

Image-Guided Navigation: A Cost Effective Practical Introduction using the Image-Guided Surgery Toolkit (IGSTK)

Özgür Güler and Ziv Yaniv

Abstract—Teaching the key technical aspects of image-guided interventions using a hands-on approach is a challenging task. This is primarily due to the high cost and lack of accessibility to imaging and tracking systems. We provide a software and data infrastructure which addresses both challenges. Our infrastructure allows students, patients, and clinicians to develop an understanding of the key technologies by using them, and possibly by developing additional components and integrating them into a simple navigation system which we provide. Our approach requires minimal hardware, LEGO blocks to construct a phantom for which we provide CT scans, and a webcam which when combined with our software provides the functionality of a tracking system. A premise of this approach is that tracking accuracy is sufficient for our purpose. We evaluate the accuracy provided by a consumer grade webcam and show that it is sufficient for educational use. We provide an open source implementation of all the components required for a basic image-guided navigation as part of the Image-Guided Surgery Toolkit (IGSTK). It has long been known that in education there is no substitute for hands-on experience, to quote Sophocles, "One must learn by doing the thing; for though you think you know it, you have no certainty, until you try.". Our work provides this missing capability in the context of image-guided navigation. Enabling a wide audience to learn and experience the use of a navigation system.

I. INTRODUCTION

Image-guided navigation systems aim to improve clinical outcomes by reducing procedure invasiveness while minimizing the use of imaging modalities that utilize ionizing radiation. This is achieved by integrating medical images with other sources of information such as tracked instruments, in effect, enhancing the physician's understanding of the underlying spatial relationships between anatomy and tools. These systems incorporate several key technologies [1], [2], [3], including: (1) Medical imaging; (2) Data visualization; (3) Segmentation; (4) Registration; (5) Tracking systems; and (6) Human Computer Interaction.

Teaching the concepts underlying image-guided navigation can take different forms, depending upon the intended audience. In this work we describe an infrastructure which supports hands-on education for two target audiences, developers (i.e. engineers) and users (i.e. patients or physicians) of the technology.

Teaching engineers the concepts underlying the technologies listed above is most often done with an emphasis on the

theoretical aspects, with students gaining limited practical experience. In addition, when taught as a set of disjoint technologies students do not appreciate the importance and complexity of system integration. Educating patients and physicians with regard to the principles of image-guided navigation has not been addressed in the past in a hands-on manner. The main reasons for the current state are related to high cost and lack of accessibility, both to imaging devices and tracking systems.

Our work addresses these issues. This is done by providing CT images of an easily constructed LEGO phantom, a software infrastructure that enables engineers to implement specific components and integrate them into a navigation system, and a set of programs that allow physicians and patients to simulate an "intervention".

A similar cost effective educational approach to image-guided robotic interventions was presented in [4]. In that tutorial a LEGO Mindstorms NXT robot was used to perform a simulated biopsy in a LEGO phantom. The provided software was implemented as a module for the open source 3D Slicer program [5]. The tutorial provides hands-on experience from a user perspective, and requires the use of a computer running the Linux operating system to control the robot. Our work reduces the cost even further, as we do not require a robot. In addition, we designed our software infrastructure so that it can serve as a teaching aid for both developers and users of image-guided navigation systems. Finally, by implementing the software components as part of the Image-Guided Surgery Toolkit (IGSTK) [6] we ensure that the software runs on a variety of operating systems, including Windows, Linux and MacOS.

The image-guided surgery toolkit is a platform independent open source C++ toolkit which is freely available for both academic and commercial uses [7]. It provides a framework for the development of image-guidance applications. One of its key goals is to serve as an educational resource for training newcomers to the field. Our current work was motivated by this goal.

II. MATERIALS AND METHODS

We next describe our data and software infrastructure and how it facilitates a hands-on approach to teaching the concepts underlying image-guided navigation. We then describe how this infrastructure can be used both to educate patients and physicians, and in an engineering curriculum.

