An Augmented Reality Approach to Surgical Telementoring

Timo Loescher
School of Industrial Engineering
Purdue University
West Lafayette, IN
tloesche@purdue.edu

Shih Yu Lee
School of Electrical
and Computer Engineering
Purdue University
West Lafayette, IN
lee1549@purdue.edu

Juan P. Wachs*
School of Industrial Engineering
Purdue University
West Lafayette, IN
jpwachs@purdue.edu

***Abstract*—Optimal surgery and trauma treatment integrates different surgical skills frequently unavailable in rural/field hospitals. Telementoring can provide the missing expertise, but current systems require the trainee to focus on a nearby telestrator, fail to illustrate coming surgical steps, and give the mentor an incomplete picture of the ongoing surgery. A new telementoring system is presented that utilizes augmented reality to enhance the sense of co-presence. The system allows a mentor to add annotations to be displayed for a mentee during surgery. The annotations are displayed on a tablet held between the mentee and the surgical site as a heads-up display. As it moves, the system uses computer vision algorithms to track and align the annotations with the surgical region. Tracking is achieved through feature matching. To assess its performance, comparisons are made between SURF and SIFT detector, brute force and FLANN matchers, and hessian blob thresholds. The results show that the combination of a FLANN matcher and a SURF detector with a 1500 hessian threshold can optimize this system across scenarios of tablet movement and occlusion.**

***Keywords*—telementoring; object-tracking; surgery; augmented reality**

I. Introduction

Telementoring systems benefit surgeons and medics by providing assistance from experienced mentors who are geographically separated [1]–[5]. In such systems, a remotely located mentor instructs a trainee or mentee surgeon through a surgical procedure through visual and verbal cues. The most rudimentary way to implement such a system is by using phones as a connection bridge to have the mentor verbally instruct the mentee [6]. The main limitation of using only verbal communication is that such a system limits the ability of the mentor and the mentee to share visual information. This information sharing is key to the completion of the procedure. Indicating the correct position of incisions and the placement of other surgical instruments allows for a more natural form of communication. Both visual and spoken interaction is necessary in the context of surgery. However, the flow of the surgery should not be interrupted by the surgeon's interaction with the system or focus shifts caused by the system. For this reason, obtrusive interfaces based on telestration are not suitable [7]. This paper discusses a system that offers:

1) An augmented reality interface for the mentee which displays the mentor's annotations in near real-time.
2) An algorithm to track and update the annotations on the patient's anatomy throughout the surgery.

The rest of the paper is organized as follows: first, the background of telementoring is presented along with the main gaps that this technology currently faces. Next, the architecture of the mentor/mentee system is discussed. Then, an evaluation of the core set of feature trackers and tracking accuracy performance is presented. Finally, the implications of the results are presented and the paper concludes with a summary and a discussion of directions for future work.

II. Related work

Telementoring is described as the assistance of one or more mentors on a task through verbal, tactile, and visual cues from a remote location. This remote instruction is commonly used in training and educational environments [8]–[10]. One area where recent focus has shifted regarding telementoring is in healthcare, specifically in surgical operating environments. It is not uncommon that surgeons are in scenarios when they could benefit from a subspecialist's expertise. Research has shown the benefits of the visual access to and of the remote proctoring of surgeries [1]–[3], as well as the potential for telementoring to improve minimally invasive surgery through remote video-assisted instruction [4], [5].

A newer branch for telementoring in surgery regards the utilization of visual assistance. Dixon et al. [11] looked at the effects of augmented reality on telementoring success with regard to visual attention. This discovery showed that introducing annotations such as anatomical contours to endoscopic surgeons improved accuracy, albeit at a cost to cognitive attentional resources. As this paper continues, we use the term augmented reality as defined by Augestad et al. to describe ”the addition of annotations to a viewport to augment the viewer's visual information” [12].

Augmented reality allows for the real time observation of desired data or critical information in a three-dimensional environment without task interruption. In the context of surgery, this includes the monitoring of vitals and deliberation with radiological scans during an operation, both of which distract the surgeon from the primary task [13], [14]. Data has shown that such distraction in surgical tasks can be common and

This work was supported by the Office of the Assistant Secretary of Defense for Health Affairs under Award No. W81XWH-14-1-0042. Opinions, interpretations, conclusions and recommendations are those of the author and are not necessarily endorsed by the Department of Defense.

