Preliminary Design and Evaluation of an Interfacing Mechanism for Maneuvering Virtual Minimally Invasive Surgical Instruments

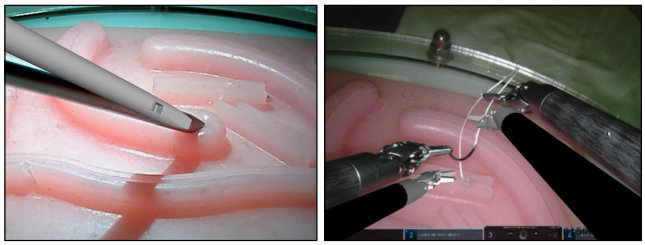
Mai Trinh[1], Jhasketan Padhan[2], Nikhil V. Navkar[2], and Zhigang Deng[1]

***Abstract*— Augmenting the motion of virtual surgical instruments onto a minimally invasive surgical field acts as a visual cue for the operating surgeon. In this work we propose an interfacing mechanism to provide input for maneuvering such virtual surgical instruments. Specifically, an interface in the form of a 3D-printed dodecahedron pen with attached binary squared planar markers is employed. The proposed tracking mechanism computes the pose of the interface from a real-time video feed acquired from a camera. The system provides accurate pose estimation with mean errors of 0.27 $\pm$ 0.06 mm in translation and 0.37 $\pm$ 0.04 degrees in rotation. The object pose estimation takes $\sim$6 ms. Utilized Azure Kinect camera with frame rate of 30 FPS and 1280 x 720 image resolution video, the tracking speed of the proposed system is $\sim$25 FPS . The easy to integrate, cost effective setup makes the interfacing mechanism particularly suitable for remote surgical tele-mentoring applications.**

I. Introduction

Minimally Invasive Surgery (MIS) has emerged as a more patient-friendly and practice-enhancing alternative to traditional open surgeries with long-term cost-effectiveness. It entails the use of elongated surgical instruments and a scope inserted through small incisions. The interaction between the surgical instrument's tooltips and the tissue to be operated is visualized using the scope and is rendered on a screen. Recent studies has demonstrated that augmenting the view of the rendered surgical field with motion of virtual surgical instruments (as shown in Fig. 1) acts as realistic visual cues and facilitates intra-operative surgical mentoring [1], [2], [3], [4]. In such scenarios, a low-cost, reproducible, easy to integrate, interfacing mechanism for maneuvering virtual minimally invasive surgical instruments would be beneficial.

In this work, we present a preliminary design of an interfacing mechanism that requires minimum configurations to be integrated with surgical setup in the operating room. It uses a single camera connected to a workstation for tracking an interface held by the operator. As the operator moves the interface (in form of a 3D Printed Dodeca Pen), the system efficiently processes each frame of the video feed from the camera, computes the pose of the interface (position and orientation), and converts it to a coordinate frame in virtual space (represented by XYZ translations and three rotational angles: roll, yaw, and pitch). Virtual tooltips of the surgical instrument mesh models are mapped to the frame. As the frame moves, the virtual surgical instrument follows the motion under the constraints defined by incision point and instrument's degrees-of-freedom. The video stream (displaying the surgical field) from the scope system is bifurcated and passed to the workstation, the motion of the surgical instruments is overlaid onto the video stream, and the video stream is rendered back onto the visualization screen in the operating room. The motion of the virtual surgical instruments rendered onto the surgical field acts as visual cues for the operating surgeon.



(a) (b)

Fig. 1. A virtual minimally invasive surgical instrument overlaid onto a surgical field. The virtual surgical instrument acts as a visual cue for the operating surgeon. View of the augmented surgical field in (a) Laparoscopic (manual) surgical setup (b) Robotic surgical setup.

This work applies a binary squared planar marker tracking method for real-time surgical instrument tracking. To track an interface and limit occlusion issues, a 3D-printed dodecahedron with attached squared planar markers is employed to ensure a good visibility (i.e. one detectable marker at all time) using only one camera. The major contributions of this work are summarized as follows:

- Real-time tracking of an interface comprising of 3D-printed dodecahedron with attached squared planar markers; and
- apply the estimation of 3D configuration algorithms to determine the pose of the tracked object using a single camera.

II. Related Work

A few surgical instrument tracking approaches using camera images have been proposed, including point cloud based tracking [5], [6], optical tracking [7], vision-based and markerless tracking [8], [9], and 2D marker based tracking [10], [11], [12]. Squared planar markers have been employed

This work is supported by National Priority Research Program (NPRP) award (NPRP12S-0119-190006) from the Qatar National Research Fund (a member of The Qatar Foundation). All opinions, findings, conclusions or recommendations expressed in this work are those of the authors and do not necessarily reflect the views of the sponsors. Zhigang Deng was a consulting professor at East China Jiaotong University, China.

