbf{R}} = V \cdot U^T, \quad \mathbf{T}'^* = \bar{P^*} - \tilde{\mathbf{R}}^* \bar{P}.} \end{gathered} \tag{1}$$

The approach requires an initial knowledge of the values for λ_i for the first frame and it estimates a guess for translation $\mathbf{T}'^*$ and rotation $\tilde{\mathbf{R}}$ from the bootstrap procedure. In the first iteration step, it assumes $\lambda_i' = \lambda_i$ and, afterward, it iteratively converges to the true $\tilde{\mathbf{R}}, \mathbf{T}'^*$ by updating λ_i'. Details and simplifications of the algorithm are discussed in [1]. This algorithm requires a minimum of three corresponding points between both images to actually compute the pose difference between two camera frames $(\tilde{\mathbf{R}}, \mathbf{T}'^*)$, which makes it more suitable for the given application.

This step requires a knowledge about the scaled relation between distances to the observed features $\{\lambda_i\}$. The resulting 3D reconstruction has the same scale m as the initial distances λ_i.

3) **Initialization:** The function of the algorithm described in the previous section relies on the knowledge of the values λ_i for one of the images. During the steady state of the operation, these λ_i values are updated based on the motion estimates $(\tilde{\mathbf{R}}, \mathbf{T}')$ in (1).

In the initial step or in cases, where the number of available corresponding points between the two images is smaller then the minimum set of three points, there are two ways to initialize the system:

- **the eight-point-algorithm** based on the estimation of the *Essential Matrix* of the system from 8 point correspondences that provides the necessary information about $(\tilde{\mathbf{R}}, \mathbf{T}'^*)$;
- **manual feature selection** in the endoscope image, where the user selects three points with known correspondences to 3D surface data and the system uses this information to build a map of the entire space.

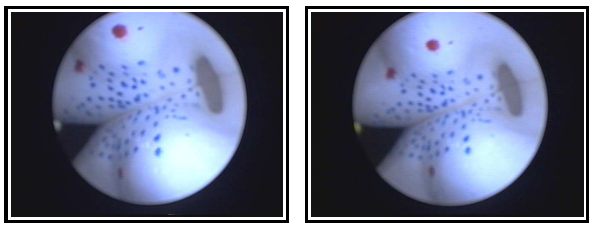

Fig. 5. Initialization based on known points (larger points in the image) with known geometrical distances to each other or from minimum of eight point correspondences between the smaller points.

Eight point correspondences allow us to calculate the so called *Essential Matrix* using the eight-point-algorithm [10]. The Essential Matrix $\tilde{\mathbf{E}}$ consists of the product of the two matrices $\tilde{\mathbf{E}} = \tilde{\mathbf{R}} \cdot \mathrm{sk}(\mathbf{T})$ with $\tilde{\mathbf{R}}$ being the rotation matrix and $\mathrm{sk}(\mathbf{T})$ being the skew matrix of the translation vector $\mathbf{T}$.

This allows us to calculate the rotation matrix $\tilde{\mathbf{R}}$. We can solve for normalized $\mathbf{T}'$ from $\tilde{\mathbf{E}}$ by calculating the position of the epipole [10] representing directly the translation vector between the two images.

This way we are able to calculate a modified form of the point transformation equation as (2)

$$\lambda_i^* \mathbf{n_i^*} = \tilde{\mathbf{R}} \cdot \lambda_i \mathbf{n_i} + \frac{\mathbf{T}}{m} \tag{2}$$

with an unknown scaling factor *m* [10].

We calculate the values for the scaled distances $\{\lambda_i^*, \lambda_i\}$ from (2) to

$$\begin{pmatrix} \lambda_i^* \\ \lambda_i \end{pmatrix} = \begin{pmatrix} \mathbf{n_i^*} & -\tilde{\mathbf{R}}\mathbf{n_i} \end{pmatrix}^{-1} \cdot \frac{\mathbf{T}}{m} \tag{3}$$

Eq. (3) has a similar form to the regular disparity equation for co-planar stereo systems with the important difference that it directly describes the depth relations on a complete surface of the projection sphere. Since the camera positions here are different for both images, the system estimates both depths directly instead of the disparity value that implies a constant depth from both images. The result is expressed in spherical instead of Cartesian coordinates, which allows simpler expressions for the depths. It corresponds better to the principal way how camera imaging works. A typical camera is measuring the angles of incident of the light rays losing the information about the radial distance to the imaged point.