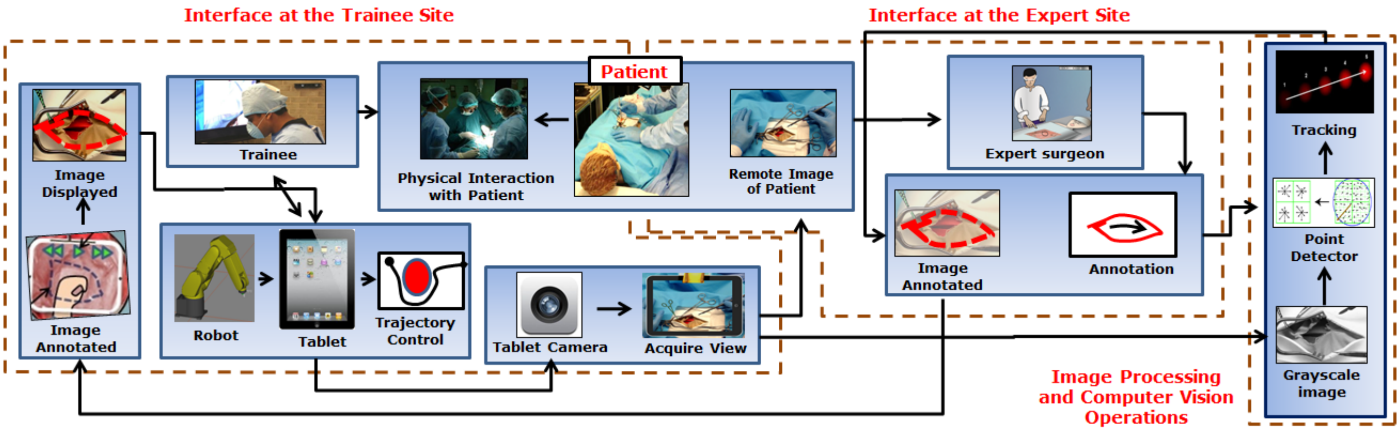


Fig. 1: System architecture

may lead to detrimental effects [15]. Even though augmented reality can bring positive assistance to surgical sites, recent research showed a disconnect from the surgical workflow due to obstacles presented by the implementations. The systems are often only beneficial for planning purposes, as they are bulky [14], their displays do not adjust over time with the region of interest [16], or the system runs too slowly to be of any use [17]. Recently, a number of applications have been developed for tablet usage during surgery [18]–[23]. They have been implemented in the context of education, navigation, and image reconstruction. However, none of these applications were used for telementoring. This paper presents a surgical telementoring system that addresses these issues. The following section discusses the design of a tablet-computer system and the evaluation of its algorithmic parameters.

III. Methods

A. System Design

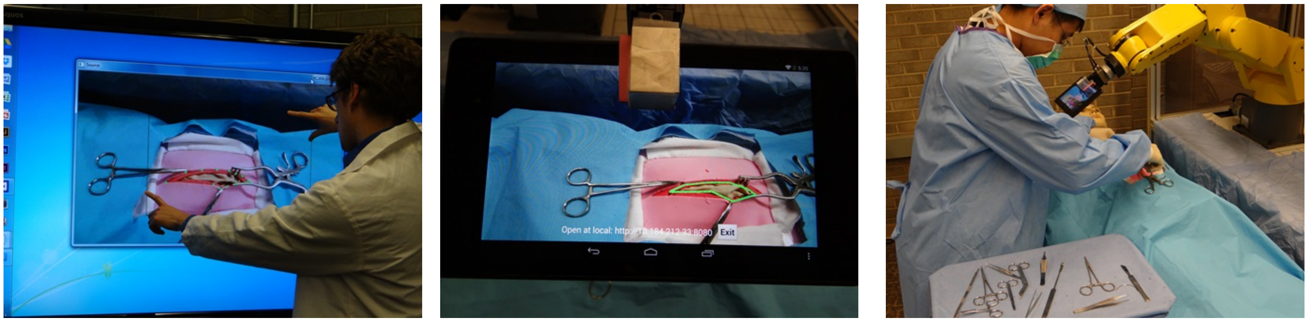
The architecture of the designed system is shown in Fig. 1. Physically, the surgical mentee stands at the patient's side when performing surgery. The mentee looks *through* a tablet screen at a real-time video feed from the rear facing camera directed at the surgical site. This affords the sense of looking through a window at the patient. The tablet is held by a robotic arm to allow the physician to move around without compromising his field of view. If the surgeon just wants a to take a glance a technologist or assistant can manually hold the tablet in his/her field of view; however, the human hand is more prone to movements and therefore may have adverse effects on the system's tracking.

