A System for Video-based Navigation for Endoscopic Endonasal Skull Base Surgery

Daniel J. Mirota, Student Member, IEEE, Hanzi Wang*, Senior Member, IEEE, Russell H. Taylor, Fellow, IEEE, Masaru Ishii, Gary L. Gallia, and Gregory D. Hager, Fellow, IEEE

Abstract—Surgery of the skull base require accuracy to safely navigate the critical anatomy. This is particularly the case for endoscopic endonasal skull base surgery (ESBS) where the surgeons work within millimeters of neurovascular structures at the skull base. Today’s navigation systems provide approximately 2 mm accuracy. Accuracy is limited by the indirect relationship of the navigation system, the image and the patient. We propose a method to directly track the position of the endoscope using video data acquired from the endoscope camera. Our method first tracks image feature points in the video and reconstructs the image feature points to produce three-dimensional (3D) points, and then registers the reconstructed point cloud to a surface segmented from pre-operative computed tomography (CT) data. After the initial registration, the system tracks image features and maintains the two-dimensional (2D)-3D correspondence of image features and 3D locations. These data are then used to update the current camera pose. We present a method for validation of our system, which achieves sub-millimeter (0.7 mm mean) target registration error (TRE) results.

Index Terms—Endoscopy, Registration, Image-guided treatment, Surgical guidance/navigation

I. INTRODUCTION

Endoscopic endonasal skull base surgery (ESBS) has gained much interest recently over traditional open surgical approaches as a treatment for pathologies involving the skull base and paranasal sinuses. Pituitary lesions, though generally benign, are the most common skull base lesion. ESBS is commonly used to treat pituitary lesions, non-neoplastic skull base lesions and other tumors of the skull base and nasal cavity. Traditional surgical approaches to the skull base are associated with significant morbidities because healthy cranial nerves are sometimes injured during surgery. Unlike traditional approaches, ESBS is less invasive and is shown to reduce operative time and decrease the length of hospital stay [1].

ESBS and traditional approaches are best contrasted with a clinical example. Fig. 1 shows the CT and magnetic resonance imaging (MRI) scans of a patient with a clival chordoma. The central location of this tumor makes it difficult to approach using traditional means, especially because the tumor involves the clivus and extends behind the carotid artery. This tumor was removed using an endoscopic endonasal approach. In this case the carotid artery was exposed via a vidian canal drill-out. The endoscopic images were taken before and just after the tumor was resected. Notice the carotid artery is completely free of the tumor at the end of the resection. Manipulating such high-value structures in cases like this is the reason that ESBS requires precise knowledge of patient anatomy. Thus, surgical navigation is key for success, especially to aid junior surgeons and for complex cases [1], as it provides the surgeon a means to both maintain orientation and to monitor progress.

In current practice, the surgeon uses a pointer tool to interact with the navigation system. The system tracks rigid bodies attached to the tool and the patient. During preparation for surgery, the rigid body attached to the patient is registered to fiducial markers on the patient. The rigid body defines the patient’s head in the navigation system. The navigation system in turn calculates a rigid-body transformation between the patient’s head and the tool to display the corresponding location in CT. A drawback of the current procedure is that each rigid-body transformation measurement contributes to localization error. In fact, localization error is typically quoted as 2 mm with a good registration, and can be much larger with a poor one [2]. Errors of this magnitude could lead to surgical error resulting in high morbidity or mortality.

To improve current navigation systems, we propose a new system that utilizes endoscopic video data for navigation. While endoscopic video presents many challenges including reflection, specularity, and low texture, our system robustly handles these challenges and creates a 3D reconstruction from video. The system then registers the 3D reconstruction to a pre-operative CT scan. After the initial registration, the system tracks the camera location by matching image features and performing robust 2D-3D pose estimation. Instead of relying on the long rigid-body transformation chain that current navigation systems

use, video-CT registration employs a more direct, accurate localization of the camera relative to the patient. We show sub-millimeter target registration error (TRE) results.

Tracking the location of a camera relative to CT has been studied in other areas of image-enhanced surgical navigation. In 2002, Shahidi et al. [3] presented a system for endoscope calibration and image-enhanced endoscopy. They achieved millimeter accuracy for a system using passive optical markers for tracking. More recently, Lapeer et al. [4] evaluated a similar system, again using passive optical markers for tracking and reported that sub-millimeter accuracy still remains elusive. Various tracker-based methods have appeared recently, including [5] and [6], where a system similar to an Optotrak was used to track the endoscope and provided an augmented display. Schulze et al. [7] reported a system using the Medtronic StealthStation along with novel rendering techniques to provide a side-by-side live versus virtual endoscopy. Daly et al. [8] similarly presented a tracker-based approach using a Polaris and demonstrated tracking with intra-operative cone-beam CT.

Video registration has been previously applied to bronchoscopy [9] where normalized mutual information was used to register to CT. In [10], visual tracking and registration was demonstrated on a skull phantom. The work here is an extension and generalization of [10]. Similar to our work, Allian et al. [11] presented an epipolar geometry method used to re-localize the endoscope image at a biopsy site. Our problem differs in that we aim to reconstruct, register and track the endoscope in a rigid environment, in contrast to localizing a single, usually defined, target. In addition, our work differs from [11] by using Singular Value Decomposition (SVD) Match [12] instead of block matching and the Adaptive Scale Kernel Consensus (ASKC) [13] instead of Maximum A Posteriori Sample Consensus (MAPSAC). Other closely related work used epipolar geometry for bronchoscopic localization in CT [14]. The most closely related work to our own was presented by Luo et al. [15] in which a system was developed for bronchoscopic tracking for navigation. Again, the goals of the work are different. In lung biopsy, the goal is for long-range navigation to successfully follow the bronchial tree to a target location, whereas the goal of our work is short-range re-registration for high-accuracy visualization of critical anatomy. Furthermore, our work has a number of clear differences, (i) the use of SVD-Match for correspondence initialization, (ii) the use of ASKC for motion estimation, (iii) the use of feature-based registration to CT [16] and (iv) the continued use of ASKC for tracking.

II. Method

There are five major components in our system. Fig. 2 shows an overview of the system with the five components highlighted in blue. First, the system extracts Scale Invariant Feature Transform (SIFT) features [17] from the video data. Next, the motion between the images is estimated, after which feature points are reconstructed. At the beginning of the video sequence, reconstructed points from the first pair of images are registered to an isosurface segmented from CT data. This initial registration is used to initialize a registration tracking algorithm that makes use of feature matches in subsequent frames. Each of these components is detailed in the following subsections.

A. Feature detection and matching

In the first step, video data are processed to extract image features using the Matlab (The Mathworks Inc., Natick, MA, USA) implementation [18] of SIFT features [17]. Previous work [19] has shown SIFT performs well on endoscopic video. We tested both the SIFT feature matching technique suggested by Lowe and the SVD SIFT matching technique [12] originally purposed for two-dimensional point features in [20]. SVD SIFT computes the normalized cross-correlation weighted by the distance between all the features. The SVD is applied to the weighted correlation matrix to normalize the feature match scores by setting the singular values to 1. This up-weights weak matches and down-weights strong matches in turn, increasing
Fig. 2: System Overview: First the endoscope images are summarized by extracting and matching features. The motion is estimated and geometry reconstructed, after which the CT is registered and the video tracked. Each of the boxes highlighted in light blue are described in sections II-A through II-E.

Fig. 3: Endoscope image of the sphenoid sinus. (a) the undistorted endoscope image from a cadaver, (b) the detected features in an endoscope image, (c) the initial correspondence from SVD SIFT with inliers (blue) and outliers (red) and (d) the result of applying the robust ASKC method to estimate the motion.

the number of matches at the possible cost of lowering the accuracy of the matches. A pair of features is considered a match if the pair is greater than 50% correlated and the features within 100 pixels of each other. As we previously demonstrated in [21], SVD SIFT provides a larger number of matches (generally in the hundreds), which increases the number of points in the reconstruction. Thus, we used SVD SIFT for initial matching. Fig. 3 shows the detected features (3b) and initial correspondence from SVD SIFT, which includes inliers (blue) and outliers (red) (3c).

