Toward Video-Based Navigation for Endoscopic Endonasal Skull Base Surgery

Daniel Mirota[1], Hanzi Wang[2], Russell H. Taylor[1], Masaru Ishii[3], and Gregory D. Hager[1]

[1] Department of Computer Science, The Johns Hopkins University, Baltimore, MD, USA
[2] School of Computer Science, The University of Adelaide, SA, Australia
[3] Department of Otolaryngology, Johns Hopkins Medical Institutions, Baltimore, MD, USA
dan@cs.jhu.edu

Abstract. Endoscopic endonasal skull base surgery (ESBS) requires high accuracy to ensure safe navigation of the critical anatomy at the anterior skull base. Current navigation systems provide approximately $2mm$ accuracy. This level of registration error is due in part from the indirect nature of tracking used. We propose a method to directly track the position of the endoscope using video data. Our method first reconstructs image feature points from video in 3D, and then registers the reconstructed point cloud to pre-operative data (e.g. CT/MRI). After the initial registration, the system tracks image features and maintains the 2D-3D correspondence of image features and 3D locations. These data are then used to update the current camera pose. We present registration results within 1mm, which matches the accuracy of our validation framework.

1 Introduction

Endoscopic endonasal skull base surgery (ESBS) has gained much interest recently over traditional open surgical approaches as treatment for cancers of the sinus and pituitary gland. Pituitary lesions, though generally benign, are the most common brain tumor. These common pituitary lesions, as well as cancers of the nasal cavity, brain cancers surrounding the nose, and cancers involving both the nose and brain are all treated with ESBS. Treatment with traditional surgical approaches to the skull base are associated with significant morbidities because healthy cranial nerves are sometimes damaged during surgery. Unlike traditional approaches, ESBS is less invasive and is shown to reduce operative time as well as decrease the length of hospital stay [1].

ESBS and traditional approaches are best contrasted with a clinical example. Figure 1 shows a coronal MRI scan of a patient with a pituitary macroadenoma. The central location of this tumor makes it difficult to approach using traditional means. The tumor is flanked by the carotid arteries and the optic chiasm, and the left optic nerve is clearly compressed by tumor. This tumor was removed using an endoscopic endonasal approach. The endoscopic image (insert) was taken just after the tumor was removed. Notice that the optic nerve (ON) has dropped significantly in height–almost to the level of the carotid artery (CA). Manipulating such high-value structures in cases like this are the reason that ESBS remains a painstaking procedure that requires precise knowledge of patient anatomy. Thus, surgical navigation is key for success, especially to aid junior

G.-Z. Yang et al. (Eds.): MICCAI 2009, Part I, LNCS 5761, pp. 91–99, 2009.

surgeons and for complex cases [1], as it provides the surgeon a means to both maintain orientation and 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 $2mm$ with a good registration, and can be far larger with a poor one [2]. Errors of this magnitude could lead to surgical error resulting in high morbidity or in 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. Then the system 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 tracking results within millimeter mean error of an Optotrak (Northern Digital Corp. Waterloo, Canada) measured camera motion used for comparison.

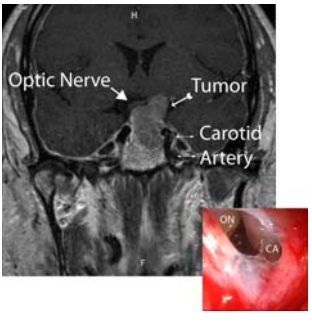

Fig. 1. An example coronal MRI of a patient and endoscopic image (insert) of how clear identification of the carotid artery and optic nerve is crucial

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 another similar system again using passive optical markers for tracking and reported that submillimeter accuracy still remains elusive. Video registration has been previously applied to bronchoscopy [5] where normalized mutual information was used to register to CT. In [6], visual tracking and registration was demonstrated on a skull phantom.