At the other remote site, a mentor, who is another surgeon, is accessing the surgical view from the Internet delivered by the camera on the tablet remotely. The mentor's computer displays the video feed from the tablet for monitoring purposes. When the mentee needs guidance, the off-site mentor selects a surgical region from the tablet feed and annotates the image, as shown in Fig. 2a. These annotations might conceptually take the form of text strings, sketches, radiology imaging overlays, or locational highlighting for tool placement. In this system, adding the annotations consists of creating a polygon on the surgical region, as well as adding strings of text (e.g. ”incision”, ”closure”). As soon as the region is selected by the mentor, the mentor's host computer immediately begins detecting and tracking that region in incoming images. As the annotations are completed, the mentee surgeon can see the mentor's notes on the annotated window the tablet provides (Fig. 2b). Then, he/she can use those annotations by looking through the tablet while working, as displayed in Fig. 2c. This continues until the mentee no longer needs the annotations, at which point the annotations can be deleted for a clear viewing pane. The main two components of the interface consist of:

- Mentee Side (Tablet): This is treated as an end user interface, and no image processing computations occur on this device. The tablet is the key tool to show and fetch the image at the front end as well as a communication interface between mentee and mentor. It also operates as the server for connection purposes.
- Mentor Side (PC): This is where the main software for processing the detection, annotation, and posting to the tablet resides. The software interface has the following functions: crop a region, create an annotation, track the region, and send the calculated annotation positions to the tablet.

The challenge of running at near-real-time is solved by a three thread parallel computing architecture. The first thread serves to pull in the video frames from the tablet so that the mentor has a clean feed as well as to facilitate inter-thread communication. The second thread handles the bulk of the calculations. It is in this thread that the processing algorithms work to detect, match, and translate points. The third thread is responsible for the communication with the tablet to ensure the most up to date annotations are displayed.

As the majority of the computational load exists in the second thread, this paper focuses on the algorithms of this thread. When the mentor selects a template, the system automatically detects the features in the template image. The locations of those template features are saved as T along with the annotation points (A) made on the template image. Then, for each iteration of the computational thread, a frame has its feature points likewise detected and stored in S - a second keypoint array. Algorithm 1 shows how each of the sets are compared to find matching sets between the two keypoint arrays. This algorithm results in an array M of matching



(a) The mentor annotating points to be displayed

(b) The tablet displaying the annotated field of view

(c) The mentee looking through the tablet at the annotated surgical site

Fig. 2: The developed system

Algorithm 1 Template and scene keypoint matching

```
Annotation points: $A = \{(x_{ak}, y_{ak})\}, k \in [1, v]$
Template feature points: $T = \{(x_{ti}, y_{ti}, f_{ti})\}, i \in [1, m]$
Scene feature points: $S = \{(x_{sj}, y_{sj}, f_{sj})\}, j \in [1, n]$
for $i \in [1, m]$ do
    for $j \in [2, n]$ do
        if $f_{ti} \cong f_{sj}$ then
            $q \leftarrow (i, j)$
        end if
    end for
    if $q$ exists then
        $M \leftarrow q$
    end if
end for
```

indexes. Using the set of matches M, along with T and S, Algorithm 2 finds the changes in pan shift, rotation, and scale. For each cloud of matched keypoints, the distances between every point pair (D_T and D_S) and the difference in angles between each corresponding point pair across (θ) is determined. The ratio of sizes comes from the median distances in D_T and D_S. The system then finds the centroids of each of the matched points clouds. All these values are used to find the projection locations of the annotations (P) by applying Equation (1) to each of k annotation points.

$$P_k = \left(\hat{x}_s - \frac{\cos(\alpha)(-x_{ak} + x_c + \hat{x}_t)}{r} + \frac{\sin(\alpha)(-y_{ak} + y_c + \hat{y}_t)}{r}\right), \left(\hat{y}_s - \frac{\sin(\alpha)(-x_{ak} + x_c + \hat{x}_t)}{r} + \frac{\cos(\alpha)(-y_{ak} + y_c + \hat{y}_t)}{r}\right) \tag{1}$$

Feature detection was chosen over tracking due to the massive amounts of occlusion surrounding the key features in a surgical context. Frame-by-frame trackers such as Lucas-Kanade lose or misinterpret tracking points too quickly to be useful in this system. As another disregarded option, template matching constrains the detections to replicas of the template image. However, continuous feature detection allows for template matching without perfect information and scene changes, and is robust during and after occlusion. This makes it an optimal choice for our surgical context.