B. Motion Estimation with ASKC

After the image features are detected and matched, the endoscope motion is estimated using the robust technique of Wang et al. [22]. To solve the motion estimation problem we consider the following setup. When a camera observes a 3D point X on a surface from two distinct positions, the point X will project to two image locations \( x_1 = (u_1, v_1)^T \) and \( x_2 = (u_2, v_2)^T \). Let \( R \) and \( t \) represent the motion between images. It is well known that the following condition holds [23]:

\[
x_2^T K_2^T s_k(t) R K_1^{-1} x_1 = 0
\]  

(1)

where \( K_1 \) and \( K_2 \) are the intrinsic camera matrices corresponding to the two images, \( s_k(t) \) is the skew matrix of the translation vector \( t \) and \( R \) is the rotation matrix. The essential matrix \( E = s_k(t) R \) encodes the motion information of the camera. Given the intrinsic camera matrices, the essential matrix \( E \) can be estimated using a nonlinear five-point algorithm [24]. The camera motion \( (R, t) \) can be recovered from \( E \) by the SVD approach [25], although the translation can only be estimated up to a scale factor. In what follows we use \( \chi \) to represent the estimated scaled translation vector and \( \tilde{\chi} = \chi \).

The scale can be recovered by registering the reconstructed 3D model to a pre-operative CT scan [10].

To robustly estimate the camera motion from \( n \) matches, we employ an Adaptive Scale Kernel Consensus (ASKC) estimator [13]:

\[
t, R = \arg \max_{t, R} \frac{1}{h_1} \sum_{i=1}^{n} \frac{1}{h_1} \frac{r_i}{h_i}
\]  

(2)

where \( ker(\cdot) \) is a kernel function. The kernel function we use in this work is the Gaussian kernel, which was previously shown in [13] and [22] to work well for this problem. The variables \( h_i \) and \( r_i \) are respectively the bandwidth estimate and the residual of the \( i \)th matches. Since ASKC is a variable-bandwidth technique, \( h_i \) is automatically estimated from the data. In this case \( r_i \) is the following:

\[
r_i = x_{1(i)}^T K_1 K_2 x_{1(i)} = x_{2(i)}^T K_2^T s_k(t) R K_1^{-1} x_{1(i)}
\]  

(3)

Fig. 3d shows an example where ASKC has correctly estimated both the epipolar geometry and the scale of inliers, as well
as selected the correct matches even though the percentage outliers was at times larger than 70%.

C. Reconstruction

Once motion estimation is complete, the motion information \((R, t)\) is used to reconstruct the 3D structure up to scale. The first pair of images is reconstructed with triangulation and subsequent pairs of images are reconstructed from the tracked image features. The image features are tracked using SVD SIFT feature matches by matching feature of pairs of images and building feature trajectories.

We used a calibrated camera and removed the optical distortion with [26]. Let \(X_i = [x_i y_i z_i 1]^T\) be a 3D point in the world coordinate system. The 3D point \(X_i\) is projected to an image point \(x_i^f = [u_i v_i 1]^T\) at viewing position, or frame, \(f\) by a 3x4 projection matrix \(P_f\).

\[
x_i^f = P_f X_i
\]

Let the first frame be at the origin of the world coordinate system, and we have:

\[
P_1 = K [I|0] \quad \text{and} \quad P_f = K [^1 R_f |^1 t_f],
\]

where \(^1 R_f\) and \(^1 t_f\) are respectively the rotation and the translation of the camera at the \(f\)th frame relative to those of the camera at the first frame. Note that the camera matrix \(K\) of the endoscope remains fixed throughout the sequence. At the beginning, the structure is initialized using two frames by triangulation [25]. These two frames are selected by using the method described in [27] or by manual selection. For a new frame \(f\), we relate it to its previous frame \(f - 1\). Assuming we have known \(P_f = K [^1 R_{f-1} |^1 t_{f-1}]\) at the frame \(f - 1\), \(P_f\) can be written as:

\[
P_f = K [^1 R_{f-1} ^{f-1} R_f | ^{f-1} t_f + \lambda_f ^{f-1} t_{f-1}].
\]

Let:

\[
C_f = ^{f-1} R_f ^{-1} t_f, \quad K = [k_1 k_2 k_3]^T,
\]

and

\[
P_f = \begin{bmatrix}
p_1^f & p_2^f & p_3^f & p_4^f \\
p_2^f & p_3^f & p_4^f & p_5^f \\
p_3^f & p_4^f & p_5^f & p_6^f \\
p_4^f & p_5^f & p_6^f & p_7^f 
\end{bmatrix}
\]

From equations (4), (6), (7) and (8), we derive:

\[
u_i^f = \frac{p_{11} \lambda_f x_i + p_{21} \lambda_f y_i + p_{31} ^{f-1} \lambda_f z_i + k_1 C_f + \lambda_f k_1 ^{f-1} t_{f-1}}{p_{31} x_i + p_{22} y_i + p_{33} ^{f-1} z_i + k_3 C_f + \lambda_f k_3 ^{f-1} t_{f-1}}
\]

\[
v_i^f = \frac{p_{21} \lambda_f x_i + p_{22} \lambda_f y_i + p_{23} ^{f-1} z_i + k_2 C_f + \lambda_f k_2 ^{f-1} t_{f-1}}{p_{31} x_i + p_{22} y_i + p_{33} ^{f-1} z_i + k_3 C_f + \lambda_f k_3 ^{f-1} t_{f-1}}.
\]

If we define the following:

\[
A_f = \begin{bmatrix}
u_i^f k_3 ^{f-1} t_{f-1} - k_1 ^{f-1} t_{f-1} \\
u_i^f k_1 ^{f-1} t_{f-1} - k_2 ^{f-1} t_{f-1}
\end{bmatrix},
\]

and

\[
B_f = \begin{bmatrix}
p_{11} x_i + p_{12} y_i + p_{13} z_i + k_1 C_f - u_i^f p_{31} x_i \\
p_{21} x_i + p_{22} y_i + p_{23} z_i + k_2 C_f - v_i^f p_{31} x_i \\
p_{31} x_i + p_{22} y_i + p_{33} z_i + k_3 C_f + u_i^f p_{21} x_i \\
p_{41} x_i + p_{42} y_i + p_{43} z_i + v_i^f p_{21} x_i
\end{bmatrix},
\]

we can calculate the scale value \(\lambda_f\) by:

\[
\lambda_f (i) = (A_f A_f)^{-1} A_f B_f.
\]

However, as both the feature’s location and the 3D points may be in error, we estimate \(\hat{\lambda}_f\) in a robust way:

\[
\hat{\lambda}_f = \arg\max_{\lambda_f (i)} \frac{1}{n} \sum_{j=1}^{n} \frac{1}{h_j} \text{ker} \left( \frac{r_j}{h_j} \right),
\]

where \(r_j = \sum |x_i^f - P_f (\lambda_f (i)) \tilde{X}_i|\) and \(h_j\) is estimated from the data with the robust \(k\) scale estimator [28]. After \(\hat{P}_j\) is estimated, the 3D points \(\{X_i\}\) having correspondences to the tracked SIFT features are refined:

\[
\tilde{X}_i = \arg\min_{\tilde{X}_i} \sum_{j=0}^{m-1} |x_i^{j-1} - \hat{P}_{j-1} \tilde{X}_i|.
\]

Newly appearing 3D points are initialized and added to the structure. Algorithm 1 gives an outline of the reconstruction algorithm.