[1]Mai Trinh and Zhigang Deng are with the Department of Computer Science, University of Houston, Houston, TX, USA. Email: mtrinh2@uh.edu, zdeng4@central.uh.edu

[2]Jhasketan Padhan and Nikhil V. Navkar are with the Department of Surgery, Hamad Medical Corporation, Doha, Qatar. Email: nnavkar@hamad.qa

in many of the above methods. Also, tracking using one marker has a limited range of motion and a narrow working area. Moreover, since 2D or planar markers are commonly attached onto flat surfaces, the issue of occlusions becomes a serious problem when tracking 2D markers attached to a cylindrical object. The proposed tracking interface approach in this work aims to overcome these limitations.

Binary squared fiducial marker based methods use an external wide black border and an internal binary matrix that identifies each marker. The pose of a square marker can be easily detected based on its four corners. Square fiduciual markers are mainly used for camera pose estimation and camera calibration [13], [14], [15]. Additionally, object tracking using square fiducial markers have become popular solutions in many applications [16], [17], [18], including autonomous robots, autonomous vehicle, and augmented reality. Its popularity comes from its ability for robustness, precision, and speed [19], [20].

Wu *et al.* [21] proposed a system for 6 DOFs tracking of a passive stylus composed of a 3D printed dodecahedron with attached binary square fiducial markers and a pen. Their system takes input images from a single monocular camera and generates a digital 2D drawing from the tracked stylus pen. This is a cost-effective method using a simple stylus; however, it is too slow for real-time applications which require a high throughput.

To tackle the relative poses between planar markers, Sarmadi *et al.* [22] proposed a novel method for object tracking using multiple cameras and multiple squared planar markers. Their method automatically estimates the 3D configuration of the markers, the camera extrinsic parameters, and the relative pose between the markers and the camera. However, it was only tested in experimental scenarios with at least three cameras.

Zhang *et al.* [10] proposed a new hybrid marker design that combines circular dots and chessboard vertices to track surgical tools to facilitate intra-operative guidance in MIS. There is a limitation in term of the work space such that the distance from the markers to the camera is in the range from 50 to 200mm. As such, their method limits the range of rotational and translational motion in practical applications.

In our approach, we utilize the DodecaPen design in [21] as our object simulator and extend the object configuration algorithm in [22] to estimate the pose of the object based on a single camera, instead of multiple cameras. Once the 6 DoFs pose of the object is obtained, a Kalman filter [23] is utilized to smooth the tracking results by filtering out noise and biased data.

III. Our Method

The overview of our proposed system is illustrated in Fig. 2. The proposed system consists of two phases: initial calibration and object tracking. At the initial calibration step, both an Azure Kinect camera and the interfacing object are calibrated beforehand using vision-based methods. At the runtime object tracking step, our system can real-time process each frame of the video feed from the Azure Kinect camera and estimate the position and orientation of the tracked interface object based on the square binary markers attached to it. Once the pose of the interface object is determined, our system updates the pose of the surgical tool and the tooltip axes and provides an input stream to a VTK + QT system, which renders the virtual surgical tool and update its motion accordingly.

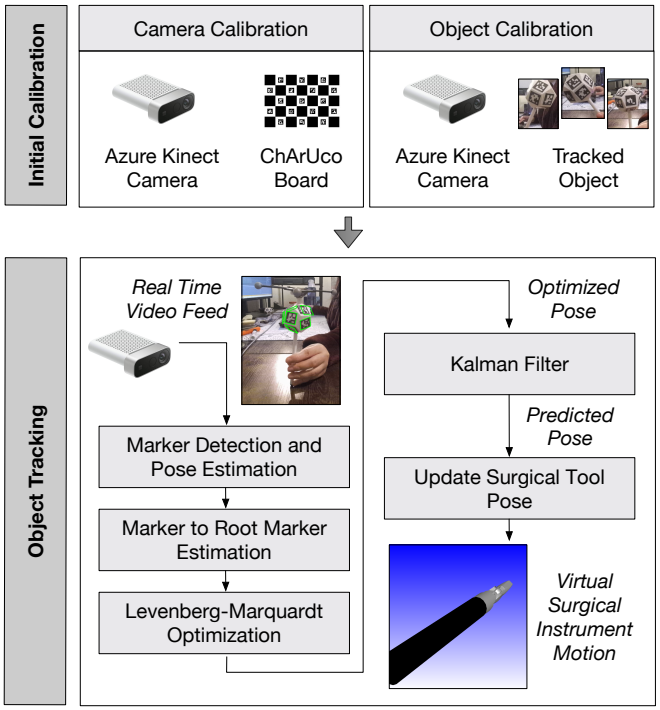


Fig. 2. Overview of the proposed system

A. Object Design

Since binary square fiducial markers are commonly attached to flat surfaces, the issue of occlusions is aggravated when tracking planar markers on a cylindrical object. To address this issue, we employ the 3D-printed dodecahederon design [21] and hand-glued the square binary markers to each of the surfaces of the dodecahedron. Each edge of the resulting dodecahedron is 22 mm in length, and each edge of the marker is 20 mm in length. The ArUco library [19] is employed to generate and detect the square binary markers. A pen or any cylindrical object with roughly 6 mm in diameter can be attached to the dodecahedron (see Fig. 4).

B. Initial Calibration

Before tracking begins, our system needs to perform a one-time initial calibration to determine the camera intrinsic parameters and the relative pose between the markers. To complete the calibration process, the users are required to capture multiple images and provides fixed parameters. By performing the calibration procedure in the beginning, object tracking can be done efficiently.