2 Method

There are five major components in our system. Figure 2 shows an overview of the system with the five components highlighted in blue. First, the system extracts SIFT features [7] 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 a segmented surface in the

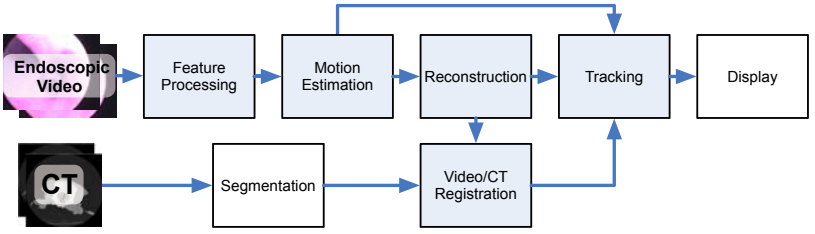

Fig. 2. System Overview

CT image. 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.

2.1 Feature Processing, Motion Estimation and Reconstruction

Before reconstruction or registration, video data are processed to extract image features using the Matlab implementation [8] of SIFT features [7]. We tested both the SIFT feature matching technique suggested by Lowe and SVD SIFT matching technique [9]. While SVD SIFT provides more matches, the matches are not of the same accuracy as Lowe's. However, the larger number of matches increases the number of points in the reconstruction. Figure 3 shows the difference in the two matching methods.

After the image features are detected and matched, the motion is estimated using the robust technique of [10]. To reconstruct points in two or more images the rigid-body transformation between the images must be computed. One 3D point imaged in two images separated by a rotation, R, and translation, $\mathbf{t}$, forms the epipolar constraint. Given a calibrated point in image 1, p_1, and a calibrated point in image 2, p_2, the epipolar constraint is written as

$$p_2 E p_1 = 0 \tag{1}$$

Where $E = sk(\mathbf{t})R$ is known as the Essential Matrix and $sk(\mathbf{t})$ is the skew-symmetric matrix of $\mathbf{t}$. Here, we solve (1) using the robust E-matrix estimator from [10] that uses the five-point algorithm [11]. Motion is estimated by finding the E-matrix that best matches a set of inliers.

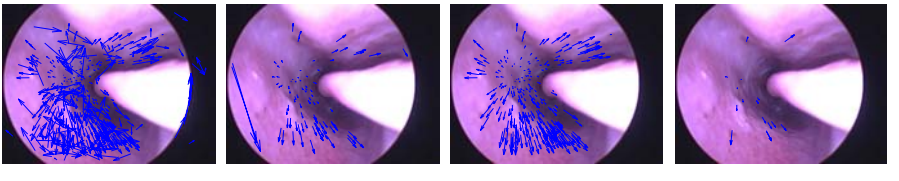

Fig. 3. From left to right: SVD SIFT matches, SIFT matches, inliers of SVD SIFT matches and inliers of SIFT matches

Once the motion estimation is completed, the motion is used to reconstruct the 3D structure. 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.

2.2 Registration and Tracking

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 and using marching cubes to create the isosurface. We applied a registration algorithm described in [12] that is derived from Trimmed ICP (TrICP) [13] and extends TrICP with scale [14]. The registration algorithm needs to estimate scale because the true scale of the 3D world is lost in the epipolar constraint (1).

After the initial 3D-3D registration, the 2D-3D correspondence between image features and the 3D surface of the CT 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 [10] with the pose algorithm of [15] to create a robust 2D-3D pose estimation method. In Algorithm 1, we present an overview of the complete system. $R_{init}, \mathbf{t}_{init}, s_{init}$ are the initial rotation, translation and scale respectively. $points$ is the initial sparse 3D reconstruction. F is a set of images from the video. $mesh$ is the surface mesh segmented from the CT data.