**Algorithm 1** \(\left\{ \{X_i\}_{i=1,\ldots,m}, \{P_f\}_{j=1,\ldots,n} \right\} = \text{Reconstruction} \left( \{(Images_j)_{j=1,\ldots,n}\} \right)\)

1: for all \(Images_j\) do
2: Extract SIFT features using SIFT detector [17]
3: //Initialize the 3D structure.
4: Choose two initial frames and detect potential matches by the SVD SIFT algorithm [12].
5: Select the correct matches by ASKC (2) and calculate the motion parameters of the endoscopic camera.
6: Initialize the structure \(\{X_i\}\) by triangulation [25].
7: //Maintain the 3D structure.
8: Obtain matches between the frames \(f\) and \(f - 1\).
9: Track the SIFT features using the feature tracking algorithm proposed in [22].
10: Compute the projection matrix \(P_f\) by (5)
11: Compute the 3D points corresponding to the new SIFT features and add them to the 3D structure.
12: Refine the existing 3D points that correspond to the tracked SIFT features.
13: end for
14: Output the reconstructed 3D structure \(\{X_i\}_{i=1,\ldots,m}\) and the projection matrices \(\{P_f\}_{j=1,\ldots,n}\).

D. Registration

The reconstructed 3D point cloud is registered to a surface segmented from a CT image of the same patient. The surface is segmented by applying a threshold at the air/tissue boundary (approximately -500 Hounsfield Units) and post-processed using marching cubes [29] to create a polygon isosurface. We applied a registration algorithm described in [16] that is derived from Trimmed ICP (TriICP) [30] and extends TriICP with scale [31]. Since the true scale of the 3D world is lost in the epipolar constraint (1), the registration algorithm also needs to estimate the scale of the 3D model.
our registration algorithm requires three inputs: a 3D point cloud of relative scale with the origin at the camera center, an isosurface from the CT, and an initial estimate of the scale and location. The 3D point cloud is the expected output of the 3D reconstruction process. We assume that the 3D point cloud is of uniform scale that need not be the same as the CT, and that the origin of the point cloud is the camera center such that the rigid-body transformation aligning the 3D point cloud is the camera location in CT coordinates. Outliers typically contaminate the 3D reconstruction as a result of mismatched features and the ambiguity of sign in the epipolar constraint.

The second input, the isosurface from the CT, is essential because the algorithm registers 3D points to a surface. While using only a surface does remove a large amount of data, there is sufficient data in the surface alone for registration. Similar to ICP [32], TrICP is prone to local minima and an approximate solution is needed to start the algorithm, thus the third input, an initial estimate of the location and scale is required.

TrICP was chosen for its robustness to outliers and simple design. Our algorithm modifies the traditional TrICP algorithm by adding scale. This modification is similar to the work of Du et al. [31] However, we assume a single uniform scale. The following is the derivation of the modified TrICP algorithm.

Let $X \in \mathbb{R}^{3\times n}$ be the matrix of all the reconstructed points as column vectors, where $n$ is the number of points. Let $B \in \mathbb{R}^{3\times n}$ be the matrix of the corresponding closest points on the model as column vectors. We then compute:

$$X_{\text{zero}} = X - \text{mean}(X), \quad B_{\text{zero}} = B - \text{mean}(B),$$

$$C_X = \frac{1}{n} X_{\text{zero}} X_T^{\text{zero}}, \quad C_B = \frac{1}{n} B_{\text{zero}} B_T^{\text{zero}}.$$  

(15)

(16)

Following Burschka et al. [10], we set up the following equation to solve for the scale. Let $\lambda_X = [\lambda_{X_1}, \lambda_{X_2}, \lambda_{X_3}]^T$ and $\lambda_B = [\lambda_{B_1}, \lambda_{B_2}, \lambda_{B_3}]^T$, vectors of the eigenvalues of $C_X$ and $C_B$, respectively. We can interpret the eigenvalues of each data set as the diagonalized covariance matrix of the points. As a result, under perfect conditions, the following relationship holds between the data sets:

$$s^2 \lambda_X = \lambda_B.$$  

(17)

Thus, we can estimate the unknown scale factor as the scale of the vector projection of $\lambda_B$ on to $\lambda_X$:

$$s = \sqrt{\frac{\lambda_X \cdot \lambda_B}{\lambda_X \cdot \lambda_X}}.$$  

(18)

The resulting modified pose computation is Algorithm 2.

The registration algorithm inputs are processed as follows: the isosurface from CT, represented as a polygonal mesh, is loaded into the renderer. Then a rendering is created at the initial camera location. The visible polygons are subsequently fed into a KD-tree for TrICP. TrICP then solves for the rigid-body transformations and scale. The new camera location is then fed back to the renderer and the process continues to convergence. Algorithm 3 shows a pseudo-code for the complete registration process. The algorithm uses the following variables: $\text{mesh}$ is the model polygon mesh, $\{X_i\}_{i=1,\ldots,m}$ is the set of 3D reconstructed points and $R, t$ and $s$ are the rotation, translation and scale between the mesh and 3D points, respectively.

Algorithm 2 \((R, t, s) = \text{registerPointSet}(X, B)\)

1: \(X_{\text{zero}} \leftarrow X - \text{mean}(X), \quad B_{\text{zero}} \leftarrow B - \text{mean}(B)\)
2: \(C_X \leftarrow \frac{1}{n} X_{\text{zero}} X_T^{\text{zero}}, \quad C_B \leftarrow \frac{1}{n} B_{\text{zero}} B_T^{\text{zero}}\)
3: \(s \leftarrow \sqrt{\frac{\lambda_X \cdot \lambda_B}{\lambda_X \cdot \lambda_X}}\)
4: \(H \leftarrow \frac{1}{n} X_{\text{zero}} B_{\text{zero}}\)
5: \(\text{inliers} \leftarrow \text{TrICPbuffer}(R_{\text{init}}, t_{\text{init}}, s_{\text{init}}, \text{mesh}, \{X_i\}_{i=1,\ldots,m})\)
6: \(R \leftarrow X_{\text{zero}} B_{\text{zero}}\)
7: \(\text{while} \ \text{Not Converged} \ \text{do}\)
8: \(\text{if} \ \text{det}(R) = -1 \ \text{then}\)
9: \(V \leftarrow \{v_1, \ v_2, -v_3\}\)
10: \(\text{end if}\)
11: \(X_{\text{zero}} \leftarrow X_{\text{zero}} - s \cdot R \cdot s \cdot \text{mean}(X)\)
12: \(X \leftarrow R_{\text{init}} \cdot t_{\text{init}}, \quad t \leftarrow t_{\text{init}}, \quad s \leftarrow s_{\text{init}}\)
13: \(R \leftarrow R_{\text{init}}, \quad t \leftarrow t_{\text{init}}, \quad s \leftarrow s_{\text{init}}\)
14: \(\text{end while}\)

The z-buffer algorithm [33] shown in algorithm 3 efficiently determines the visible polygons that are the exact surface the reconstruction needs to be registered. The use of the z-buffer algorithm allows the entire sinus model to be loaded only once and enables registration to any section of the sinus.