On the client side where the server runs, the client receives a sequence of data through an *http* form for communication. For every post action, values are received from the tablet to instruct how the annotation string must be decoded. From this, the sequence of points is extracted, and the desired overlay information is re-rendered on the current view at the mentee side generating the augmented reality.

Algorithm 2 Extracting parameters for projection

```
Top-left crop point for template: $(x_c, y_c)$
for $(i, j) \in M$ do
    for $(i, j) \in M$ where $index(\hat{i}, \hat{j}) > index(i, j)$ do
        $D_T \leftarrow \sqrt{(x_{ti} - x_{t\hat{i}})^2 + (y_{ti} - y_{t\hat{i}})^2}$
        $D_S \leftarrow \sqrt{(x_{sj} - x_{s\hat{j}})^2 + (y_{sj} - y_{s\hat{j}})^2}$
        $\theta \leftarrow \tan^{-1}(\frac{x_{ti} - x_{t\hat{i}}}{y_{ti} - y_{t\hat{i}}}) - \tan^{-1}(\frac{x_{sj} - x_{s\hat{j}}}{y_{sj} - y_{s\hat{j}}})$
    end for
end for
$\bar{d}_t \leftarrow median(D_T)$; $\bar{d}_s \leftarrow median(D_S)$
$r \leftarrow \frac{\bar{d}_s}{\bar{d}_t}$
$a \leftarrow median(\theta)$
$\bar{x}_t \leftarrow mean(x_t)$; $\bar{y}_t \leftarrow mean(y_t)$; $\bar{x}_s \leftarrow mean(x_s)$; $\bar{y}_s \leftarrow mean(y_s)$
Project points $\in A$ using Equation (1)
```

B. Evaluation

The crucial aspect of this system relies on tracking precision and annotation frame rate. To assess performance based on these two criteria, two state of the art feature detection algorithms were chosen to perform the tracking: Scale Invariant Feature Transform (SIFT) and Speeded Up Robust Features (SURF). These each take a parameter known as a hessian value that determines how descriptive a given point is. The stronger the point, the greater the hessian; therefore as the threshold goes up, the number of detected points goes down while their strength goes up. In order to perform complete tracking, feature matchers were used to find good matches between the surgical region of interest and the target image view. Thus, a brute force matcher and Fast Library for Approximate Nearest Neighbors (FLANN) matcher were used to determine which settings result in the best performance. The experimental procedure is based on evaluating the combinations of different feature detectors and matchers (see Table I) in different video contexts.

The system's annotation update rate was determined by C/T with $C = 50$ calculated frames and T being measured as the time taken to post those 50 frames. This reflects how

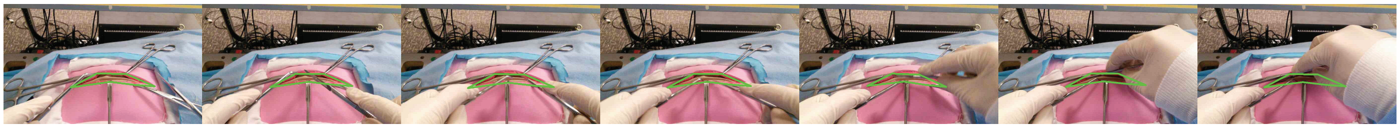


Fig. 3: Two example video strips (three frames apart each) taken for the evaluation

TABLE I: The list of control variables

Variables	Parameters	Values
X_1	Feature Detector	SURF / SIFT
X_2	Matcher	Brute force / FLANN
X_3	Hessian Threshold	0 - 2500 (100 step increments)
X_4	Video Contexts	Stationary, Pan, Zoom, Skew, Minor occlusion, Major occlusion

often mentees has their annotations updated with the most current information. Each 50 frames constituted 1 trial, and 25 trials were collected for each of the combinations of the system parameters: feature detector type, matcher, and hessian threshold (X_1, X_2, and X_3 shown in Table I). The video was held stationary for each of these trials.