Algorithm 1. $(R, \mathbf{t})$ = Tracking$(R_{init}, \mathbf{t}_{init}, s_{init}, points, F, mesh)$

$R \leftarrow R_{init}, \quad \mathbf{t} \leftarrow \mathbf{t}_{init}, \quad currentPoints \leftarrow s_{init} points$
for all $f_1, f_2 \in F$ where f_1, f_2 are 3 frames apart **do**
  $(\hat{f}_1, \hat{f}_2)$ =undistort(f_1, f_2)
  (sift$_1$, sift$_2$) = detect SIFT feature$(\hat{f}_1, \hat{f}_2)$
  $matches$ = SVDSIFT match(sift$_1$, sift$_2$)
  $(E, inliers)$ = robustMotionEstimator($matches$)
  $currentPoints$ = tracker($matches$,$inliers$,$currentPoints$)
  $(\hat{R}, \hat{\mathbf{t}})$ = robustPoseEstimator($currentPoints, matches, R, \mathbf{t}$)
  $reprojectedInliersPoints$ = reprojectPoints($currentPoints, \hat{R}, \hat{\mathbf{t}}$)
  $(R, \mathbf{t})$ = robustPoseEstimator($reprojectedInliersPoints, matches, \hat{R}, \hat{\mathbf{t}}$)
  $refinedPoints$ = refinePoints($reprojectedInliersPoints$);
  for all previous $R_i, \mathbf{t}_i$ **do**
    $(R_i, \mathbf{t}_i)$ = robustPoseEstimator($refinedPoints_i, matches_i, R_i, \mathbf{t}_i$)
  end for
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 [10]. Once the new camera pose is estimated, we applied one of the following two methods to refine both the 3D points

and pose estimates. Method I, applies all of the previously seen image features and projection matrices to determine the refined 3D point by setting up the following null space problem. First, given a simple projection model $\begin{bmatrix} u_i & v_i & 1 \end{bmatrix}^T = \begin{bmatrix} R_i & \mathbf{t}_i \end{bmatrix} \begin{bmatrix} x_i & y_i & z_i & 1 \end{bmatrix}^T$.

Let $R_i = \begin{bmatrix} r_{1,i}^T & r_{2,i}^T & r_{3,i}^T \end{bmatrix}^T$ and $\mathbf{t}_i = [t_{1,i}\ t_{2,i}\ t_{3,i}]^T$. Then the 3D point which was created by all of the imaged points is the null space solution of

$$0 = \begin{bmatrix} r_{1,1}^T\ t_{1,1} - u_1\, r_{3,1}^T\ t_{3,1} \\ r_{2,1}^T\ t_{2,1} - v_1\, r_{3,1}^T\ t_{3,1} \\ \vdots \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{bmatrix} \tag{2}$$

Method II, uses the initially registered camera position to raycast image feature points of the same 3D point onto the CT scan and calculates the mean of the projected points. We compute each ray as follows. First given the camera's intrinsic parameters, K, rotation, R, and translation, $\mathbf{t}$. We project a point, p, by applying the intrinsic parameters, $p_n = K^{-1}p$. Then the ray is $p_r = (Rp_n + \mathbf{t}) - \mathbf{t}$. The refined points are then used with the robust 2D-3D pose estimator to compute the new camera pose.

3 Experiments

3.1 Data Collection

We collected endoscopic ex-vivo porcine sinus video data. The video was captured at 640x480 using framegrabber attached to a Storz Telecam, 202212113U NTSC with a zero-degree rigid monocular endoscope. Optotrak rigid-bodies were attached to the endoscope and porcine specimen. The Optotrak and video data were simultaneously recorded. The Optotrak motion data were used as the ground truth to compare with the estimated endoscopic motion. Before the data collection, images of a checkerboard calibration grid were also recorded using the endoscope. We performed an offline camera calibration of the endoscope using the Matlab Camera Calibration Toolkit [16]. After the camera calibration, the hand-eye calibration between the Optotrak rigid-body and the camera was estimated by solving the AX=XB problem using the method from [17]. The CT data used had 0.5x0.5x$1.25mm$ voxels.

Our data collection had three major sources of error: one, the camera calibration; two, the Optotrak to camera calibration; and three the Optotrak to CT registration. The camera was calibrated within a pixel error of $[0.38, 0.37]$. The aforementioned Optotrak/camera configuration had an estimated position uncertainty in millimeters of the camera center as measured by the Optotrak of $[1.1, 0.4, 0.1]$. Finally, the Optotrak to CT registration was generally within $.5mm$ RMS point distance error. Each of these contributes to an overall location uncertainty of approximately $1.5mm$ in the absolute position of the endoscope optical center, and approximately $1.1mm$ relative position accuracy.