Algorithm 3 \((R, t, s, \text{inliers}) = \text{TrICPbuffer}(R_{\text{init}}, t_{\text{init}}, s_{\text{init}}, \text{mesh}, \{X_i\}_{i=1,\ldots,m})\)

1: \(R \leftarrow R_{\text{init}}, \quad t \leftarrow t_{\text{init}}, \quad s \leftarrow s_{\text{init}}\)
2: \(\text{while} \ \text{Not Converged} \ \text{do}\)
3: \(\text{visible\_mesh} = \text{render}(\text{mesh}, R, t)\)
4: \(\text{Create kD\_tree(\text{visible\_mesh})}\)
5: \(\text{for all} \ X_i \in \{X_i\} \ \text{do}\)
6: \(X_i \leftarrow R \cdot s \cdot X_i + t\)
7: \(b_i \leftarrow \text{kD\_tree\_closest\_point}(X_i)\)
8: \(\text{error}_i \leftarrow \|X_i - b_i\|\)
9: \(\text{end for}\)
10: \(\text{sorted\_error} = \text{sort(error)}_{\text{floor}((s+5)/5)}\)
11: \(\text{inliers} \leftarrow \forall i \ s.t. \ i < \text{error}_i \cdot m\)
12: \(X_i \leftarrow R \cdot s \cdot X_i + t\)
13: \(\text{end while}\)

E. Tracking

After the initial 3D-3D registration, the 2D-3D correspondence between image features and the 3D surface of the CT data is established. Now, a more efficient 2D-3D pose estimator can be used to update the camera pose. Here we combine the robust sampling method of [22] with the pose algorithm of [34] to create a robust 2D-3D pose estimation method. In Algorithm 4, we present an overview of the complete tracking system. $R_{\text{init}}, t_{\text{init}}, s_{\text{init}}$ are the initial rotation, translation and scale respectively. $\{X_i\}_{i \in \text{inliers}}$ is the set of 3D reconstructed points which are inliers after the initial registration. $\{\text{Images}_j\}_{j=1,\ldots,n}$ is a set of images from the video. $\text{mesh}$ is the surface mesh segmented from the CT data.
Algorithm 4 \((\{R_j\}_{j=1,...,n},\{t_j\}_{j=1,...,m}) = \text{Camera Pose Tracker}\) (\(R_{\text{init}}, t_{\text{init}}, s_{\text{init}}; \{X_i\}_{i=1,...,m}, \{\text{Images}_j\}_{j=1,...,n}, \text{mesh}\))

1: \(R \leftarrow R_{\text{init}}, \ t \leftarrow t_{\text{init}}, \ X_i^{\text{current}} \leftarrow s_{\text{init}}X_i\)
2: for \(j \leftarrow 2\) to \(n\) do
3: \(\text{if } j = 2 \text{ then}\)
4: \(\text{Image}_1 \leftarrow \text{undistort}(\text{Images}_1)\)
5: \(s_{\text{init}} \leftarrow \text{detect SIFT feature}(\text{Image}_1)\)
6: \(\text{else}\)
7: \(\hat{\text{Image}}_1 \leftarrow \hat{\text{Image}}_2\)
8: \(s_{\text{init}} \leftarrow s_{\text{init}}\)
9: end if
10: \(\text{Image}_2 \leftarrow \text{undistort}(\text{Images}_j)\)
11: \(s_{\text{init}} \leftarrow \text{detect SIFT feature}(\text{Image}_j)\)
12: \(\text{matches} = \text{SVDSIFT match}(s_{\text{init}}, s_{\text{init}})[12]\)
13: \((E, \text{inliers}) = \text{robustMotionEstimator}(\text{matches})\) \(2\)
14: //Track the old sift feature and add new matches.
15: \(\{X_i^{\text{current}}\} = \text{tracker}(\text{matches, inliers, } \{X_i^{\text{current}}\})\) \(22\)
16: \((\hat{R}, \hat{t}) = \text{robustPoseEstimator}(X_i^{\text{current}}, \text{matches, } R, t)\) \(2\)
17: \(\{X_i^{\text{reprojectedInliers}}\} = \text{reprojectPoints}(\{X_i^{\text{current}}\}, \hat{R}, \hat{t})\)
18: \((R, t) = \text{robustPoseEstimator}(\{X_i^{\text{reprojectedInliers}}\}, \text{matches, } R, t)\) \(2\)
19: //Remove outliers from refined points
20: \(\{X_i^{\text{refined}}\} = \text{refinePoints}(\{X_i^{\text{reprojectedInliers}}\}, \text{mesh})\) \(19\)
21: for all previous \(R_i, t_i\) do
22: \((R_i, t_i) = \text{robustPoseEstimator}(X_i^{\text{refined}}, \text{matches, } R_i, t_i)\) \(2\)
23: end for
24: end for

The system first undistorts the images, then SIFT features are detected and matched. After finding the set of matched image features, these image feature points are used to estimate the motion of the frame pair. The inliers of the motion estimation are then tracked using the SIFT feature tracking method from [22]. Once the new camera pose is estimated, we refine both the 3D points and pose estimates. The 3D points are refined by applying all of the previously seen image features and projection matrices and then solving the following null space problem. First, given a simple projection model from (4), let \(R_i = [\hat{r}_{1,i}^T \, \hat{r}_{2,i}^T \, \hat{r}_{3,i}^T]^T\) and \(t_i = [t_{1,i} \, t_{2,i} \, t_{3,i}]^T\). The 3D point which was created by all of the imaged points is the null space solution of

\[
0 = \begin{bmatrix}
\hat{r}_{1,i}^T & t_{1,i} - u_1 \hat{r}_{3,i}^T & t_{3,i} \\
\hat{r}_{2,i}^T & t_{2,i} - v_1 \hat{r}_{3,i}^T & t_{3,i} \\
\vdots & \vdots & \vdots
\end{bmatrix}
\begin{bmatrix}
x_1 \\
y_1 \\
z_1
\end{bmatrix}.
\tag{19}
\]

III. IMPLEMENTATION AND EXPERIMENTS

The algorithm was prototyped in Matlab. After the initial evaluation previously reported in [22], [16], and [21], we translated some portions of the code into C++ and merged the code into the TREK image-guided surgery software platform [35]. The Matlab code was directly merged into Python [36] for use with TREK via mlabwrap [37]. The TREK platform enabled easy access to the visualization components of 3D Slicer [38] and the processing components of the cist library [39].

A. Experimental Setup

For our experiments we used both a skull phantom and cadaveric human head. Both were set up similarly.

1) Phantom: A skull phantom was used to perform the experiments in a dry lab. A Sawbones (Pacific Research Laboratories, Vashon, Washington, USA) skull phantom was prepared with #4 and #6 aluminum sheet metal screws. Aluminum, having a low atomic number, produces minimal artifacts in CT scans. Twelve of the larger #6 screws were placed in the front and sides of the cranium as shown in Fig. 4. The smaller #4 screws were placed approximately where the sphenoid sinus would be. Though the phantom did not have a sphenoid sinus, it enabled us to mock up the geometry of the entire skull and nasal passage. The #6 screws were used to register the skull to the navigation system during the video recording. The skull phantom was mounted to a stand and clamped to the work bench. An Optotrak rigid-body was also attached to the work bench. The skull phantom was finally registered to the navigation system using a ball pointer tool.

2) Cadaver: The protocol for the phantom study was then transferred to the wet lab in our cadaver study. Similarly, ten #6 screws were implanted in the front and sides of the cranium of the cadaver head. After a craniotomy was performed, 27 gauge syringe needles were placed in the sphenoid sinus wall, the pituitary gland, carotid arteries and optic nerves, shown in Fig. 5. After implanting the needles into the bone or tissue, Great Stuff (Dow Chemical Company, Midland, Michigan, USA) was used to secure and support the needles. Each of the needles were used as a target structure or critical structure that requires high-precision navigation to identify. Again the #6 screws were used to register the navigation system to the video during recording. The cadaver head was secured in a Mayfield clamp. We performed the cadaver experiment in two different specimens. In our first study, we examined navigation error (defined below in section III-D) versus TRE. The second study was used to compare pointer-based, tracker-based and video-based navigation. Table I shows a summary of the different methods.