The other performance measure studied was tracking accuracy. This was tested with all four tracking parameter combinations, each with the wide range of hessian values (Table I). To test the accuracy, three sequences of videos (a total of 456 frames) were saved and manually annotated using LabelMe [24]. The videos simulated different contextual uses (the X_4 parameter in Table I). These videos were collected from the tablet as a robotic arm held and manipulated its positions in a controlled and pre-programmed fashion. The first video incorporated slow movements (20 *mm/s*) by the robotic arm and conducted panning, zooming, and skewing motions. The second mirrored the first but ran at 50 *mm/s*. The third and final video showed a stationary video with minor occlusion (surgical tools), major occlusion (tools and hands), and no occlusion at all. These image sequences (without annotations) were then fed into the system in lieu of the tablet video stream. Fig. 3 shows such a filmstrip incorporating occlusion. The annotated points were the two edges of a simulated incision and the four surgical tools used to hold open the surgical site. For each frame, the differences in the posted annotation values and the corresponding a priori hand annotation values were squared and averaged to find the Mean Squared Error (MSE) for each frame.

IV. Results

A. Update Rate

The update rate plot of all four algorithms (Fig. 4) is presented as a function of the hessian value of the detector. All four algorithms have similar slopes until the curves change at a 200-300 hessian value. The curves for SIFT decline after reaching that point, while the SURF detectors continue increasing albeit at a slower pace. In addition, it is notable that as the hessian threshold value increases, the SURF detectors' variability increases as well. Interestingly, the SIFT detector's variability remained relatively low compared to the SURF algorithms. Finally, the SURF detectors reach up to around 6 updated frames/sec in sharp contrast the SIFT detectors, which reach a peak around 2.5 updated frames/sec.