1) Camera Calibration: To perform camera calibration, we take several images of the printed ChArUco board from multiple angles without moving the camera. A ChArUco

board is a combination of chessboard and ArUco markers to allow partial occlusion and achieve a high accuracy [24]. Given an input image, it detects and estimates the pose of the ArUco markers; then four corners of the ChArUco board are interpolated from these markers. The camera calibration function will return the camera intrinsic parameters and the reprojection error obtained from the calibration. The reprojection error should be close to 0 to ensure that the calibration is performed correctly with a high accuracy. Each time the camera is moved, the camera needs to be re-calibrated.

2) Object Calibration: Object calibration is used to determine the relative pose of each marker with respect to the reference marker. Assuming that all the markers remain fixed during the experiment, we first capture multiple images of the tracked object from different view points without moving the camera. Then we apply the algorithm proposed by Sarmadi et al. [22] to find the best transformation from each marker to the reference marker as shown in Fig. 3. Given two different markers i and j, and their transformations to the camera, denoted as T_i^t and T_j^t respectively, we can determine the pair-wise transformation from marker i to marker j at time t as follows:

$$M_{i,j}^t = \left(T_j^t\right)^{-1} T_i^t. \tag{1}$$

As stated in the work of [22], due to noise and the ambiguity problem, more than one transformation per marker in an image is possible. To find the optimal transformation from marker i to marker j, Sarmadi's algorithm considers the transform that minimizes the sum of the distances of one transform to all transforms. Lastly, the algorithm uses graph analysis and constructs a minimum spanning tree of pair-wise transformation to find the optimal relative transformation from each marker to the reference marker from several observations. More detailed description of the algorithm can be found in [22].

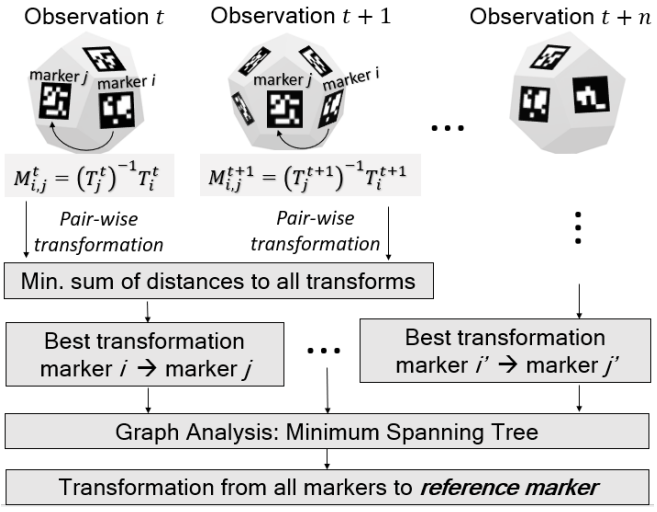


Fig. 3. Overview of Sarmadi's object calibration algorithm used in our proposed system.

C. Object Tracking

The second phase of our proposed system is the object tracking. From each frame of real-time video feed captured from the camera, our system determines the position and orientation of the object. Utilizing the ArUco library [19], our system can detect the square binary markers attached to the object, and get the pose of each marker. This is done by first detecting the four corners of the marker and applying the standard Perspective-n-Point (PnP) algorithm for estimating the marker pose [25]. Using the parameters obtained from camera calibration and the relative transformation from the reference marker, we can determine the relative transformation of the reference marker with respect to the camera as long as at least one marker is detected. The estimated pose of the object is represented as a transformation matrix consisting of R, a 3x3 rotation matrix, and t, a 3x1 translation vector, as follows:

$$\begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2}$$

The Levenberg-Marquardt optimization algorithm [26] is used to minimize the reprojection error, i.e., the total sum of squared distances between the observed points and the projected points.

The final step of object tracking is to apply a Kalman Filter to the estimated result. A Kalman Filter method is used to estimate the state of a linear system where the state is assumed to be distributed by a Gaussian [23]. The Kalman filter is a recursive estimator which uses the estimated state from the previous time step and the measurement of the current time step to estimate the state for the current time step. In our implementation of the Kalman Filter, the system state is stored as a vector composed of the translation vector, t, and rotation vector, r of the object. The state vector is given by:

$$X = \begin{bmatrix} t_x & t_y & t_z & r_x & r_y & r_z \end{bmatrix}^T \tag{3}$$

Our algorithm follows the standard Kalman Filter method which consists of two steps, prediction and correction [27]. Our algorithm first predicts the 3D object pose at each frame, and then refines the prediction every time a new measurement of the object is available. The general time update equations for state prediction are as follows:

$$\bar{X_k} = A * X_{k-1} + w_{k-1}, \tag{4}$$

$$\bar{P_k} = A * P_{k-1} * A^T + Q_{k-1}. \tag{5}$$

where $\bar{X_k}$ is the predicted state at time k, X_{k-1} is the vector representing the state at time $k-1$, A is a transition matrix from the state at the previous time step $k-1$ to the state at the current time step k, and w is a random variable representing the noise in the process. $\bar{P_{k-1}}$ is a matrix representing the prediction error covariance at time $k-1$, and Q is the noise.