3.2 Results

We tested our algorithm on 14 randomly selected segments of video from a porcine sinus. Each segment was from a different region of the sinus from anterior to posterior shown

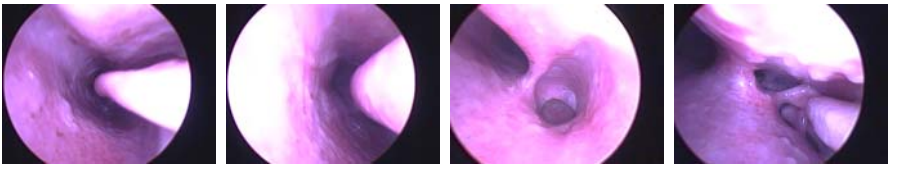

Fig. 4. From left to right the first image of video segments 1 through 4

in figure 4. Figure 4 also shows the challenges of these images including specularities, highlights, motion blurring, lack of texture, and over saturation.

In figure 5 we present both our estimated and Optotrak measured trajectories as well as error along the trajectories for video segment 1. The error is computed as follows. Each image has both an estimated pose and pose measured by the Optotrak. For example $R_1, \mathbf{t}_1$ and $R_{O1}, \mathbf{t}_{O1}$ where $_O$ denotes an Optotrak measurement. The error is then given by $F_{error} = \begin{bmatrix} R_{O1} & \mathbf{t}_{O1} \\ \mathbf{0}^T & 1 \end{bmatrix}^{-1} \begin{bmatrix} R_1 & \mathbf{t}_1 \\ \mathbf{0}^T & 1 \end{bmatrix}$. The translation error is the l_2-norm of the translation component of F_{error}. The rotation error is the absolute sum of the Euler angles of the rotation component of F_{error}. The results are aligned to the Optotrak pose.

Of the 14 segments, one failed to maintain at least five points and three diverged. The three that diverged did not have sufficient 3D structure in the scene for initial registration. The overall accuracy of the remaining 10 achieved by the proposed algorithm is within approximately 1 millimeter of the Optotrak measurement of relative motion. That is, our results are within the error envelope of our measurement system itself. The quantitative tracking results are shown in table 1. We compare our method versus the Optotrak visually in figure 6 which reveals that the Optotrak (the right of each pair) does not align as well as the visually registered image.

Table 1. Mean pose error and standard deviation along of the trajectories

	Trans. Error (mm)		Rot. Error (deg)	
	Method 1	Method 2	Method 1	Method 2
Segment 1	0.83 (0.38)	1.08 (0.52)	1.42 (0.94)	1.39 (1.08)
Segment 2	0.90 (0.40)	0.94 (0.74)	1.07 (0.86)	1.80 (1.90)
Segment 3	0.94 (0.76)	0.51 (0.40)	0.66 (0.56)	1.46 (1.34)
Segment 4	1.24 (0.83)	0.96 (0.52)	1.79 (1.00)	0.91 (0.84)
Segment 5	0.64 (0.34)	0.58 (0.29)	1.14 (0.98)	0.65 (0.39)
Segment 6	0.41 (0.21)	0.75 (0.48)	1.93 (1.02)	1.49 (0.96)
Segment 7	1.32 (0.97)	2.5 (2.21)	3.68 (1.54)	6.29 (4.34)
Segment 8	0.40 (0.24)	0.33 (0.22)	1.79 (1.09)	1.54 (1.43)
Segment 9	1.03 (0.72)	0.43 (0.29)	0.59 (0.24)	0.66 (0.46)
Segment 10	0.56 (0.31)	0.55 (0.31)	2.94 (2.15)	0.90 (0.54)
Mean Error	0.83	0.86	1.70	1.71