<table>
<thead>
<tr>
<th>Method Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pointer-based</td>
<td>Targets are identified and localized with a pointer tool.</td>
</tr>
<tr>
<td>Tracker-based</td>
<td>Targets are identified and visualized in the video registered to the CT via a tracking system</td>
</tr>
<tr>
<td>Video-based (Our Method)</td>
<td>Targets are identified and visualized in the video registered to the CT via video reconstruction and registration</td>
</tr>
</tbody>
</table>

TABLE I: A summary of three methods that will be compared in our experiments.
Fig. 4: Distribution of screws (a,b) and CT slices with an endoscope image (c) of the skull phantom.

Fig. 5: Cadaver: Shown in (a) and (b) are the distribution of screws and needles with Great Stuff in the cadaver. (c) shows CT slices and an endoscope image.

Fig. 6: Shown in (a) is the configuration of Optotrak rigid-bodies on the camera and endoscope. Presented in (b) is an overview of the experimental system setup, showing each of the tracking systems (Optotrak and StealthStation) used and their relationship to the phantom.
B. Data Collection

We collected endoscopic phantom sinus and ex-vivo human sinus video data. The video was captured at $1024 \times 768$, 30 hz using a Flea2 (Point Grey Research, Richmond, British Columbia, Canada) firewire camera attached to a zero-degree rigid monocular Storz (Karl Storz, Tuttlingen, Germany) endoscope. Optotrak rigid bodies were attached to the endoscope and specimen, shown in Fig. 6. The Optotrak motion data were used to initialize the endoscopic motion by transforming the motion of the rigid body attached to the endoscope to the camera center by way of the hand-eye calibration. The Optotrak and Flea2 were synchronized via the triggering pin on the Optotrak. In this way the clock signal the camera emits was the clock for the Optotrak. To verify that the video and tracker motion were synchronized, video of a pure translation while viewing a fixed point was recorded. Before the data collection, images of a checkerboard calibration grid were also recorded using the endoscope. We performed an offline camera and hand-eye calibration of the endoscope using the German Aerospace Center (DLR) Camera Calibration Toolbox [40]. The CT data used had 0.46 mm $\times$ 0.46 mm $\times$ 0.5 mm voxels.

C. Sources of Error

Our data collection had four potential sources of error: 1) the camera calibration; 2) the Optotrak to camera calibration; 3) the Optotrak to CT registration; and 4) unaccounted for motion. The camera was calibrated within an error of [0.38, 0.37] pixels. The aforementioned Optotrak/camera configuration had an estimated position uncertainty in millimeters of the camera center as measured by the Optotrak of [1.1, 0.4, 0.1]. The Optotrak to CT registration was generally within .5 mm RMS point distance error. Each of these contributes to an overall location uncertainty of approximately 1.5 mm in the absolute position of the endoscope optical center, and approximately 1.1 mm relative position accuracy.

D. Evaluation Methodology

We compared our method to both pointer-based navigation and tracker-based endoscopy using three error metrics. The metrics we considered were two target registration errors and tracker-based endoscopy using three error metrics. The error metrics we applied in our evaluation.

We define $TRE_1$ as described by Fitzpatrick and West [41] for the traditional pointer-based approach.

$$TRE_1 = \| p_{CT} - (CTT_{Navigation}) p_{pointer} \|$$  \hspace{1cm} (20)

Where $p_{CT}$ is the target segmented from the CT, $CTT_{Navigation}$ is the transformation from the tracking system to the CT as computed with the fiducial points and $p_{pointer}$ is the current pointer location.

For both tracker-based endoscopy and video-based registration, we defined $TRE_2$ as described in Mirota et al. [42]

$$r = RK^{-1}p_{image} - t,$$  \hspace{1cm} (21)

where $R$, $K$ and $t$ are as defined in (5) and $p_{image}$ is the point segmented in the video image. Equation (21) defines a ray from the camera center ($t$) in the direction of the target in the image. If the camera was exactly registered, the ray would pass through the center of the target in the CT. We adopt this metric to be able to directly compare tracker-based endoscopy with our video-based registration, because only camera and image data are available in tracker-based endoscopy. The ray ($r$) is then projected to the closest point to the target ($p_{CT}$) which is used to compute the TRE as follows:

$$TRE_2 = \| p_{CT} - (t + r \left( \frac{r \cdot (p_{CT} - t)}{r \cdot r} \right)) \|.$$  \hspace{1cm} (22)

Another way to think about (22) is that it creates a virtual pointer along the ray from the endoscope with end point equal to the closest point to the target along the ray. The major difference in the two TREs are that in the first case 3D point data are compared directly, whereas in the second case a camera location and ray are used to measure the error in the camera image.

We defined $NGE$ as the relative distance between a target segmented in the CT data and the location the navigation system reported as the location of the target. Mathematically, $NGE$ is the same as (22). $NGE$ is distinct from $TRE$ in that the targets are not visible in the endoscope and only the surgeon’s perception and the navigation system guide the location of the target. In both TREs the targets are clearly visible in both the endoscope and the navigation system. Table II summarizes the error metrics we applied in our evaluation.

For context, we measure traditional pointer-based $TRE$ and tracker-based endoscopy $TRE$ to show the differences between our method and that which is currently available in the operating room as well as a method similar to that of the state of the art in tracker-based endoscopy [8]. For the tracker-based results the Optotrak was used to record camera motion. Additionally, we initialize the camera location in the video-based method with the pose of the camera measured by the Optotrak. The StealthStation could be used instead of the Optotrak to provide an initialization. To avoid single sample basis we took 10 measurements and recorded the error for each target that was visible. We then averaged over the samples of a target to produce a measurement at a single target.

1) Statistical Evaluation of $TRE$: For further statistical evaluation of TRE, we assume that the three orthogonal components of both $TRE_1$ and $TRE_2$ are independent, normally disturbed variables with zero mean. The expectation of $TRE^2$ will be of the form:

$$\langle TRE^2 \rangle = \sigma_1^2 + \sigma_2^2 + \sigma_3^2,$$  \hspace{1cm} (23)

where $\sigma_1^2$, $\sigma_2^2$, and $\sigma_3^2$ represent the variances of the three component random variables. Our estimate of $\langle TRE^2 \rangle$ is

<table>
<thead>
<tr>
<th>Metric Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$TRE_1$</td>
<td>Metric for evaluating pointer-based methods</td>
</tr>
<tr>
<td>$TRE_2$</td>
<td>Metric for evaluating tracker-based and video-based methods</td>
</tr>
<tr>
<td>$NGE$</td>
<td>Same as $TRE_2$, however, the target is not visible in the endoscope image.</td>
</tr>
</tbody>
</table>
therefore a measure of total variance, so we test the null hypothesis that the expected mean $TRE^2$ values are equal to elucidate the relative performance of our navigation systems. Since $TRE^2$ represents the sum of three chi-squared variables, its distribution is not normal. We use the zero-skewness log transform to normality to account for this. We used a mixed linear model for data analysis. For the cadaver head study, we account for three major sources of variation: i) a head term which accounts for experimental variability caused by differences between heads; ii) a pin term which accounts for experimental variability due to differences between pins; this accounts for differences between target registration error between pins due to the relationship between target registration error and fiducial registration error and variability between the surgeons’ ability to touch pins in the nose due to anatomic constraints of the cadaver specimens; and iii) residual error. Residual error will be a function of the scatter in data caused by the collection method, i.e., test subjects touching pins with a pointer, in addition to the traditional sources of error attributed to regression analysis. The pin term is nested within the heads term. To this random effects model we added three fixed covariates to elucidate the effects of navigation methods on target registration error. We assumed the Optotrak represented the gold standard and use it as our reference.