After predicting the state X_k and its error covariance, the Kalman Filter uses the measurement to correct the prediction.

$$K_k = P_k * H^T (H * P_k * H^T + R)^{-1}, \tag{6}$$

$$X_k = \bar{X}_k + K_k * (Z_k - H * \bar{X}_k), \tag{7}$$

$$P_k = (I - K_k * H) * \bar{P}_k. \tag{8}$$

In Eq. (6), K_k is the Kalman gain, H is a matrix that transforms the state space into the measurement space, R is the measurement noise. The actual state X_k is updated using the predicted state $\bar{X}_k$, the Kalman gain K_k, and a measurement Z_k of the state in Eq. (7). Z_k is the translation vector and rotation vector of the object in the frame. The process is repeated to predict the object state for the next time step. The Kalman Filter is used for smoothing out the motion by avoiding/filtering cluttering noise. In the case of no detection, the Kalman Filter can predict the marker position and orientation for a short interval based on the previous captured data.

IV. Results and Evaluations

In this section, we describe the results of our experiments conducted to evaluate the proposed method. We evaluate the accuracy of our object tracking method by directly comparing it with both ground-truth data obtained from the OptiTrack motion capture system [28] and the existing surgical instrument tracking method by Zhang et al. [10].

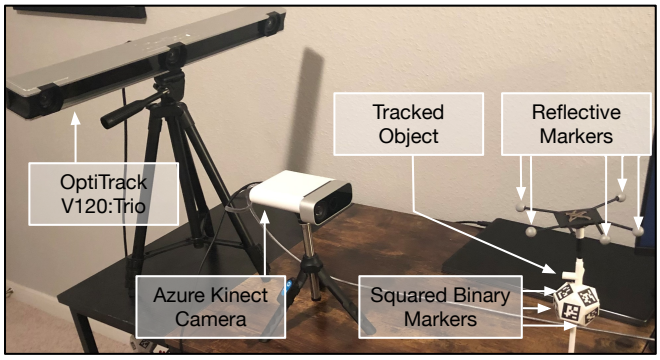


Fig. 4. The experimental setup consists of an OptiTrack V120:Trio system to track reflective markers and an Azure Kinect camera to track the square binary markers attached to the tracked object (i.e., 3D-printed Dodecahedron Pen).

A. Experiment Design

Our system ran on a desktop computer with 2.5 GHz CPU and 32 GB RAM. We use an Azure Kinect Sensor with a color camera resolution of 1280 x 720 and a frame rate of 30 FPS. In addition, an OptiTrack V120:Trio motion capture system with three cameras is used to acquire the ground truth motion of the object (The experimental setup is illustrated in Fig. 4). The motion capture system can track the positions of an object by capturing the pose of reflective markers attached to the object. Five reflective markers are attached to the object using a plastic rigid body attachment as shown in Fig. 5.

The OptiTrack bar and the Azure Kinect camera are positioned in parallel with one another and with the capture volume. Since the distance from the camera to the object can affect the accuracy of the tracking results, we set the volume of the tracking area to be within 0.5 meter from the camera.

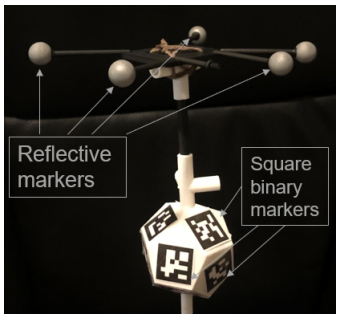


Fig. 5. Tracked object with attached reflective markers and square binary markers

While doing experiments, the object is moved freely within the volume.

B. Experimental Results

To evaluate the proposed approach, we measured the accuracy of the pose estimation of the object. In order to quantitatively validate the pose estimation error, we used the OptiTrack system as shown in Fig. 4 to obtain the ground truth. For experimental validation, we ran eight trials at different distances from the camera to the object: 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, and 0.5 meters. For each trial, a total of 5 captures were made to obtain the ground truths and the estimated results by our approach. The translation error, rotation error, and computational time were computed for each capture. To compute these errors, we first aligned the trajectories obtained from our approach and the ground truth using the Umeyama alignment method [29]. Then the translation error of the estimated pose is computed using the Euclidean distance of 3D measurements given by:

$$ERR_T = \|\bar{t} - t\|, \tag{9}$$

where $\bar{t}$ is the ground truth translation vector, and t is the estimated translation vector.

The rotational error of the estimated pose is computed by:

$$ERR_R = arccos\left(\frac{Tr(R^T \cdot \bar{R}) - 1}{2}\right), \tag{10}$$

where R is the estimated rotation matrix, $\bar{R}$ is the ground truth rotation matrix, *arccos* is the arccosine function in degrees, and Tr represents the trace of the rotation matrix, i.e. the sum of the diagonal elements of the rotation matrix.