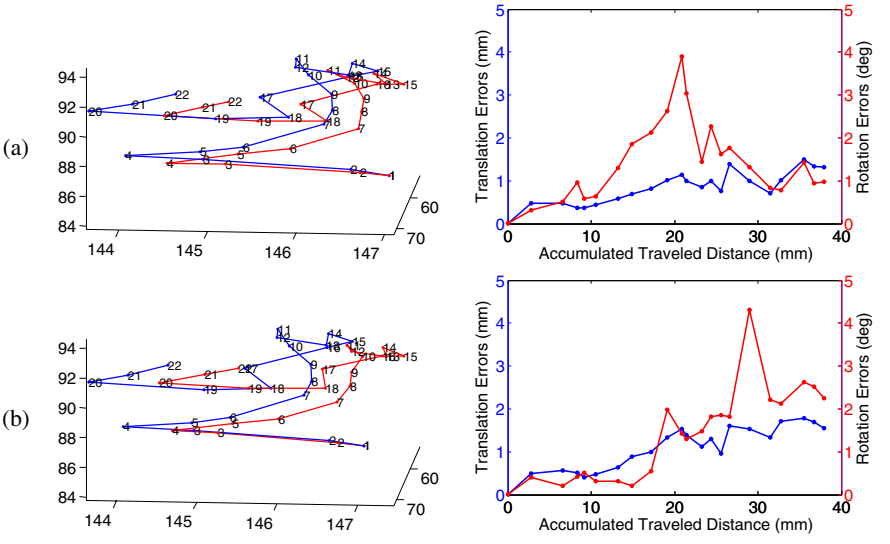

Fig. 5. Results from the video segment 1 with Method I (a) and Method II (b). Left:The trajectories of our method (red) and Optotrak (blue), Right:frame to frame relative error along the trajectories.

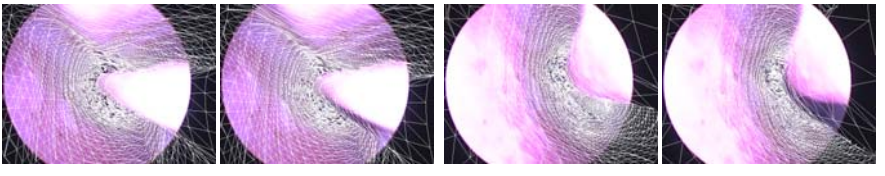

Fig. 6. Comparison of our method (Left) and Optotrak (Right). Left Pair, beginning of tracking. Right Pair, end of tracking.

4 Discussion and Conclusion

Our results indicate that video-CT registration can be computed to within $1mm$ of our ground truth validation system. Recall $1mm$ is the expected error of the Optotrak measurements alone. We hypothesize that the video-CT solution is well below $1mm$ error since we are within the error envelope of the Optotrak, and because visual inspection of the registrations often shows a more consistent result from the video-CT algorithm. We present relative error as we continue investigate a method that is more accurate than the Optotrak tracking a rigid endoscope.

In future work, we will validate that we are indeed below $1mm$ error by using CT-compatible screws that would be imaged in both the endoscope and CT. Using the segmented location from both the endoscope and CT, a 2D-3D pose estimator can be used to validate the registration error to within the resolution of the CT image itself.

We will also investigate robust features to more accurately track over large translations. While SIFT features do offer image features to track, they do not track well over large translation. SIFT features fail to track over large translation in endoscopy because SIFT is not particularly robust to illumination changes. In endoscopy the light source is collocated with the camera and thus the illumination is always changing as the camera moves. For a more accurate reconstruction a large translation is preferred. Beyond image features, the selection of image pairs to use for reconstruction can be automated. It is important to select image pairs with motion greater than the noise of the image features tracked to ensure accurate reconstruction.

The current system is implemented in Matlab and processes the data offline taking about two minutes per frame pair. We focused on accuracy and robustness instead of speed. This methodology is common to vision literature. With further engineering the algorithm could be turned into an online system suitable for clinical use.