Ö. Güler is with The Sheikh Zayed Institute for Pediatric Surgical Innovation, Children's National Medical Center, Washington, DC, USA
OGuler@childrensnational.org

Z. Yaniv is with The Sheikh Zayed Institute for Pediatric Surgical Innovation, Children's National Medical Center, Washington, DC, USA
ZYaniv@childrensnational.org

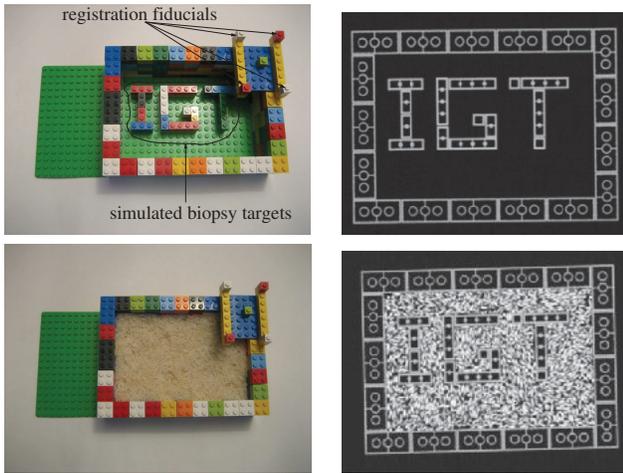


Fig. 1. Phantom and a coronal slice from corresponding CT data sets. The volumetric images can be used for hands-on experimentation with intensity and fiducial based image-to-image registration, and for guidance of a simulated biopsy when filled with rice (internal target structures are no longer visible).

A. Phantoms and Data

For compatibility with the image-guided robotic biopsy tutorial [4] we used the same LEGO set, "Deluxe Brick Box Building Set (6167)", to construct our phantom. As we only require standard LEGO blocks other sets are also appropriate. We provide instructions for a simple phantom which can be constructed in less than 15 minutes. We also provide three CT scans of the phantom, oriented along the scan direction, oriented at an angle to the scan direction, and with rice placed inside the phantom. By filling the phantom with rice we obstruct the view of its internal structures, necessitating navigation guidance when simulating a biopsy procedure. Figure 1 shows the phantom and coronal slices from two of the scans. All scans were acquired with a GE Healthcare Discovery 690 PET/CT scanner and reconstructed with a spatial resolution of $0.39 \times 0.39 \times 1.25 \text{mm}^3$.

B. Software

We provide a set of software components that implement the following elements that comprise a fully functioning image-guided navigation system: camera calibration, tracking, pointer tool calibration, pre-operative planning, registration, and visualization.

Our tracking approach requires a calibrated camera. We provide a camera calibration application based on the OpenCV toolkit's [8] implementation of Zhang's camera calibration algorithm [9]. Images of a planar checkerboard pattern in multiple orientations, with regard to the camera, are acquired. The x-corner points of the pattern are automatically detected and the camera is calibrated. Calibration is validated by acquiring images of a second checkerboard pattern with known metric properties. Two sets of corners are detected, each comprised of four vertices of a rectangle that define a local coordinate system. The transformation from the camera coordinate system to each of these local coordinate systems is estimated. The error in the estimated distances

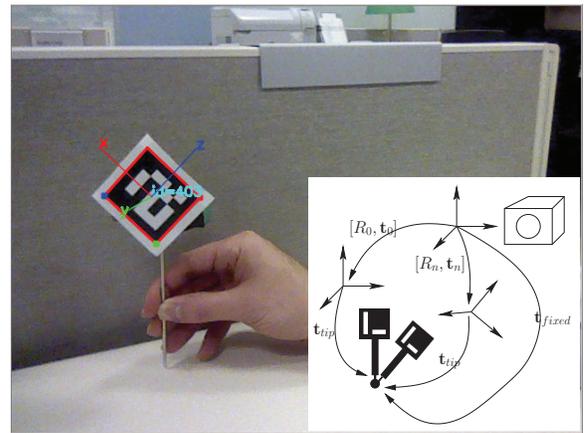


Fig. 2. Image from laptop webcam used to track a pointer tool. The tool is pivoted around a fixed point to estimate the translation between the fiducial marker's coordinate system (projected onto the image) and the tool's tip. Inset shows coordinate systems and transformations used for calibration.

and orientations is used to quantify the calibration error. The calibration process, image acquisition and computation, is repeated until a sufficiently accurate calibration is achieved.

Tracking of tools and anatomical structures is most often performed using costly specialized hardware that provides sub-millimetric accuracy [10]. Currently, the most cost effective clinically used tracking systems cost on the order of several thousand US dollars. Thus, it is not common to use these systems in an educational setting. We provide monocular tracking of fiducial markers using a calibrated webcam and software based pose estimation.