The obtained results are shown in Fig. 6. The top graph shows the mean translation errors (in millimeters) of our proposed approach at eight different camera distances. The middle graph shows the average rotation errors (in degrees). In general, our approach can achieve a good estimation within the distance of 0.5m from the camera. The mean errors at the distance ranging from 0.2m to 0.4m are consistently low. The mean errors increase towards the two ends of the graph, with the highest errors at 0.15m and 0.5m from the camera. Our experiment results indicate that the distance between the object and the camera plays a role in the accuracy of the tracking task. That is, if the object is too

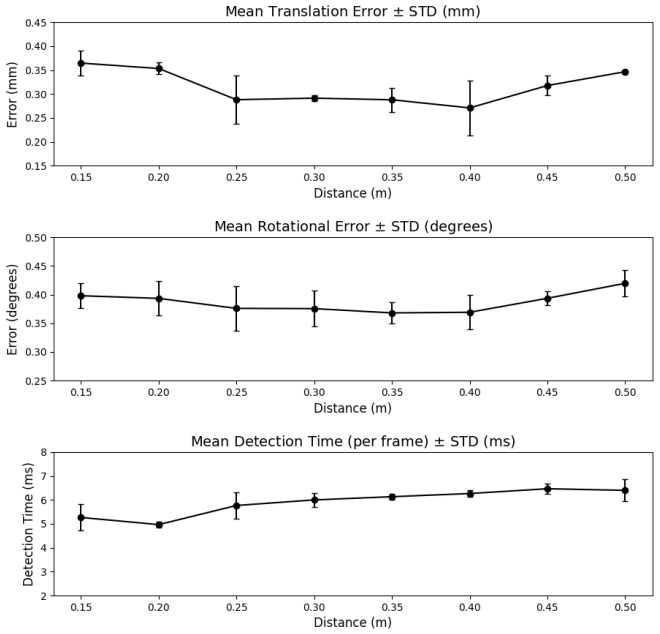


Fig. 6. The mean pose estimation errors and the average runtime of the proposed method with different distances from the camera

close to the camera or too far away from the camera, our approach may not be able to achieve the optimal solution.

The bottom graph in Fig. 6 shows the average detection time (in milliseconds per frame) of our approach. Since the initial calibration is performed offline once, we only concern about the execution time during continuous tracking. In this context, tracking task consists of detecting markers in the frame and estimating the object pose. On average, the object pose estimation takes approximately 6 ms per frame.

In Fig. 7, we show the tracking results of our proposed approach in one of the capture sessions in comparison with the corresponding ground truth data. The ground truth trajectories of the object are depicted in orange color, while the estimated trajectories are depicted in blue color. As shown in Fig. 7, we can observe that overall the tracked results are reasonably close to the ground truth; but at some parts, there are visible differences between the estimated results and the ground truth due to occlusions and motion blur. For instance, in the Z-axis of the rotation graph, ground-truth trajectory drops from 200° to -200° at frame 42, which is the result of a fast object motion. OptiTrack system running at 120 FPS was able to capture such rapid motion, however, our system, capturing frames at 30 FPS, was unable to response quick enough to the sudden changes. Our system estimates the motion trajectory of the object based on the previous data, thus we see gradual changes rather than a slope.

In MIS, the surgical instruments' motions are restricted due to kinematic constraints at incision points. Thus, we implemented these motions such that the incision point is fixed and unchanged for the entire duration of the experiment. The tracked object controls the pose of the tooltips. For our experiment, we focused on the laparoscopic tooltip which involved 3 degree-of-freedom for translation and 1 degree-

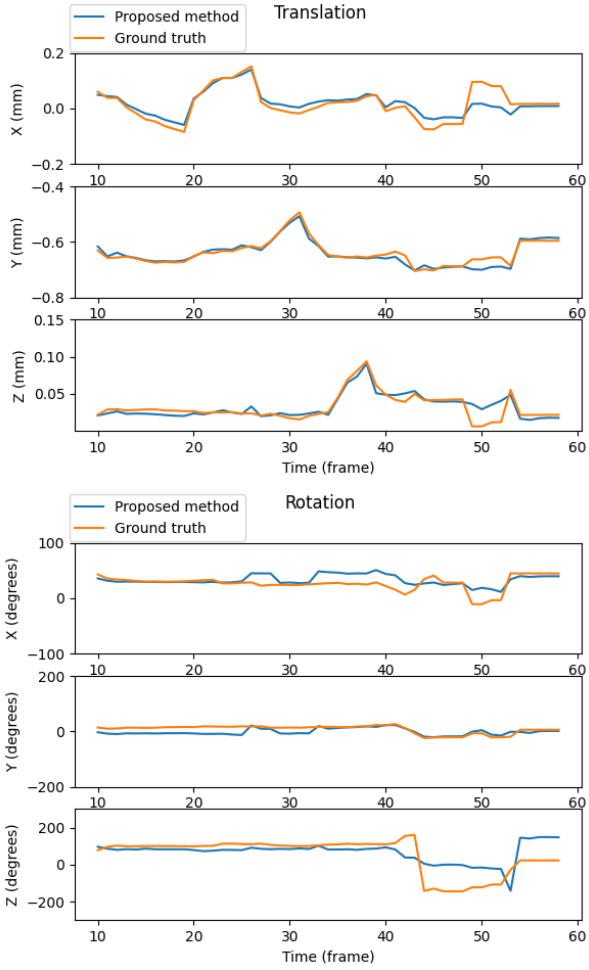
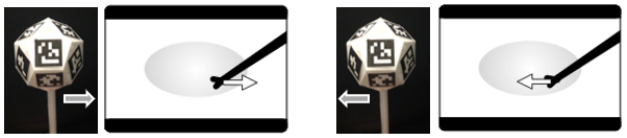