We used a similar regression model to analyze the phantom data. Since we only performed one phantom study, we dropped the head term from the phantom data analysis. To compare the navigation systems, we calculate the marginal means treating the factor variables as balanced and then perform post hoc Wald tests corrected for multiple comparisons using Bonferroni’s method. The post hoc tests perform hypothesis tests on the differences in transformed squared errors between methods. For cadaver head one we also compared $TRE^2$ to $NGE^2$ for the tracker based and video-based navigation systems. We used a two-factor analysis of variance (ANOVA) for this comparison. The squared errors served as the dependent variables. Two factors (navigation method and error type) with two levels each (tracker-based and video-based; $TRE^2$ and $NGE^2$) were used as dependent variables. The two factors were allowed to interact. We set the global type I error at 0.05. Since we compared $TRE^2$ to each other and $TRE^2$ to $NGE^2$ for each for the cadaver study we use an alpha of 0.025 for each test.

2) Evaluation of Tracking Accuracy: Since we previously established that the absolute position of the Optotrak is potentially not as accurate as video tracking, we verify our method has similar relative trajectory accuracy by evaluating relative motion difference.

$$F_{\text{difference}} = \begin{bmatrix} R_{O_i} & t_{O_i} \\ 0^T & 1 \end{bmatrix}^{-1} \begin{bmatrix} R_i & t_i \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} R_{O_i} & t_{O_i} \\ 0^T & 1 \end{bmatrix}^{-1}$$

(24)

The variable $i$ is from 1 to the number of images of the sequence. Using the cadaver data, we evaluated 10 arbitrary segments of video consisting of 67 images each.

3) Registration Initialization Sensitivity: Since we used the tracker to initialize the registration process, we evaluated the effects of errors in the tracker data. We introduced errors into the tracker by adding an in-plane or out-of-plane perturbation vector with varying magnitude from 0 to 5 mm. The in-plane motion was defined by the estimated pose of the imaging plane and out-of-plane motion is any motion perpendicular to the imaging plane.

4) Evaluation of Shape Accuracy: To evaluate the accuracy of the shapes from the reconstruction process we examined the mean distances to the CT surface.

IV. RESULTS

The following is a discussion of the TRE measured in each of the experiments. In all of the following box plots the rank statistics from top to bottom are maximum, third quartile, median, first quartile and minimum.

A. Synthetic Data

1) Tracking: We focused our evaluation on tracking. We evaluated the robustness of the method to noise and outliers by creating projected 2D points of a known 3D model, then adding uniform noise or uniform outliers to the projected 2D points. We assumed the registration to the model was perfect to isolate the analysis to only tracking. Fig. 7 shows the error of tracking in each frame of the 10 frames used in the simulation. Translation error is the norm of the translation component of the motion difference. Rotation error is norm of the Euler angles of the rotation component of the motion difference. The tests on synthetic data showed that the tracking method was robust to large outliers and was very robust to noise. It is clear that noise has little effect on performance. Since the points were sampled with ASKC, it quickly rejects both outliers and points with large noise. Additionally, it can be seen that the tracker drifts in the presence of high noise (greater than 60 pixels) or high percentage of outliers (greater than 40%).

B. Phantom Study

Fig. 8 summarizes the overall distribution of all TRE measurements within the rigid Sawbone phantom. The Optotrak pointer and Medtronic StealthStation pointer provide context for the tracker-based and video-based methods. The Optotrak pointer, the gold standard in tool tracking, showed the highest accuracy and precision, as expected. The mean TRE of the Optotrak information was 0.51 mm (with 0.44 mm first quartile and 0.38 mm range), while the StealthStation mean TRE was 2.56 mm (with 1.73 mm first quartile and 2.16 mm range). The mean TRE for the tracker-based method was 1.30 mm (with 1.28 mm first quartile and 1.49 mm range). By comparison, the mean TRE for the video-based method was 0.98 mm (with 0.83 mm first quartile and 0.98 mm range). The improvement in TRE over the tracker-based method was statistically significant ($p < 0.01$), with the majority of all targets having sub-millimeter precision.

C. Cadaver Studies

In the first cadaver study, the navigation error and TRE were measured. Fig. 9 shows the distribution of both metrics in the study. The navigation error was greater than the TRE, which was expected because the needle tips are hidden from
Fig. 7: Results from our tracking method with synthetic data in which the effects of outliers and noise on tracking performance were characterized. (a) Effect of noise on the translation component of the motion. (b) Effect of noise on the rotation component. The effects of outliers and nominal noise on translation and rotation are shown in (c) and (d).

Fig. 8: Overall distributions of TRE for the phantom study. Our method is labeled video-based. The rank statistics from top to bottom are the, third quartile, median, first quartile and the minimum.

view and not protruding into the sinus. In the navigation error case, the tracker-based registration method mean error was 2.36 mm (with 1.71 mm first quartile and 3.09 mm range), while the video-based registration had a mean error of 1.54 mm (with 1.13 mm first quartile and 1.87 mm range). When looking at TRE, the video-based method (with a mean of 1.25 mm, 0.96 mm first quartile and 1.51 mm range) outperformed the tracker-based method (with a mean of 1.43 mm, 1.28 mm first quartile and 0.77 mm range). The results demonstrate that video-based registration is moderately more accurate than tracker-based methods.

Shown in Fig. 10 is the overall distribution of TRE measured in the cadaver. Again, for context, the Optotrak pointer and Medtronic StealthStation pointer are shown for the targets. The Optotrak had a mean TRE of 1.25 mm (with 1.14 mm first quartile and 0.93 mm range). The StealthStation was closer to the Optotrak in the experiment with a mean TRE of 1.49 mm (with 1.28 mm first quartile and 1.43 mm range). For the tracker-based registration method, the TRE was 1.01 mm (with 0.93 mm first quartile and 0.72 mm range). By comparison, the mean TRE for the video-based method was 0.70 mm (with 0.56 mm first quartile and 0.50 mm range). The results again demonstrate a significant improvement ($p < 0.001$) for the video-based registration approach with all but one target with sub-millimeter precision.

In Fig. 11 we compare the overall distribution of TRE measured in both cadaver studies. Both studies exhibit a general trend of improvement, although there is a clear difference in the range of the video-based method between the two cadaver studies. In the first study the maximum of the video-based method (max 2.16 mm) was greater than the tracker-based method (max 1.88 mm). The difference was caused by two samples of the target with maximum error in which the scale of the 3D point cloud was not estimated correctly because of motion blur in the original images. We prevented motion blur in the second cadaver study by adjusting the gain, exposure and shutter settings on the camera.

Finally, Fig. 12 presents an example registration result showing how the video-based method brings the targets into closer alignment than the tracker-based method.
(a) Tracker-based TRE
(b) Tracker-based TRE Colormap
(c) Video-based TRE
(d) Video-based TRE Colormap

Fig. 12: Example registration result shows the TRE in the video. In (a-d) needle targets (placement of which is shown in Fig. 3a) in the video are labeled with solid green circles and the respective targets in the CT are labeled with solid blue circles.

D. Statistical Evaluation

In the following tables CI stands for Confidence Interval. The \( k \) parameter for the log skew normality transformation for the \( \text{TRE}^2 \) cadaver head data was \(-0.109\). The regression results are displayed in Table III and Table IV. The marginal means as a function of navigation method are shown in Table V. We note that the Optotrak had the largest marginal mean and therefore the highest total variance while the video-based navigation had the lowest overall marginal mean and therefore the smallest total variance. Post hoc testing (Table VI) shows that the difference in total variance between the Optotrak and StealthStation was not statistically significant; all other comparisons were statistically significant, suggesting that tracker-based navigation performs better than traditional navigation systems and that video-based navigation performs best.