The two common approaches to monocular pose estimation are, detection based estimation and recursive estimation [11], [12]. In the former, fiducial pose is estimated independently from each image. In the latter, fiducial pose is estimated based on the current image and on the pose in previous images. Consequentially, detection based estimation is more robust to marker occlusion, but it also results in noisier pose estimates. The augmented reality research community has developed several libraries that provide detection based tracking using planar markers. Among others, these include the AR-Toolkit [13], the AR-ToolkitPlus [14], and the ArUco library [15].

We have incorporated the ArUco library into the IGSTK framework, as it is a lightweight open source C++ library with a BSD license, making it the most compatible with the toolkit. It allows us to track the spatial location and orientation of printed fiducial markers using a calibrated camera. Figure 2 shows a pointer tool, tracked using a laptop webcam. The planar markers we use have a size of $50 \times 50 \text{mm}^2$.

To digitize points in the physical world we use a tracked pointer tool. This requires that the tool be calibrated. That is, the translation between the tracked marker and the pointer tip needs to be estimated. This is done by pivoting, rotating the pointer while keeping its tip in a fixed location. We provide a program that allows the user to perform this pivot calibration.

Figure 2 shows the physical setup and a schematic of the process. Every marker transformation $T_i = [R_i, \mathbf{t}_i]$ yields the following three equations in six unknowns:

$$R_i \mathbf{t}_{tip} + \mathbf{t}_i = \mathbf{t}_{fixed}$$

We acquire multiple transformations and solve a least squares problem:

$$\begin{bmatrix} R_0 | -I \\ \vdots \\ R_n | -I \end{bmatrix} \begin{bmatrix} \mathbf{t}_{tip} \\ \mathbf{t}_{fixed} \end{bmatrix} = \begin{bmatrix} -\mathbf{t}_0 \\ \vdots \\ -\mathbf{t}_n \end{bmatrix}$$

The pre-operative planning software we provide allows the user to visualize the CT image using the standard radiological views, and to indicate registration fiducial locations and target locations which are saved and used as input to the navigation software.

Our navigation software allows the user to load the CT image, register the image space with the physical world, and provides navigation guidance. The system overlays an iconic representation of the pointer tool onto a dynamically updated display of the image planes where the displayed planes coincide with the location of the pointer tool's tip. Registration is implemented using the paired-point rigid registration algorithm described in [16]. Points in the image are defined in the pre-operative planning software and the corresponding points are digitized using the calibrated pointer.

A key premise of our approach is that the tracking accuracy provided by a consumer grade webcam is sufficient, at least for educational use. We thus need to evaluate the tracking accuracy.

C. Tracking Evaluation

We assess the quality of our monocular tracking implementation using two criteria, its refresh rate and its accuracy. In all experiments we used a laptop webcam having a resolution of 640×480 . The size of the planar markers was $50 \times 50 \text{mm}^2$.

To evaluate refresh rate we acquired 1000 transformations for a marker while moving it inside the tracked volume. Each transformation was time stamped and saved in memory. At the end of the acquisition all data was written to file.

Tracking accuracy was evaluated by using two markers with a known transformation between them. The transformation to each of the markers is estimated and the transformation between the two coordinate systems is obtained. We then compare this transformation to the known one using the error introduced to point location, $TRE = \|T_{known}\mathbf{p} - T_{estimated}\mathbf{p}\|$, with a set of 20 arbitrary points in the spatial region $[0, 100] \times [0, 100] \times [0, 100] \text{mm}^3$. The experiment was repeated at 27 spatial locations in the tracked volume. Figure 3 shows our experimental setup.

D. Image-Guided Navigation Tutorial

For users of image-guided navigation systems, both physicians and patients, our software infrastructure enables them

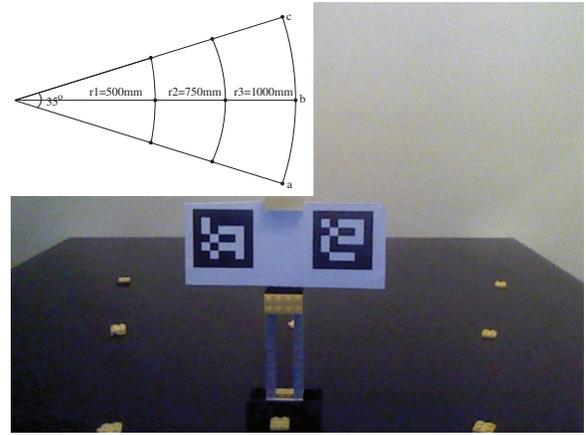


Fig. 3. Tracked markers used for accuracy evaluation. Transformation between the two markers is known (translation of 100mm). Inset shows the nine locations at which data was acquired. In each location the markers were placed at three elevations (5mm, 115mm, 260mm) providing us with error estimates at 27 spatial locations in the tracking volume. Picture is from webcam with markers at location (r1,b) with elevation of 115mm.