Fig. 7. Comparison between the tracked results and the ground-truth data

of-freedom for rotation. Fig. 8 shows the integrated motion of the proposed interfacing object to control the virtual surgical instrument. For example, when the user moves the tracked object along the x-axis (left/right), the virtual surgical tooltip moves from the left side to the right side of the screen accordingly following the object motion. Similarly, motion control along the Y axis or Z axis is also illustrated in the figure. When the user rotates the interfacing object along the Y axis, the virtual surgical tool rotates along its y-axis on the screen.

C. Comparison with Existing Methods

We also evaluated our approach by directly comparing it with a recent, most related, surgical instrument tracking methods proposed by Zhang et al. [10]. Their method uses a hybrid marker design that combines circular dots and chessboard vertices to track cylindrical surgical tools. It is noteworthy that, while the purposes of both our approach and their approach are the same (that is, to improve the visualization of surgical tool tracking in minimally invasive surgery), their specific tasks and environments have some differences. Our method is designed for a tele-surgery mentoring framework that requires little setup by the expert



(a) Translation along z-axis. Left/Right motion control of object.

(b) Translation along y-axis. Up/Down motion control of object.

(c) Translation along z-axis. Forward/Backward motion control of object.

(d) Rotate motion control of object.

Fig. 8. Motion of the proposed interfacing object and the controlled motion of the virtual surgical instrument

surgeon in the remote location (i.e. no surgical setup or surgical instruments is required). Whereas, their method [10] is primarily designed for use in a surgical environment with a monocular laparoscope and various types of surgical tools such as needle driver, and cautery. Thus, this comparison may not be perfectly fair, but their method is the most related, recent work for comparison with our method, to the best of our knowledge.

Specifically, we modified the work space to fit our comparison experiment. The cylindrical object used for tracking was a pen with the diameter of 6 mm. We used the same calibrated camera parameters from the above experiments for both methods to ensure a consistent analysis. Our comparison experiment adhered to the work space of Zhang et al.'s method [10], that is, the distance from the interfacing object to the camera must be within the range from 30 to 125 mm. Moreover, the method of [10] limits the rotational motion of the markers. The rotation around the roll, pitch, and yaw axes is up to $\pm 68^\circ$, $\pm 56^\circ$, and $\pm 81^\circ$, respectively. The results of the comparison experiment are reported in Table I. Our approach can outperform [10] in term of accuracy. More importantly, as shown in this table, our method is an order of magnitude faster than [10] in term of tracking speed. Lastly but not least, our method does not limit the range of rotational and translation motion, thus providing a larger work space than [10].

D. Limitations

Despite the validated accuracy and efficiency, our current method has the following limitations. First, the tracking accuracy of our current approach is susceptible to occlusions and motion blur which occurs when the object is moved too fast. Tracking failure could happen if markers failed to be detected due to varied lighting conditions. Also, the distance between the object and the camera has a significant effect on the accuracy of pose estimation. The movement of the object is confined in a volume within 0.5 meter from the camera to obtain the best accuracy. If the object is too close to the camera or too far from the camera, the accuracy of pose estimation tasks degrades. This problem can be mitigated by the use of additional cameras as reported in [22]. Second, the initial calibration step is done offline and needs to be performed again if the camera position/orientation is changed. A potential improvement of our current approach is to include simultaneous camera calibration as the markers are tracked. Also, tracking of the surgical tooltip controller is unavailable in our current approach. As the future work, we plan to add this feature to enhance the applicability of our approach in clinical use. Third, the experiment performed mainly to evaluate the technical performance of our method in term of accuracy and efficiency. Additional user study are required to evaluate the method in term of usability and ergonomics requirements of the surgeon. The end-user studies will assess the clinical usage of the software [30], [31]. Finally, while our initial implementation assigns a fixed incision point and the scope camera pose to render the motion of the virtual surgical instrument, these values would need to be real-time tracked intra-operatively by using an external tracking system [32]. The angulation and articulation of the scope would also need to be incorporated [33], [34] .

TABLE I

POSE ESTIMATION ERROR USING DIFFERENT METHODS

	Our approach	**Zhang *et al.***
Translation error (mm)	0.374	0.376
Rotational error (degrees)	0.58	0.603
Detection time (ms per frame)	4.75	94.53

V. CONCLUSION

In this paper, we present a tracking interface for controlling motion of virtual minimally invasive surgical instruments. The proposed method starts with an offline initial calibration of the camera and the interfacing object. Then, at runtime it can real-time track and estimate the poses of the interfacing object based on the video feed from a single camera. Several experiments have been conducted to evaluate the accuracy and efficiency of our method. Our experimental results show that our method can achieve sub-millimeter translation accuracy and sub-degree rotation accuracy in object pose estimation when comparing it with the ground truth obtained from a motion capture system. One of the potential application for the proposed interfacing mechanism is surgical tele-mentoring [4]. As a part of future work, we plan use the interfaces with mixed reality devices to overlay and control virtual surgical tools in open surgeries and image-guided interventions [35], [36].