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, our algorithm would not perform well. 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. This work was supported principally by the National Institutes of Health under grant number R21 EB005201. Further support was provided by the NSF ERC Grant EEC9731748 and by Johns Hopkins University. We thank Dr. Darius Burschka for providing the porcine video data.

References

1. Nasseri, S.S., Kasperbauer, J.L., Strome, S.E., McCaffrey, T.V., Atkinson, J.L., Meyer, F.B.: Endoscopic transnasal pituitary surgery: Report on 180 cases. American Journal of Rhinology 15(4), 281–287 (2001)
2. Chassat, F., Lavallée, S.: Experimental protocol of accuracy evaluation of 6-D localizers for computer-integrated surgery: Application to four optical localizers. In: Wells, W.M., Colchester, A.C.F., Delp, S.L. (eds.) MICCAI 1998. LNCS, vol. 1496, pp. 277–284. Springer, Heidelberg (1998)
3. Shahidi, R., Bax, M., Maurer, C.R., Johnson, J., Wilkinson, E., Wang, B., West, J., Citardi, M., Manwaring, K., Khadem, R.: Implementation, calibration and accuracy testing of an image-enhanced endoscopy system. Med. Imag., IEEE Trans. 21(12), 1524–1535 (2002)
4. Lapeer, R., Chen, M.S., Gonzalez, G., Linney, A., Alusi, G.: Image-enhanced surgical navigation for endoscopic sinus surgery: evaluating calibration, registration and tracking. Intl. J. of Med. Rob. and Comp. Assisted Surg. 4(1), 32–45 (2008)
5. Helferty, J.P., Hoffman, E.A., McLennan, G., Higgins, W.E.: CT-video registration accuracy for virtual guidance of bronchoscopy. In: SPIE Med. Imaging, vol. 5369, pp. 150–164 (2004)
6. Burschka, D., Li, M., Ishii, M., Taylor, R.H., Hager, G.D.: Scale-invariant registration of monocular endoscopic images to CT-scans for sinus surgery. MIA 9, 413–426 (2005)
7. Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 20, 91–110 (2003)

8. Vedaldi, A.: SIFT for matlab, http://www.vlfeat.org/~vedaldi/code/sift.html (last accessed May 29, 2009)
9. Delponte, E., Isgr, F., Odone, F., Verri, A.: SVD-matching using SIFT features. In: Proc. of the Intl. Conf. on Vision, Video and Graphics, July 2005, pp. 125–132 (2005)
10. Wang, H., Mirota, D., Ishii, M., Hager, G.: Robust motion estimation and structure recovery from endoscopic image sequences with an Adaptive Scale Kernel Consensus estimator. In: CVPR. IEEE Conf. on, June 2008, pp. 1–7 (2008)
11. Nister, D.: An efficient solution to the five-point relative pose problem. IEEE Trans. PAMI 26(6), 756–770 (2004)
12. Mirota, D., Taylor, R.H., Ishii, M., Hager, G.D.: Direct endoscopic video registration for sinus surgery. In: Medical Imaging 2009: Visualization, Image-guided Procedures and Modeling, Proc. of the SPIE, February 2009, vol. 7261, pp. 72612K–1,72612K–8 (2009)
13. Chetverikov, D., Svirko, D., Stepanov, D., Krsek, P.: The trimmed iterative closest point algorithm. ICPR 3, 545–548 (2002)
14. Du, S., Zheng, N., Ying, S., You, Q., Wu, Y.: An extension of the ICP algorithm considering scale factor. ICIP 5, 193–196 (2007)
15. Lu, C.P., Hager, G., Mjolsness, E.: Fast and globally convergent pose estimation from video images. PAMI, IEEE Trans. 22(6), 610–622 (2000)
16. Bouget, J.Y.: The matlab camera calibration toolkit, http://www.vision.caltech.edu/bouguetj/calib_doc/ (last accessed, March 3 2008)
17. Park, F., Martin, B.: Robot sensor calibration: solving AX=XB on the Euclidean group. IEEE Transactions on Robotics and Automation 10(5), 717–721 (1994)