to develop an understanding of several key concepts including calibration, registration, and tracking. As a first step, the users build the LEGO phantom. They are then introduced to the concept of calibration in two forms, camera calibration, relating points in space with their location in the image, and pointer tool calibration, relating the location of the tool tip with regard to the marker location. The users are then introduced to the concept of registration, relating points in space with those in the image. The terminology associated with paired point registration denoting the various errors [17], Fiducial Localization Error (FLE), Fiducial Registration Error (FRE), and Target Registration Error (TRE), is also described. The users then see that FRE does not reflect the quality of registration by using the navigation system and physically changing the fiducial locations. Finally, they are introduced to the concept of optical tracking, and the use of a Dynamic Reference Frame (DRF) which allows movement of the tracking device or patient while maintaining the relationship between the physical world and the image.

For engineering students our infrastructure serves two functions, it allows them both, to develop an understanding of key concepts, as described above, and it serves as a scaffolding for practical course exercises. In this setting various software components can be developed and integrated into the existing programs. Potential components include: (1) Volume rendering of the anatomy [18]. In this task the students replace the current visualization in which three orthogonal planes are displayed in space with volume rendering; (2) Iso-surface extraction using the marching cubes algorithm [19]. In this task the three CT volumes are used to illustrate various aspects of the algorithm, such as the global effect of the iso-surface value where both the rice and LEGO blocks become part of the same iso-surface, and the issue of aliasing which arises when the sampling along the scanning direction is not sufficiently dense when the phantom is at an angle to the scan direction. (3) Improving the tracking robustness.

In this task the students modify the tracker implementation so that multiple markers are associated with the same tool. This enables tracking of the pointer tool from all angles relaxing the requirement that the planar marker face the camera at all times. (4) Implementation of the Iterative Closest Point algorithm [20]. In this task the students replace the paired-point registration we provide with a surface based registration, aligning the iso-surface they obtained in the previous task with points digitized on the phantom's surface. (5) Implementation of a virtual keyboard [21]. In this task the students implement a custom keyboard, a tracked panel, that activates specific commands when the pointer tool is in close proximity to various locations on the panel.

III. EXPERIMENTAL RESULTS

The refresh rate estimated using the approach described above was 23Hz which is sufficient for our application.

The tracking accuracy varied according to spatial location and was better when the markers were closer to the center of the image and degraded when they were in locations closer to the periphery. Table I summarizes our evaluation.

	a	b	c
r1	12.2, 9.7, 22.0	5.4, 13.3, 4.7	31.2, 20.1, 18.1
r2	30.1, 7.4, 24.1	10.7, 9.6, 7.1	2.0, 23.1, 12.6
r3	34.2, 15.9, 13.3	20.8, 4.8, 12.4	19.8, 10.7, 4.2

TABLE I

TRE ERRORS, IN MM, FOR ACCURACY EXPERIMENTS CORRESPONDING TO LOCATIONS AS SHOWN IN FIGURE 3. AT EACH LOCATION RESULTS ARE ORDERED ACCORDING TO HEIGHT FROM TOP TO BOTTOM.

IV. DISCUSSION AND CONCLUSIONS

It is often the case concepts taught in a theoretical manner are only understood when they are used in practice. In the domain of image-guided navigation the high cost and limited accessibility of imaging and tracking systems has precluded the use of hands-on learning outside of research laboratories. We provide an infrastructure for supporting a hands-on approach to teaching the concepts of image-guided navigation. Our approach is based on providing CT data of a simple LEGO phantom and a set of software components that provide all of the required functionality of a basic navigation system.

We illustrated the use of the framework for educating users and developers of navigation systems. For the former our infrastructure is used to familiarize them with the nomenclature and the possible pitfalls associated with the technology. For the latter the infrastructure can be used as a scaffolding for practical course exercises.