We tested the other alternative - manual initialization - as well. In this case, we used points with known geometrical relations (large red dots in Fig. 5). The position of the points was verified with the *Northern Digital* OPTOTRAK system and used to initialize the system. This arrangement allows to estimate directly the scale of the reconstruction $m = 1$, but it has the disadvantage of the manual selection at the beginning to find the corresponding points. We simplified the procedure on the phantom skull by using dots with a different color. In real endoscope images, the surgeon has to define the initial correspondences.

In this case, we calculate the projections of $n \geq 3$ non-collinear points $\{G_i\}$ onto a virtual camera plane parallel to the plane $\mathcal{E}$ defined by them and by moving the focal point of the projection by a given distance h away from the point cloud. We estimate the rotation matrix $\tilde{\mathbf{R}}_e$ between the world coordinate frame of the points $\{G_i\}$ and the local camera frame as follows

$$\begin{gathered} \bar{G} = \tfrac{1}{n}\sum_{i=1}^{n} G_i, \quad v_1 = G_1 - G_2, \quad v_2 = G_3 - G_2, \\ \text{normal vector to} \quad \mathcal{E}: \quad n_e = \tfrac{v_1 \times v_2}{||v_1 \times v_2||} = \begin{pmatrix} n_{ex} \\ n_{ey} \\ n_{ez} \end{pmatrix} \\ n_\perp = \tfrac{(0\ n_{ez}\ -n_{ey})^T}{||(0\ n_{ez}\ -n_{ey})^T||} \Rightarrow \tilde{\mathbf{R}}_e = ((n_\perp \times n_e)\, n_\perp\, n_e) \end{gathered} \tag{4}$$

The rotation matrix $\tilde{\mathbf{R}}_e$ in (4) is used to rotate the points $\{G_i\}$ to $\{G_i^*\}$ in the coordinate frame of the camera parallel to the plane $\mathcal{E}$. The subtraction of the mean value $\bar{G}$ lets the optical axis in the virtual view intersect the mean point $\bar{G}$.

$$G_i^* = \tilde{\mathbf{R}}_e^T \cdot (G_i - \bar{G}) = (x_i, y_i, z_i)^T \tag{5}$$

Assuming that the projective geometry of the camera is modeled by perspective projection [5], a point G_i^*, which coordinates are expressed with respect to the camera with a focal length $f = 1$, will project onto the image plane at coordinates $p = (u_i, \nu_i)^T$.

We calculate the vector n_i and the initial length λ_i in (2) to

$$n_i = \frac{(u_i\ \nu_i\ 1)^T}{||(u_i\ \nu_i\ 1)^T||}, \quad \lambda_i = \sqrt{x_i^2 + y_i^2 + z_i^2}. \tag{6}$$

These values represent the distances and projections for the assumed coplanar camera view. We use them in the algorithm described in the section II-A.2 to estimate the initial values for the true orientation and distances to camera in the initial frame. This orientation is generally not identical with the coplanar assumption made in (4), but the true values are easily calculated using the presented iterative algorithm.

Note, that even though the estimates for the initial step may be erroneous, small deviations can be tolerated and the correct values for the distances $\{D_i\}$ can be estimated after the final alignment of the data with the 3D surface.