REFERENCES

[1] A. M. Vera, M. Russo, A. Mohsin, and S. Tsuda, "Augmented reality telementoring (ART) platform: a randomized controlled trial to assess the efficacy of a new surgical education technology," *Surg Endosc*, vol. 28, no. 12, pp. 3467–3472, Dec 2014.

[2] A. M. Jarc, A. A. Stanley, T. Clifford, I. S. Gill, and A. J. Hung, "Proctors exploit three-dimensional ghost tools during clinical-like training scenarios: a preliminary study," *World J Urol*, vol. 35, no. 6, pp. 957–965, Jun 2017.

[3] A. M. Jarc, S. H. Shah, T. Adebar, E. Hwang, M. Aron, I. S. Gill, and A. J. Hung, "Beyond 2D telestration: an evaluation of novel proctoring tools for robot-assisted minimally invasive surgery," *J Robot Surg*, vol. 10, no. 2, pp. 103–109, Jun 2016.

[4] D. Shabir, N. Abdurahiman, J. Padhan, M. Trinh, S. Balakrishnan, M. Kurer, O. Ali, A. Al-Ansari, E. Yaacoub, Z. Deng, A. Erbad, A. Mohammed, and N. V. Navkar, "Towards development of a telementoring framework for minimally invasive surgeries," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 17, p. e2305, 2021.

[5] Y. Zhang, X. Shen, and Y. Hu, "Face registration and surgical instrument tracking for image-guided surgical navigation," in *2016 International Conference on Virtual Reality and Visualization (ICVRV)*, 2016, pp. 65–71.

[6] S. C. Lee, B. Fuerst, K. Tateno, A. Johnson, J. Fotouhi, G. Osgood, F. Tombari, and N. Navab, "Multi-modal imaging, model-based tracking and mixed reality visualisation for orthopaedic surgery," *Healthcare Technology Letters*, pp. 168–173, 2017.

[7] M. Wieczorek, A. Aichert, O. Kutter, C. Bichlmeier, J. Landes, S.-M. Heining, E. Euler, and N. Navab, "Gpu-accelerated rendering for medical augmented reality in minimally-invasive procedures," *CEUR Workshop Proceedings*, vol. 574, July 2010.

[8] P. Mountney, B. Lo, S. Thiemjarus, D. Stoyanov, and G. Zhong-Yang, "A probabilistic framework for tracking deformable soft tissue in minimally invasive surgery," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2007*, 2007, pp. 34–41.

[9] Z. Chiu, A. Z. Liao, F. Richter, B. Johnson, and M. C. Yip, "Markerless suture needle 6d pose tracking with robust uncertainty estimation for autonomous minimally invasive robotic surgery," *CoRR*, vol. abs/2109.12722, Sep 2021.

[10] L. Zhang, M. Ye, P.-L. Chan, and G.-Z. Yang, "Real-time surgical tool tracking and pose estimation using a hybrid cylindrical marker," *International Journal of Computer Assisted Radiology and Surgery*, vol. 12, pp. 1–10, 2017.

[11] B. Huang, Y.-Y. Tsai, J. Cartucho, K. Vyas, D. Tuch, S. Giannarou, and D. Elson, "Tracking and visualization of the sensing area for a tethered laparoscopic gamma probe," *International Journal of Computer Assisted Radiology and Surgery*, pp. 1389–1397, 2020.

[12] J. Cartucho, C. Wang, B. Huang, D. Elson, A. Darzi, and S. Giannarou, "An enhanced marker pattern that achieves improved accuracy in surgical tool tracking," *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, pp. 1–9, 2021.

[13] F. Ababsa and M. Mallem, "Robust camera pose estimation using 2d fiducials tracking for real-time augmented reality systems," in *Proceedings of the 2004 ACM SIGGRAPH International Conference on Virtual Reality Continuum and Its Applications in Industry*, 2004, p. 431–435.

[14] B. Atcheson, F. Heide, and W. Heidrich, "Caltag: High precision fiducial markers for camera calibration," in *Vision, Modeling, and Visualization (2010)*, 2010.

[15] D. Tang, T. J. Hu, L. Shen, M. Zhaowei, and C. Pan, "Apriltag array-aided extrinsic calibration of camera–laser multi-sensor system," *Robotics and Biomimetics*, 2016.

[16] J. Bačík, F. Durovsky, P. Fedor, and D. Perdukova, "Autonomous flying with quadrocopter using fuzzy control and aruco markers," *Intelligent Service Robotics*, vol. 10, pp. 185–194, 2017.

[17] Z. Li, Y. Chen, H. Lu, H. Wu, and L. Cheng, "Uav autonomous landing technology based on apriltags vision positioning algorithm," in *2019 Chinese Control Conference (CCC)*, 2019, pp. 8148–8153.