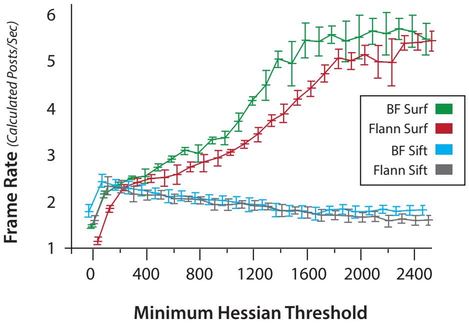


Fig. 4: Experiment 1 - Update rates against different algorithms

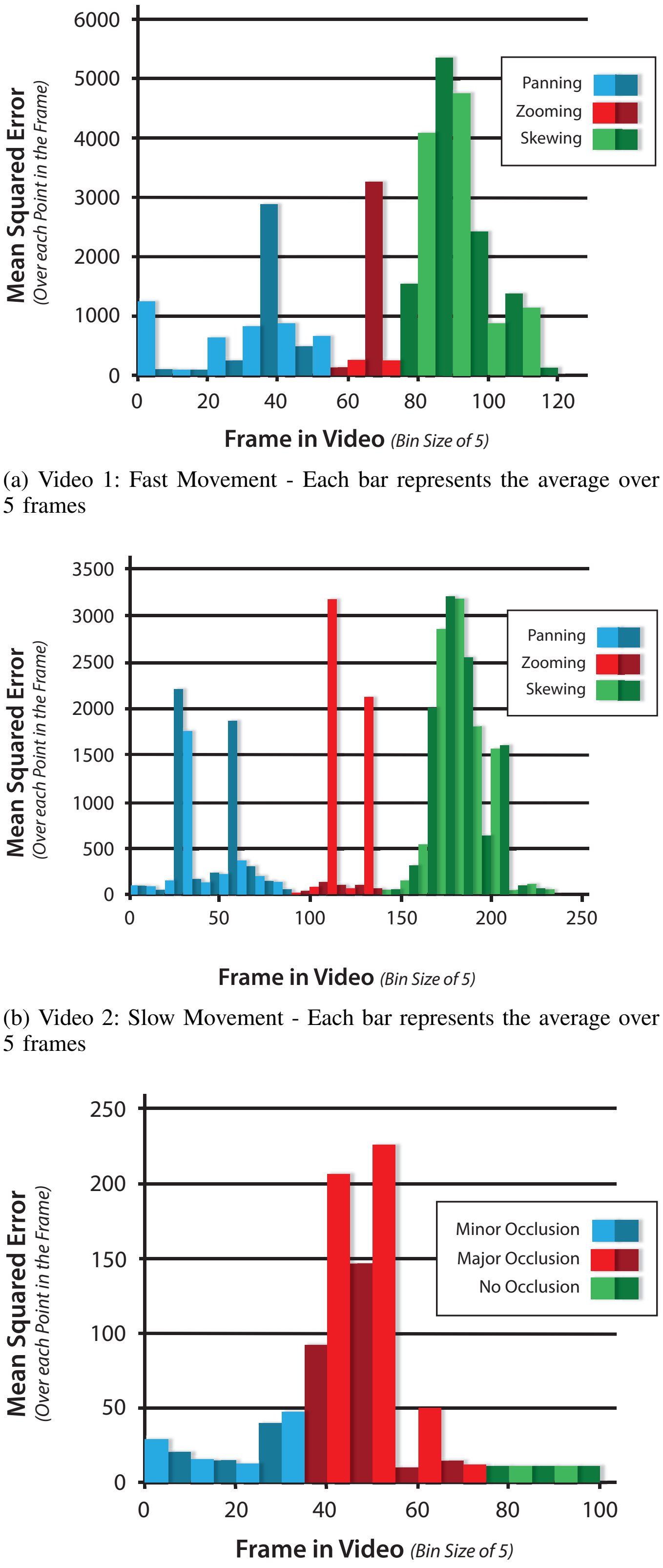
B. Tracking Accuracy

While the data for update rates shows some clear trends, the accuracy data is far noisier. The following three graphs (Fig. 5a, Fig. 5b and Fig. 5c) show the recognition accuracy for each sequence clip according to the best average overall algorithm: a SURF detector at 1500 hessian with a FLANN matcher. The graphs bin 5 frames together to show the average performance trends throughout the video as different tasks were performed. When accuracy was compared between matchers and against hessian thresholds, it was found that SURF and SIFT have means on the same order of magnitude. However, upon further inspection, it was discovered that the high average MSE for the SURF detectors comes from spikes on the frames of incredibly large error lasting a single frame at a time. Anderson-Darling normality tests run on the data found the SURF detectors to be non-normal (p-value > 0.05) while the SIFT detectors were found to be normal (p-value $<= 0.05$). Fig. 6 shows these differences from the means and medians, along with the different standard deviations for the sets.

V. Discussion

In order to find the most adequate algorithm for the telementoring system, the rates of four algorithms have been tested. As shown in Fig. 4, brute force SURF was the fastest among the tested algorithms. The hessian value threshold indirectly influenced the number of points to show on the frame by filtering out poor features. According to this, it is reasonable that matching fewer points was faster than many points. However, the fact that SIFT does not have nearly the increase in update rate seems to confound this logic. In any case, the speed of each algorithm is only important when the algorithm is able to adequately track the target region.

Fortunately, SURF excelled in tracking accuracy beyond SIFT as well. Although less stable with large single-frame errors, the SURF detector-based system corrected itself quickly and showed a much lower median than SIFT. A trade-off between speed and accuracy was expected; however, none



(a) Video 1: Fast Movement - Each bar represents the average over 5 frames

(b) Video 2: Slow Movement - Each bar represents the average over 5 frames

(c) Video 3: Static with Occlusion - Each bar represents the average over 5 frames

Fig. 5: Experiment 2 - Tracking error over various video contexts for an optimal tracker

of the algorithms showed a statistically strong correlation (p-values all > 0.05). It should be noted that for video 3, when the view is stable and only occlusion is applied, the tracking is very accurate (MSE of 25.34 pixels^2 for minor occlusion and 94.87 pixels^2 for major occlusion). It is assumed that the main use scenario would align greater with this video context than the first 2 videos; if the robotic arm was moving, for the surgeon to use the system they would have to be moving with the tablet.