4) **Addition of New Features:** As already presented in [2], Eq. (1) updates the distance values for all tracked points P_i' for the new frame. New points can easily be added to the system utilizing the rigid body assumption for the imaged points by solving (7)

$$\begin{pmatrix} \tilde{\mathbf{R}}\mathbf{n_x} & -\mathbf{n_x^*} \end{pmatrix} \begin{pmatrix} \lambda_x \\ \lambda_x^* \end{pmatrix} = \tilde{\mathbf{R}}\lambda_1 \mathbf{n_1} - \lambda_1^* \mathbf{n_1^*} \tag{7}$$

or in a more robust way from 3 frames to (8)

$$\begin{pmatrix} \tilde{\mathbf{R_1}}\mathbf{n_x} & -\mathbf{n_x^*} & 0 \\ \tilde{\mathbf{R_2}}\tilde{\mathbf{R_1}}\mathbf{n_x} & 0 & \mathbf{n_x^{**}} \end{pmatrix} \begin{pmatrix} \lambda_x \\ \lambda_x^* \\ \lambda_x^{**} \end{pmatrix} =$$

$$= \begin{pmatrix} \tilde{\mathbf{R_1}}\lambda_1 \mathbf{n_1} - \lambda_1^* \mathbf{n_1^*} \\ \tilde{\mathbf{R_2}}\tilde{\mathbf{R_1}}\lambda_1 \mathbf{n_1} - \lambda_1^{**} \mathbf{n_1^{**}} \end{pmatrix}. \tag{8}$$

The pose change from image $1 \rightarrow 2$ is annotated here as $(\tilde{\mathbf{R_1}}, \mathbf{T_1})$ and the pose change between images $2 \rightarrow 3$ is annotated as $(\tilde{\mathbf{R_2}}, \mathbf{T_2})$. This equation estimates the distance λ_x to a new point P_x in the scale of an already known point P_1 from the currently tracked set of points. This way the newly added points are still measured with the same scaling factor m and the resulting 3D model has a uniform scale.

B. Scale Recovery for 3D Reconstruction

The scaling factor m in Section II-A depends on the scale of the λ_i-values for the initial set of points G_i (6). In case of an unsupervised bootstrap using the *eight point algorithm* (Sec. II-A.3) the resulting reconstruction has an arbitrary scale that depends on the scale of the translation vector $\mathbf{T}'$, which is usually assumed to be a unit vector.

The system has a usually a rough estimate of the current camera position. We use this estimate to carve out part of the CT surface data that falls into the expected visibility cone of the camera. This cone is slightly enlarged in all directions to compensate for the unknown camera motion. The size of the extracted patch depends on the uncertainty about the camera position.

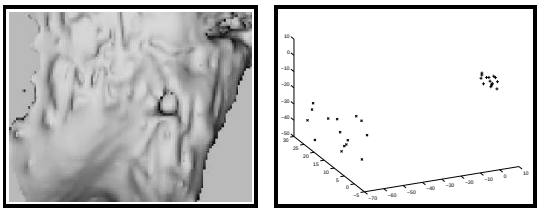

Fig. 6. Scaled reconstruction of surface points: (left) CT scan visualization of the area, (right) matched surface points with ICP {left point cloud}, scaled reconstructed points {right point cloud}.

The visible regions are usually surfaces with two dominant directions of the cavity walls with a third dimension representing the surface structure or combinations of such walls. We assume for now that the CT data patch consists of a single surface with some surface structure on it. We will discuss below, how to split more complex structures into simpler planar patches.

We use the property of two dominant surface directions for our scale recovery by calculating the covariance matrix $\widetilde{C}_k$ of the point cloud in the selected CT scan region and in the current camera reconstruction. The eigenvalues and eigenvectors of $\widetilde{C}_k$ define a new coordinate system with the two eigenvectors calculated from the larger eigenvalues defining the supporting plane in the cloud and the third eigenvector describing the depth variation in the measurement.