As we utilize a standard consumer grade webcam to perform the tracking we evaluated this approach's accuracy. With our specific choices of webcam and marker size tracking accuracy is only sufficient for instructional use. While accuracy remains an issue, even with limited accuracy this tracking approach is useful as a surrogate for commercial tracking systems when developing a navigation system. In

this manner software can be developed without requiring the availability of the commercial tracking system at all times.

The data, executables and source code are available from http://public.kitware.com/IGSTKWIKI/index.php/IGI_Tutorial

ACKNOWLEDGMENT

We would like to thank Dr. N. Safdar for his help acquiring the CT images.

REFERENCES

- [1] K. Cleary and T. M. Peters, "Image-guided interventions: technology review and clinical applications," *Annu. Rev. Biomed. Eng.*, vol. 12, pp. 119–142, 2010.
- [2] T. Peters and K. Cleary, Eds., *Image-Guided Interventions Technology and Applications*. Springer, 2008.
- [3] Z. Yaniv and K. Cleary, "Image-guided procedures: A review," Image Science and Information Systems Center, Georgetown University, Tech. Rep. CAIMR TR-2006-3, April 2006.
- [4] D. F. Pace, R. Kikinis, and N. Hata, "An accessible, hands-on tutorial system for image-guided therapy and medical robotics using a robot and open-source software," in *Open Science Workshop (MICCAI)*, 2007.
- [5] "3D Slicer," last accessed March 15 2012. [Online]. Available: <http://www.slicer.org/>
- [6] A. Enquobahrie, P. Cheng, K. Gary, L. Ibáñez, D. Gobbi, F. Lindseth, Z. Yaniv, S. Aylward, J. Jomier, and K. Cleary, "The image-guided surgery toolkit IGSTK: An open source C++ software toolkit," *Journal of Digital Imaging*, vol. 20, no. Suppl. 1, pp. 21–33, 2007.
- [7] "The Image-Guided Surgery Toolkit," last accessed March 15 2012. [Online]. Available: <http://www.igstk.org>
- [8] G. Bradski, "The OpenCV Library," *Dr. Dobbs' Journal of Software Tools*, vol. 25, no. 11, pp. 120–126, 2000.
- [9] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 11, pp. 1330–1334, 2000.
- [10] W. Birkfellner, J. Hummel, E. Wilson, and K. Cleary, "Tracking devices," in *Image-Guided Interventions Technology and Applications*. Springer, 2008.
- [11] V. Lepetit and P. Fua, "Monocular model-based 3D tracking of rigid objects: A survey," *Foundations and Trends in Computer Graphics and Vision*, vol. 1, no. 1, 2005.
- [12] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, 2006.
- [13] H. Kato and M. Billinghurst, "Marker tracking and HMD calibration for a video-based augmented reality conferencing system," in *Proceedings. 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR '99)*, 1999, pp. 85–94.
- [14] D. Wagner and D. Schmalstieg, "ARToolKitPlus for pose tracking on mobile devices," in *Proceedings of 12th Computer Vision Winter Workshop (CVWW'07)*, 2007.
- [15] "ArUco: a minimal library for augmented reality applications based on OpenCV," last accessed March 15 2012. [Online]. Available: <http://www.uco.es/investiga/grupos/ava/node/26>
- [16] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Journal of the Optical Society of America A*, vol. 4, no. 4, pp. 629–642, April 1987.
- [17] C. R. Maurer, Jr., J. M. Fitzpatrick, M. Y. Wang, R. L. Galloway, Jr., R. J. Maciunas, and G. S. Allen, "Registration of head volume images using implantable fiducial markers," *IEEE Trans. Med. Imag.*, vol. 16, no. 4, pp. 447–462, 1997.
- [18] B. Lichtenbelt, R. Crane, and S. Naqvi, *Introduction to volume rendering*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1998.
- [19] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," *Computer Graphics*, vol. 21, no. 4, pp. 163–169, 1987.
- [20] P. J. Besl and N. D. McKay, "A method for registration of 3D shapes," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, no. 2, pp. 239–255, 1992.
- [21] H. Visarius, J. Gong, C. Scheer, S. Haralamb, and L. P. Nolte, "Man-machine interfaces in computer assisted surgery," *Computer Aided Surgery*, vol. 2, no. 2, pp. 102–107, 1997.