The next step of evaluation is to include human subjects as part of the contextual testing. This will be done with the best parameter combination found in the current study, and usability metrics will be evaluated. In the future, such a study with surgeons will shed light on whether such update rates and accuracies are acceptable for the task. A small limitation comes from the small sample size in experiment 2. In addition, the results would benefit from a wider sampling of real or simulated surgical scenarios (longer videos, different surgical regions and tools, etc.). In terms of stability, the system presented can detect and compensate for movement (skew, and in plane rotation) within a range, however, it works best when stable. Therefore holding the tablet with human hands may have some impact on the tracking accuracy, because humans cannot hold a video perfectly still. This scenario should be tested to assess the extent of the impact of a human holding the tablet. Finally, surgical environments are meant to be sterile. While sterilizing a tablet computer is problematic, placing it in a clear plastic bag may allow acceptable levels of sterility. It is currently unknown to what degree this solution or other similar solutions would impact the integrity of the system's design

This work serves as the base of the system's design, but as time progresses, features to be implemented include body tracking and gestural interaction with the robotic arm to seamlessly integrate the tablet into the environment, the addition of surgical tool image overlays and other annotations beyond point-sketching, and work on the mentor interface to increase the mentor's sense of telepresence. With these additions, the system should become more contextually generalizable. Working with surgeons and training hospitals will help ensure that the features to be added will indeed achieve these goals.

VI. Conclusion

There are many contexts in which surgical telementoring and augmented reality can come together to provide value to patients and physicians alike. The development of such a system is challenging, yet not impossible. In this work, a prototype system was developed and presented, and data on the various parameters that went into the system design were collected. It was found that the tracking module when implemented with SURF was superior to SIFT in speed and accuracy, with an optimal hessian threshold at 1500. Within SURF, it seems that FLANN is slightly more accurate while being slightly slower. While this is contrary to our original ideas, it provides insights into the nature of the matchers in this contexts, and justifies our decisions to test these differing parameters systematically. Going forward, the designed system will continue to be improved and tested with users in surgical contexts of training and consulting.

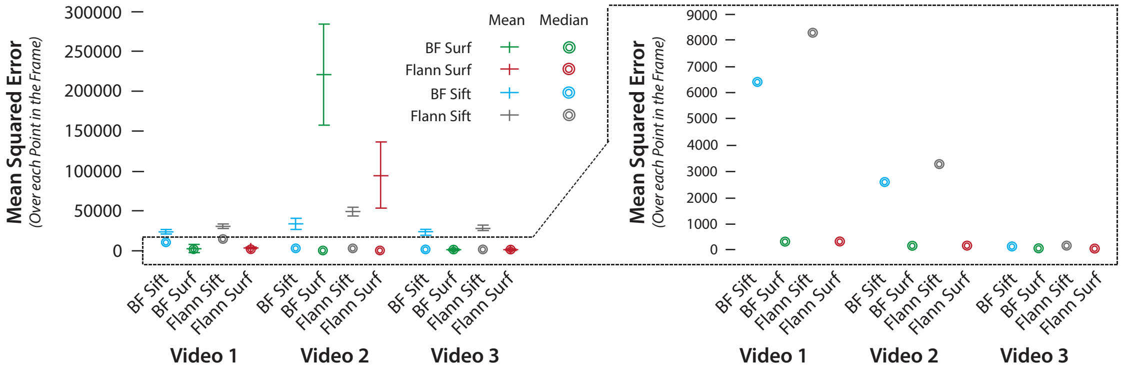


Fig. 6: Experiment 2 - Comparisons between algorithms' mean squared errors (MSE) over all hessian thresholds parameterized by mean and median

Acknowledgments

The two first authors contributed equally to this work. We would like to acknowledge the open source projects without which we could not have accomplished our work: OpenCV, Android-Eye, cURL, and LabelMe.

References

[1] A. Q. Ereso *et al.*, "Live transference of surgical subspecialty skills using telerobotic proctoring to remote general surgeons," *Journal of the American College of Surgeons*, vol. 211, no. 3, pp. 400–411, Sep 2010.

[2] D. Wood, "No surgeon should operate alone: How telementoring could change operations," *Telemedicine and e-Health*, vol. 17, no. 3, pp. 150–152, Apr 2011.

[3] G. H. Ballantyne, "Robotic surgery, telerobotic surgery, telepresence, and telementoring," *Surg Endosc*, vol. 16, no. 10, pp. 1389–1402, Oct 2002.

[4] B. Challacombe, L. Kavoussi, A. Patriciu, D. Stoianovici, and P. Dasgupta, "Technology insight: telementoring and telesurgery in urology," *Nat Clin Pract Urol*, vol. 3, no. 11, pp. 611–617, Nov 2006.