[18] H. Sarmadi, R. Muñoz-Salinas, M. Álvaro Berbís, A. Luna, and R. Medina-Carnicer, "Joint scene and object tracking for cost-effective augmented reality guided patient positioning in radiation therapy," *Computer Methods and Programs in Biomedicine*, vol. 209, p. 106296, 2021.

[19] S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas, and M. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, pp. 2280–2292, 2014.

[20] M. Kalaitzakis, B. Cain, S. Carroll, A. Ambrosi, C. Whitehead, and N. Vitzilaios, "Fiducial markers for pose estimation," *Journal of Intelligent & Robotic Systems*, 2021.

[21] P. Wu, R. Wang, K. Kin, C. Twigg, S. Han, M. Yang, and S. Chien, "Dodecapen: Accurate 6dof tracking of a passive stylus," in *UIST 2017 - Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, 2017, pp. 365–374.

[22] H. Sarmadi, R. Muñoz-Salinas, M. A. Berbís, and R. Medina-Carnicer, "Simultaneous multi-view camera pose estimation and object tracking with squared planar markers," *IEEE Access*, vol. 7, pp. 22 927–22 940, 2019.

[23] P. R. Gunjal, B. R. Gunjal, H. A. Shinde, S. M. Vanam, and S. S. Aher, "Moving object tracking using kalman filter," in *2018 International Conference On Advances in Communication and Computing Technology (ICACCT)*, 2018, pp. 544–547.

[24] OpenCV. Calibration with aruco and charuco. [Online]. Available: https://docs.opencv.org/3.4.15/da/d13/tutorial_aruco_calibration.html

[25] ——. Detection of aruco markers. [Online]. Available: https://docs.opencv.org/3.4.15/d5/dae/tutorial_aruco_detection.html

[26] A. Ranganathan, "The levenberg-marquardt algorithm," *Tutoral on LM algorithm*, vol. 11, no. 1, pp. 101–110, 2004.

[27] G. Welch and G. Bishop, "An introduction to the kalman filter," *Proc. Siggraph Course*, vol. 8, 01 2006.

[28] NaturalPoint. Optitrack website. [Online]. Available: https://optitrack.com/

[29] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 4, pp. 376–380, 1991.

[30] J. D. Velazco-Garcia, N. V. Navkar, S. Balakrishnan, J. Abinahed, A. Al-Ansari, A. Darweesh, K. Al-Rumaihi, E. G. Christoforou, E. L. Leiss, M. Karkoub, P. Tsiamyrtzis, and N. V. Tsekos, "Evaluation of interventional planning software features for mr-guided transrectal prostate biopsies," in *2020 IEEE 20th International Conference on Bioinformatics and Bioengineering (BIBE)*, 2020, pp. 951–954.

[31] J. D. Velazco-Garcia, N. V. Navkar, S. Balakrishnan, J. Abi-Nahed, K. Al-Rumaihi, A. Darweesh, A. Al-Ansari, E. G. Christoforou, M. Karkoub, E. L. Leiss, P. Tsiamyrtzis, and N. V. Tsekos, "End-user evaluation of software-generated intervention planning environment for transrectal magnetic resonance-guided prostate biopsies," *Int J Med Robot*, vol. 17, no. 1, pp. 1–12, Feb 2021.

[32] M. K. Chmarra, C. A. Grimbergen, and J. Dankelman, "Systems for tracking minimally invasive surgical instruments," *Minim Invasive Ther Allied Technol*, vol. 16, no. 6, pp. 328–340, 2007.

[33] C. A. Velasquez, N. V. Navkar, A. Alsaied, S. Balakrishnan, J. Abinahed, A. A. Al-Ansari, and W. Jong Yoon, "Preliminary design of an actuated imaging probe for generation of additional visual cues in a robotic surgery," *Surg Endosc*, vol. 30, no. 6, pp. 2641–2648, 06 2016.

[34] B. Seeliger, M. Diana, J. P. Ruurda, K. M. Konstantinidis, J. Marescaux, and L. L. Swanström, "Enabling single-site laparoscopy: the SPORT platform," *Surg Endosc*, vol. 33, no. 11, pp. 3696–3703, 11 2019.

[35] C. M. Morales Mojica, J. D. Velazco Garcia, N. V. Navkar, S. Balakrishnan, J. Abinahed, W. El Ansari, K. Al-Rumaihi, A. Darweesh, A. Al-Ansari, M. Gharib, M. Karkoub, E. L. Leiss, I. Seimenis, and N. V. Tsekos, "A Prototype Holographic Augmented Reality Interface for Image-Guided Prostate Cancer Interventions," in *Eurographics Workshop on Visual Computing for Biology and Medicine*. The Eurographics Association, 2018.

[36] J. D. Velazco-Garcia, N. V. Navkar, S. Balakrishnan, G. Younes, J. Abi-Nahed, K. Al-Rumaihi, A. Darweesh, M. S. M. Elakkad, A. Al-Ansari, E. G. Christoforou, M. Karkoub, E. L. Leiss, P. Tsiamyrtzis, and N. V. Tsekos, "Evaluation of how users interface with holographic augmented reality surgical scenes: Interactive planning MR-Guided prostate biopsies," *Int J Med Robot*, vol. 17, no. 5, p. e2290, Oct 2021.