In both cases, the smallest eigenvalue (E_{ct}, E_{3d}) represents a metrics for the depth variations in the surface of the CT scan and in the reconstructed point cloud. The normalized eigen-vectors $\{V_{ctx}\}$ and $\{V_{3dx}\}$ and the eigenvalues allow us to calculate the scale m and the rotation $\tilde{R}_{tot}$ between the two data sets to (9). The rotation matrix $\tilde{R}_{tot}$ aligns both dominant surfaces along their normal vectors, which are represented by the eigenvector calculated from the smallest eigenvalue (last column in each of the rotation matrices in (9)). The rotation around the normal vector cannot be restored in this way.

$$\begin{gathered} m = \frac{\sqrt{E_{ct}}}{\sqrt{E_{3d}}}, \quad V_{p\in\{CT,3D\}} = (V_{px}\ V_{py}\ V_{pz})^T, \\ V_{n-p} = \frac{(0\ V_{pz}\ -V_{py})^T}{||(0\ V_{pz}\ -V_{py})^T||} \\ \tilde{R}_{ct} = \begin{pmatrix} (V_{n-ct} \times V_{ct}) & V_{n-ct} & V_{ct} \end{pmatrix}, \\ \tilde{R}_{3d} = \begin{pmatrix} (V_{n-3d} \times V_{3d}) & V_{n-3d} & V_{3d} \end{pmatrix}, \\ \tilde{R}_{tot} = \tilde{R}_{3d} \cdot \tilde{R}_{ct}^T \end{gathered} \tag{9}$$

We apply the scaling m and rotation $\tilde{R}_{tot}$ to the zero

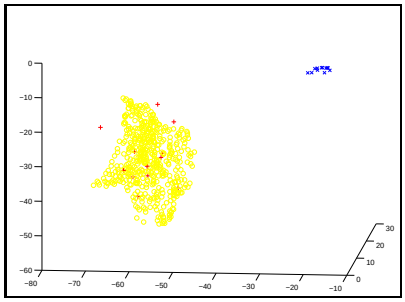

Fig. 7. After the alignment along the normal vector to the supporting plane the scale is roughly recovered, but rotation around the normal vector is possible.

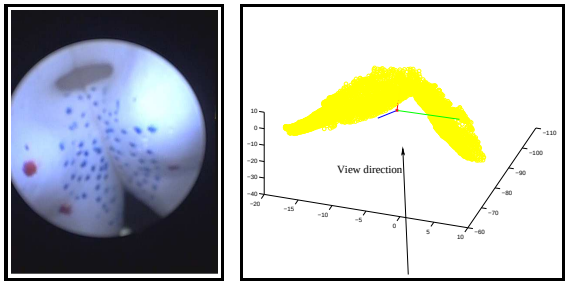

Fig. 8. The two surfaces need to be separated first, before our PCA-based scale recovery can be applied.

mean point clouds that were used to calculate the covariance matrices above. This way, we obtain two point clouds with the same alignment, but the CT-scan represents a much larger area because of the expected unpredictable camera movement. Both clouds have a similar scale and alignment of the supporting plane. Fig. 7 depicts an alignment result of the scaled point cloud in the top right corner to the 3D surface from the CT scan. The transformed points may have a significant rotational error around the normal vector visible in the above figure as crosses ('+'). We correct this error in the next step.

We consider now both rotated clouds as sparse "images", where each "pixel" is represented by its distance to the plane calculated from the covariance matrix. We use the reconstructed 3D structure from the current view as a template that is matched to the "image" constructed from the CT scan data using standard coarse-to-fine pattern matching techniques. Significant points $\{S_i\}_{ct}, \{S_j\}_{3D}$ with large deviation above a threshold ϵ_d from the supporting planes $\mathcal{E}_{ct}, \mathcal{E}_{3D}$ are identified in both "images" first. Three points from the set $\{S_j\}_{3D}$ are randomly picked. The most significant of them is matched to a similar value in $\{S_i\}_{ct}$ and the other two are searched based on distance from the first point and their value. This match is verified and refined based on the remaining points from the reconstruction. If a specific selection fails a new set of three points is generated from the $\{S_j\}_{3D}$ set. The process is similar to an SSD match with the only difference here that both "images" have different non-equidistant samplings.

The physical position of the sampling points, especially in the 3D reconstruction, does not necessarily correspond to the extreme values of the surface hull. Therefore, the above match trial can fail for a specific set of three points. The 3D reconstruction may not have reconstructed the peak value but some random value along the slope instead.