**TABLE III: Head Study Fixed Effects Results.**

<table>
<thead>
<tr>
<th>Covariate</th>
<th>Regression Coefficient</th>
<th>Standard Error</th>
<th>p-value</th>
<th>97.5% CI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stealth-Station</td>
<td>-0.23</td>
<td>0.51</td>
<td>0.66</td>
<td>(-1.369, 0.916)</td>
</tr>
<tr>
<td>Tracker-Based</td>
<td>-1.03</td>
<td>0.42</td>
<td>0.01</td>
<td>(-1.945, -1.115)</td>
</tr>
<tr>
<td>Video-Based</td>
<td>-1.56</td>
<td>0.40</td>
<td>&lt;0.001</td>
<td>(-2.475, -0.645)</td>
</tr>
<tr>
<td>Constant</td>
<td>1.32</td>
<td>0.57</td>
<td>0.02</td>
<td>(0.487, 2.596)</td>
</tr>
</tbody>
</table>

**TABLE IV: Head Study Random Effects Results.**

<table>
<thead>
<tr>
<th>Random Effects Parameters</th>
<th>Estimate</th>
<th>Standard Error</th>
<th>97.5% CI</th>
</tr>
</thead>
<tbody>
<tr>
<td>SD: Head Term</td>
<td>0.68</td>
<td>0.40</td>
<td>(0.132, 3.471)</td>
</tr>
<tr>
<td>SD: Pin Term</td>
<td>0.31</td>
<td>0.07</td>
<td>(0.179, 0.526)</td>
</tr>
<tr>
<td>SD: Residual</td>
<td>0.76</td>
<td>0.03</td>
<td>(0.703, 0.828)</td>
</tr>
</tbody>
</table>

**TABLE V: Head Study Marginal Means.**

<table>
<thead>
<tr>
<th>Navigation</th>
<th>Marginal Means</th>
<th>Standard Error</th>
<th>p-value</th>
<th>97.5 % CI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optotrak</td>
<td>1.13</td>
<td>0.49</td>
<td>0.02</td>
<td>(0.033, 2.231)</td>
</tr>
<tr>
<td>Stealth-Station</td>
<td>1.11</td>
<td>0.49</td>
<td>0.02</td>
<td>(0.004, 2.216)</td>
</tr>
<tr>
<td>Tracker-Based</td>
<td>0.29</td>
<td>0.49</td>
<td>0.55</td>
<td>(-0.808, 1.389)</td>
</tr>
<tr>
<td>Video-Based</td>
<td>-0.25</td>
<td>0.49</td>
<td>0.6</td>
<td>(-1.352, 0.844)</td>
</tr>
</tbody>
</table>
The $k$ parameter for the log skew normality transformation for the $NGE^2$ was $-0.345$. Both the navigation method term ($F(1, 154) = 9.16; p < 0.001$) and the error term ($F(1, 154) = 9.16; p = 0.003$) were statistically significant. The interaction term was not significant. Post hoc testing shows that all of the ANOVA contrast comes from the fact that the tracker-based $NGE^2$ is significantly larger than all other errors.

The $k$ parameter for the log skew transformation to normality for the $TRE^2$ phantom data was $-0.068$. The regression results are shown in Tables VII and VIII. The marginal means are shown in Table IX. In the phantom study the Optotrak had the lowest total variance and the Stealth station had the greatest total variance. The video-based navigation and tracker-based navigation were in between with the video-based navigation’s variance being smaller than the tracker based navigation. Post hoc testing shows that all differences in total variance were statistically significant (see Table X).

### TABLE VII: Phantom Study Fixed Effects Results.

<table>
<thead>
<tr>
<th>Covariate</th>
<th>Regression Coefficient</th>
<th>Standard Error</th>
<th>p-value</th>
<th>95% CI</th>
</tr>
</thead>
<tbody>
<tr>
<td>StealthStation</td>
<td>2.47</td>
<td>0.18</td>
<td>&lt; 0.001</td>
<td>(2.108, 2.823)</td>
</tr>
<tr>
<td>Tracker-Based</td>
<td>1.73</td>
<td>0.17</td>
<td>&lt; 0.001</td>
<td>(1.389, 2.073)</td>
</tr>
<tr>
<td>Video-Based</td>
<td>0.91</td>
<td>0.17</td>
<td>&lt; 0.001</td>
<td>(0.566, 1.249)</td>
</tr>
<tr>
<td>Constant</td>
<td>-1.15</td>
<td>0.13</td>
<td>&lt; 0.001</td>
<td>(-1.405, -0.898)</td>
</tr>
</tbody>
</table>

### TABLE VIII: Phantom Study Random Effects Results.

<table>
<thead>
<tr>
<th>Random Effects Parameters</th>
<th>Estimate</th>
<th>Standard Error</th>
<th>p-value</th>
<th>95% CI</th>
</tr>
</thead>
<tbody>
<tr>
<td>SD: Pin Term</td>
<td>&lt; 0.001</td>
<td>&lt; 0.001</td>
<td></td>
<td>(0, 0.000004)</td>
</tr>
<tr>
<td>SD: Residual</td>
<td>1</td>
<td>0.04</td>
<td></td>
<td>(0.905, 1.101)</td>
</tr>
</tbody>
</table>

### TABLE IX: Phantom Study Marginal Means.

<table>
<thead>
<tr>
<th>Navigation</th>
<th>Marginal Means</th>
<th>Standard Error</th>
<th>p-value</th>
<th>95% CI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optotrak</td>
<td>-1.15</td>
<td>0.13</td>
<td>&lt; 0.001</td>
<td>(-1.405, -0.898)</td>
</tr>
<tr>
<td>StealthStation</td>
<td>1.31</td>
<td>0.13</td>
<td>&lt; 0.001</td>
<td>(1.062, 1.567)</td>
</tr>
<tr>
<td>Tracker-Based</td>
<td>0.58</td>
<td>0.12</td>
<td>&lt; 0.001</td>
<td>(0.350, 0.811)</td>
</tr>
<tr>
<td>Video-Based</td>
<td>-0.24</td>
<td>0.12</td>
<td>0.04</td>
<td>(-0.474, -0.013)</td>
</tr>
</tbody>
</table>

### TABLE X: Phantom Study Hypothesis Testing

<table>
<thead>
<tr>
<th>Covariate</th>
<th>Trans. Difference (mm)</th>
<th>Rot. Difference (deg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segment 1</td>
<td>1.54 (1.04)</td>
<td>0.05 (0.03)</td>
</tr>
<tr>
<td>Segment 2</td>
<td>1.68 (0.80)</td>
<td>0.04 (0.02)</td>
</tr>
<tr>
<td>Segment 3</td>
<td>0.78 (0.30)</td>
<td>0.06 (0.03)</td>
</tr>
<tr>
<td>Segment 4</td>
<td>0.93 (0.81)</td>
<td>0.03 (0.03)</td>
</tr>
<tr>
<td>Segment 5</td>
<td>0.90 (0.54)</td>
<td>0.03 (0.01)</td>
</tr>
<tr>
<td>Segment 6</td>
<td>0.87 (0.59)</td>
<td>0.03 (0.02)</td>
</tr>
<tr>
<td>Segment 7</td>
<td>0.66 (0.28)</td>
<td>0.03 (0.01)</td>
</tr>
<tr>
<td>Segment 8</td>
<td>2.70 (1.23)</td>
<td>0.19 (0.10)</td>
</tr>
<tr>
<td>Segment 9</td>
<td>0.73 (0.14)</td>
<td>0.03 (0.06)</td>
</tr>
<tr>
<td>Segment 10</td>
<td>0.70 (0.30)</td>
<td>0.04 (0.01)</td>
</tr>
<tr>
<td>Mean Difference</td>
<td>1.15</td>
<td>0.05</td>
</tr>
</tbody>
</table>

### Fig. 13: An example tracked trajectory with Optotrak in blue and our video-based method in red.

**E. Tracking with Cadaver Data**

Table XI shows the relative motion difference between the Optotrak and the tracking methods. It is worth noting that after the initialization, no Optotrak information was used while tracking. A majority of the segments show submillimeter differences. Though three segments were greater than a millimeter, tracking was never lost over the course of all of the video. In Fig. 13, we display an example Optotrak trajectory and the corresponding video-based tracking trajectory. The two agree nearly completely with the exception of a different starting location, which was the result of the video-based registration.