[5] B. R. Lee and R. Moore, "International telementoring: a feasible method of instruction," *World journal of urology*, vol. 18, no. 4, pp. 296–298, 2000.

[6] L. H. Eadie, A. M. Seifalian, and B. R. Davidson, "Telemedicine in surgery," *British Journal of Surgery*, vol. 90, no. 6, pp. 647–658, 2003.

[7] M. B. Shenai *et al.*, "Virtual interactive presence and augmented reality (VIPAR) for remote surgical assistance." *Neurosurgery*, 2011.

[8] J. B. Harris and G. Jones, "A descriptive study of telementoring among students, subject matter experts, and teachers: Message flow and function patterns," *Journal of research on computing in education*, vol. 32, pp. 36–53, 1999.

[9] M. A. Price and H.-H. Chen, "Promises and challenges: Exploring a collaborative telementoring programme in a preservice teacher education programme," *Mentoring and Tutoring*, vol. 11, no. 1, pp. 105–117, 2003.

[10] C. A. Kasprisin, P. B. Single, R. M. Single, and C. B. Muller, "Building a better bridge: Testing e-training to improve e-mentoring programmes in higher education," *Mentoring and Tutoring*, vol. 11, no. 1, pp. 67–78, 2003.

[11] B. J. Dixon *et al.*, "Surgeons blinded by enhanced navigation: the effect of augmented reality on attention," *Surgical endoscopy*, vol. 27, no. 2, pp. 454–461, 2013.

[12] K. M. Augestad *et al.*, "Clinical and educational benefits of surgical telementoring," in *Simulation Training in Laparoscopy and Robotic Surgery*. Springer, Jan 2012, pp. 75–89.

[13] R. Azuma *et al.*, "Recent advances in augmented reality," *IEEE Computer Graphics and Applications*, vol. 21, no. 6, pp. 34–47, Nov 2001.

[14] R. K. Miyake *et al.*, "Vein imaging: a new method of near infrared imaging, where a processed image is projected onto the skin for the enhancement of vein treatment," *Dermatologic surgery*, vol. 32, no. 8, pp. 1031–1038, 2006.

[15] A. N. Healey, C. P. Primus, and M. Koutantji, "Quantifying distraction and interruption in urological surgery," *Qual Saf Health Care*, vol. 16, no. 2, pp. 135–139, Apr 2007.

[16] "FraunhoferMEVIS: Mobile Liver Explorer." [Online]. Available: http://www.mevis.fraunhofer.de/en/solutions/mobile-liver-explorer.html

[17] H. Fuchs *et al.*, "Augmented reality visualization for laparoscopic surgery," in *Medical Image Computing and Computer-Assisted Intervention*. Springer, 1998, pp. 934–943.

[18] P. L. Kubben, "Neurosurgical apps for iPhone, iPod touch, iPad and android," *Surgical neurology international*, vol. 1, no. 1, p. 89, 2010.

[19] T. Eguchi *et al.*, "Three-dimensional imaging navigation during a lung segmentectomy using an iPad," *European Journal of Cardio-Thoracic Surgery*, vol. 41, no. 4, pp. 893–897, 2012.

[20] J. J. Rassweiler *et al.*, "iPad-assisted percutaneous access to the kidney using marker-based navigation: initial clinical experience," *European urology*, vol. 61, no. 3, pp. 628–631, 2012.

[21] B. G. Lindeque, O. I. Franko, and S. Bhola, "iPad apps for orthopedic surgeons," *Orthopedics*, vol. 34, no. 12, pp. 978–981, 2011.

[22] R. J. Rohrich, D. Sullivan, E. Tynan, and K. Abramson, "Introducing the new PRS iPad app: The new world of plastic surgery education," *Plastic and reconstructive surgery*, vol. 128, no. 3, pp. 799–802, 2011.

[23] F. Volont, J. H. Robert, O. Ratib, and F. Triponez, "A lung segmentectomy performed with 3D reconstruction images available on the operating table with an iPad," *Interactive cardiovascular and thoracic surgery*, vol. 12, no. 6, pp. 1066–1068, 2011.

[24] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "LabelMe: a database and web-based tool for image annotation," *Int J Comput Vis*, vol. 77, no. 1-3, pp. 157–173, May 2008.