The resulting match is used to align the two data sets. The residual error is due to imperfect sampling and the coarse structure of the point sets, especially in the case of the reconstructed data from the phantom skull.

The 3D scaling step needs to be performed just in the initial bootstrap phase and in cases when the system was not able to maintain the minimum number of three features and needs to re-initialize the distance measurements.

In case, when the CT or the reconstructed data set contains two or more surfaces in a corner arrangement (Fig. 8), the structure needs to be split into single surfaces before we apply the above scaling. We use a modified version of a split-and-merge algorithm here. We establish for the entire data-set the plane equation of the supporting plane $\mathcal{A}$ based on $\widetilde{C}_k$ and (9). For each point P_i of the data-set, we calculate the distance d_i from the plane

$$d_i = (P_i - \bar{P}) \cdot V_n \tag{10}$$

with V_n being the normal vector of the estimated plane $\mathcal{A}$.

Point with the largest deviation from the original plane $\mathcal{A}$ and border points on both sides of the data set are used to split the original surface consecutively into sub-surfaces defined by these points. Each three points define one sub-surface. We split the points of the original surface into sub-surfaces depending on the distances to the resulting sub-planes. The above evaluation is repeated on the sub-surfaces until the maximum deviation $\max\{\mathrm{d_i}\}$ is smaller than the expected depth structure in the surfaces.

C. Iterative Closest Point (ICP) Alignment

At this point, we have reconstructed and localized the 3D dataset with endoscopic images, which has right scale and similar orientation and translation in the coordinate frame of the CT scan.

Rigid registration between CT images and physical data reconstructed by endoscopic images is achieved using the Iterative Closest Point (ICP) algorithm. For some applications in the endoscopic surgery, a deformable registration method can be further applied based on the results of the ICP.

We use a covariance tree data structure to search for the closest point for ICP. 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. 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 [11].

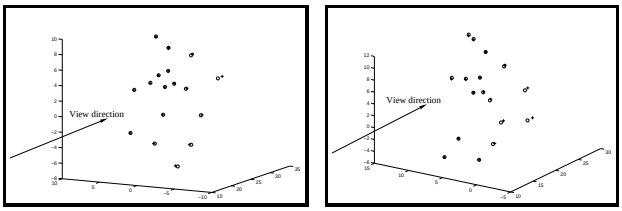

Fig. 9. 3D reconstruction results in camera coordinate frame from 2 reconstructions: (left) small rotation in addition to the translation, (right) significant rotation that still allows to detect the corresponding points in both images.

III. Experimental Results

Fig. 9 shows the results of the 3D reconstruction from our phantom skull marked with '+' compared to a reconstruction based on the OptoTrack information that we used as ground-truth for our experiments. In this case, the scale was $m = 1$ for better comparison between the 3D points.

The above results show that the system is capable of estimating the motion between two images without external information. The errors in the motion estimates for the above examples are: the rotational error expressed as Rodigues vector r=(0.0017, 0.0032, 0.0004), (-0.0123, -0.0117, -0.0052) and the translational error $\Delta T = (0.05, -0.398, 0.2172)^T, (-0.29, 0.423 - 0.4027)^T [\mathrm{mm}]$.

We tested our registration with different reconstruction results (patches) that were registered to CT skull images. Because the reconstructed 3D surface data may not cover the whole surface patch, we were interested in the sensitivity to drop-outs. We deliberately removed parts of the data from the reconstructed patch. Our experiments with the phantom show that the ICP can compensate noise levels in the data up to 0.6mm, combined with translational offsets up to 10mm, and rotational offset within 10 degrees. The vision-based reconstruction gives us errors an order of magnitude below these limits.