### TABLE XI: Mean pose error and standard deviation of the trajectories

<table>
<thead>
<tr>
<th>Navigation</th>
<th>Cause</th>
<th>Trans. Difference (mm)</th>
<th>Rot. Difference (deg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optotrak</td>
<td></td>
<td>1.54 (1.04)</td>
<td>0.05 (0.03)</td>
</tr>
<tr>
<td>StealthStation</td>
<td>1.68</td>
<td>0.04 (0.02)</td>
<td></td>
</tr>
<tr>
<td>Tracker-Based</td>
<td>0.78</td>
<td>0.06 (0.03)</td>
<td></td>
</tr>
<tr>
<td>Video-Based</td>
<td>0.93</td>
<td>0.03 (0.03)</td>
<td></td>
</tr>
<tr>
<td>Mean Difference</td>
<td>1.15</td>
<td>0.05</td>
<td></td>
</tr>
</tbody>
</table>

**F. Initialization Sensitivity**

Fig. 14 is a comparison of the Optotrak location TRE with in-plane and out-of-plane perturbation and the resulting video-
based registration TRE. The red and blue lines are the Tracker-based and the Video-based median TRE, respectively. The error bars are the first and third quartiles of the TRE distribution and the line, the median. As we mention in section III-D1, TRE is not normally distributed, so again we used the zero-skewness log transform to normality and a two-sample Student’s t-test to test the difference of the distribution. We applied Bonferroni’s correction once more to account for multiple tests. The video-based registration has a significant improvement for a majority of the perturbations in-plane. A single star (*) indicates \( p < 0.0024, \alpha = 0.05/21 \), and no stars indicate no significant difference \( (p > 0.0024, \alpha = 0.05/21) \). In Fig. 14 trend regions between the tracker-based and video-based TRE are evident, indicating that the scene geometry was not sufficient to minimize the registration further. In both the in-plane and out-of-plane motion cases there are minima at -1 mm, revealing the inherent 1 mm error of the Optotrak measurements. Also in Fig. 14b beyond 2 mm the scale was not solved correctly. We conclude that for best performance a tracking system with at least 2 mm accuracy would be needed to initialize the registration to ensure sub-millimeter registration and tracking.

G. Shape Accuracy

We observed that the shape reconstructed agreed well with the CT surface and consistently fit within 1 mm. Table XII details the result of the reconstruction for each of the segments used for tracking.

**TABLE XII: Mean distance to the surface of the CT**

<table>
<thead>
<tr>
<th>Segment</th>
<th>Mean Distance (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.30</td>
</tr>
<tr>
<td>2</td>
<td>0.36</td>
</tr>
<tr>
<td>3</td>
<td>0.27</td>
</tr>
<tr>
<td>4</td>
<td>0.16</td>
</tr>
<tr>
<td>5</td>
<td>0.18</td>
</tr>
<tr>
<td>6</td>
<td>0.20</td>
</tr>
<tr>
<td>7</td>
<td>0.20</td>
</tr>
<tr>
<td>8</td>
<td>0.22</td>
</tr>
<tr>
<td>9</td>
<td>0.25</td>
</tr>
<tr>
<td>10</td>
<td>0.22</td>
</tr>
<tr>
<td>Mean</td>
<td>0.24</td>
</tr>
</tbody>
</table>

H. Runtime

At present, the algorithm runs in approximately 30 seconds per frame pair on the cadaver dataset. That is 0.13 seconds for feature processing, 18.48 seconds for motion estimation, 0.27 seconds for reconstruction, 8.78 seconds for Video/CT Registration and 6.18 seconds for tracking. This was an improvement over the initial implementation that ran in approximately two minutes per frame pair. We focused on accuracy and robustness instead of speed in initial work. This methodology is common to vision literature. With further engineering, similar methods have been accelerated [43] and our algorithm could potentially be implemented as a real-time system suitable for clinical use. We realized a 300x speed-up of the feature processing by moving this component to the Graphic Processing Unit (GPU) and we continue to investigate porting to a GPU. We see further improvement within reach by taking advantage of the nature parallelization of the sampling in the ASKC method.

V. Discussion and Conclusion

Our results show that video-CT registration can provide sub-millimeter visualization of targets. These results confirm our
previous hypothesis [16] that the video-CT solution is well below 1 mm TRE.

In the phantom study, we saw that the Optotrak clearly outperformed all of the other methods. This is because the phantom was entirely rigid and the aluminium screw targets were segmented and presented little beam-hardening artifact. Such a perfect scenario is only possible under the idealized case the phantom presented but does not represent the surgical scenario.

We have shown that 2 mm accuracy is sufficient to initialize our system, thus existing systems such as the StealthStation provide sufficient accuracy to benefit from video-based registration. In the cadaver study, the 27 gauge syringe needles presented a segmentation challenge as well as a smaller target to touch with a pointer. However, the needle targets are still clear in the video, which explains the improvement of the video-based method in the cadaver study and demonstrates the video-based method as the best method for the cadaver study. We observed, in the first cadaver study, conditions in which our method failed to register but we were able to correct for these with minor changes to the camera settings.

We continue to investigate robust features and tracking methods. Although SIFT features are capable of tracking, they do not track well over the large translations or large illumination changes that are expected endoscopy. For a more accurate reconstruction a large translation is preferred. There has also been recent work in the computer vision literature showing that real-time dense reconstruction is possible [43]. However, the method is tested on images of an office environment with diverse texture. It remains to be shown if the accuracy of the method holds with endoscope images. Other tracking methods we are investigating include direct image registration using image similarity to optimize the pose of the virtual rendering. In addition to speed for clinical use there are other challenges including smoke, dust and bleeding. We are in the process of receiving institutional review board (IRB) approval to record video and track the endoscope in clinical cases. This data will prove valuable in validating our method. To deal with theses challenges, we are looking into ways of toggling video tracking for only video that is clear. To generalize to other areas of endoscopy, the basis for our method would need to be updated to soften our rigid assumption to a semi-rigid or deformable model or utilize intraoperative CT [44].

We also acknowledge that our algorithm does require that there be enough 3D structure and 2D texture in the scene to register and to track. If the endoscope is pointed at a flat surface or in a textureless region such as a surface covered uniformly with blood, our algorithm would not perform well as is. Additionally, if the camera is imaging a plane or symmetric tube, the registration will fail and the camera would need to be moved to a location imaging sufficient 3D structure required to solve the registration. Although, enough motion (approximately 2 mm) is required for an initial registration, afterwards the camera is still able to be tracked from the 2D-3D correspondences, even during periods of small image-to-image motion or no motion. However, our algorithm could be used for local registration enhancement and would therefore add higher accuracy capability to existing tracking systems. We envision a visual re-registration feature that would offer surgeons the option to have higher accuracy for the current scene.

ACKNOWLEDGMENTS

A preliminary version of this work was presented at a main track of MICCAI 2009 [16], CVPR 2008 [22] and SPIE Medical Imaging 2009 [21]. We thank Jeffrey H. Siewerdsen for the use of the TREK software platform for the evaluation of the data. Special thanks to Mariela Zeledón for her kind editorial comments. Finally, we thank the reviewers for their helpful suggestions that helped improve the paper significantly.

REFERENCES


K.-M. Lee, P. Meer, and R.-H. Park, "Robust adaptive segmentation"

Y. Ma, S. Soatto, J. Kosecka, and S. Sastry, "An implementation of sift detector and descriptor,"