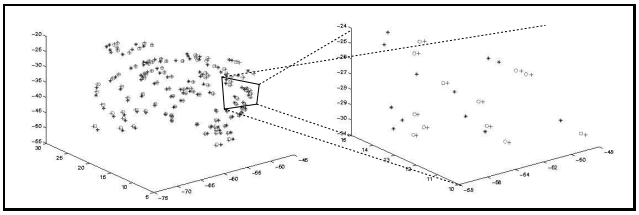

Fig. 10. The relative displacements of the sparse samples (+), their initial position recovered by VGPS(*) and their final position after alignment by ICP (o). Left is the global view of the sample data for a patch. Right is the close look.

After ICP alignment, the average distance error for the sample points is around 0.65mm. This compares favorably to the fiducial-based registration, whose residual error is around 0.40mm for four fiducials that are attached to the surface of the skull. However, our method directly tells the registration error of the target region for the surgery. The target residual error (TRE) calculated from fiducial residual error (FRE) is around 1.25mm.

The system is capable of running with a frame rate of 10Hz on a Pentium 4 2GHz processor running LinuxOS for the tracking and a SLAM part of the system.

IV. Conclusions and Future Work

The presented system allows an accurate reconstruction of 3D surface points and their registration to 3D surface data from CT scans or laser-range finder reconstructions. The points are successfully aligned with the surface models. In case our example application in endonasal surgery, the reconstructed points were successfully aligned with the CT scans of our phantom skull in the sinus area without the use of implanted fiducials. Our vision-based localization was an order of magnitude better than the fiducial-based method due to error accumulations in the steps involved in the other process.

Our major goal is to test more extensively our system in different parts of the skull and on other range images to better evaluate the performance of the system. We are currently investigating the feature type that can be used for a robust estimation and tracking of our *point features* in real endonasal images obtained in a preliminary experiment on a human subject.

Acknowledgments

Partial funding of this research was provided by the National Science Foundation under grants EEC9731748 (CISST ERC), IIS9801684, IIS0099770,and IIS0205318. This work was also partially funded by the DARPA Mars grant. The authors want to thank Dr. Masaru Ishii for his help in obtaining the preliminary data set of real endonasal images.

References

[1] Darius Burschka and Gregory D. Hager. V-GPS – Image-Based Control for 3D Guidance Systems. In *Proc. of IROS*, pages 1789–1795, October 2003.

[2] Darius Burschka and Gregory D. Hager. V-GPS(SLAM): – Vision-Based Inertial System for Mobile Robots. In *Proc. of ICRA*, April 2004. to appear.

[3] G.D. Hager and P. Belhumeur. Real-Time Tracking of Image Regions with Changes in Geometry and Illumination. *Proceedings of the IEEE Conference on Computer Vision and, Pattern Recognition*, pages 403–410, 1996.

[4] Gregory D. Hager. Human-machine cooperative manipulation with vision-based motion constraints. In *Proc. Workshop on Visual Servoing, with IROS*, 2002.

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

[6] D.W. Kennedy, W.E. Bolger, S.J. Zinreich, and J. Zinreich. *Diseases of the Sinuses: Diagnosis and Management*. 2001.

[7] M. Li and R.H. Taylor. Spacial Motion Constraints in Medical Robot Using Virtual Fixtures Generated by Anatomy. *Proceddings of the IEEE Conference on ICRA, April 2004*, page to appear, 2004.

[8] R. C. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. *Autonomous Robot Vehicles*, Springer-Verlag:167–193, 1990.

[9] R.H. Taylor, J. Patrick, L.L. Whitcomb, A. Barnes, R. Kumar, D. Soianovici, P. Gupta, Z. Wang, E. deJuan, and L. Kavoussi. A Steady-Hand Robotic System for Microsurgical Augmentation. *International Journal of Robotics Research*, 18:1201–1210, 1999.

[10] E. Trucco and A. Verri. *Introductory Techniques for 3-D Computer Vision*. Prentice Hall, 1998.

[11] J.P. Williams, R.H. Taylor, and L.B. Wolff. Augmented k-D Techniques for Accelerated Registration and Distance Measurement of Surfaces. In *Computer Aided Surgery: Computer-Integrated Surgery of the Head and Spine*, pages 1–21, 1997.