2014 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS 2014)
September 14-18, 2014, Chicago, IL, USA

A Fast, Low-Cost, Computer Vision Approach for Tracking Surgical Tools

Rodney Dockter[1], Robert Sweet M.D.[2], and Timothy Kowalewski Ph.D.[3]

Abstract— **Given the rise in surgeries performed with surgical robots and associated robotics research efforts, tool tracking methods have the potential to provide quantitative feedback concerning surgical performance and establish absolute tool tracking to help advance surgical robotics research. We have created a platform-agnostic method for low-cost tracking of surgical tool shafts in Cartesian space in near real time. We employ a joint Hough Transform - Geometric Constraint approach to locate the tool tips in the stereo camera channels independently. Cartesian coordinates are registered using a custom polynomial depth - disparity model.**

The algorithm was developed using a low-cost experimental webcam setup and evaluated using a da Vinci surgical endoscope. The algorithm was benchmarked for 3D tracking accuracy and computational speed. The system can locate the tool tip in 3D space with an average accuracy of 3.05 mm at 25.86 frames per second using the webcam setup. For the endoscope setup this algorithm has an average tracking accuracy of 8.68 mm in 3D and 1.88 mm in 2D with an average frame rate of 26.9 FPS. The algorithm also demonstrated successful tracking of tools using captured video from a real surgical procedure.

I. Introduction

The number of surgeries performed worldwide with the da Vinci surgical robot (Intuitive Inc, Sunnyvale, CA) in 2005 was under 50,000. This number rose to over 350,000 by 2011 [1]. Robot-Assisted Minimally Invasive Surgery (RMIS) has facilitated improvement in success rates and patient experiences in surgical procedures. This new era has spurred a departure from traditional Halstedian models, wherein an apprentice learns by watching a master and demands new training and evaluation methods in order to teach and evaluate surgeons before granting access to patients.

Recent developments in training curricula require metrics related to spatial and temporal data. Studies have shown that such data can provide valuable information for discriminating expert from novice surgeons [2]. Certain key motion metrics can be used to differentiate surgical skill level, including; path length, economy of motion, time, motion smoothness and response orientation [3]. These motion metrics require a fast and accurate tool tracking method. Furthermore, within the field of medical training and simulation, cost is commonly a barrier to entry for new systems. With this in mind, the proposed tracking system was designed to be low-cost and platform independent in order to increase the chance of adoption.

A low-cost and accurate tool tracking algorithm is also beneficial to research groups utilizing the RAVEN surgical robotics platform. Currently this system utilizes kinematic data in order to infer tool tip position [4]. For the development of sophisticated controls algorithms or other human interfaces where tool position is utilized, the kinematic calculation of the tool tip may not be accurate enough. Such instances reveal the need for absolute position tracking of robotic tools.

A review of the literature indicated four main methods for tracking laparoscopic and robotic tools. These methods include mechanical, electromagnetic, ultrasound, and computer-vision techniques [5]. The mechanical, electromagnetic and ultrasound systems are all ill-suited for robotic applications due to prohibitive hardware issues (tool augmentation and workspace constraints) as well as cost. The joint kinematics from the da Vinci robot cannot be widely used to track the tool tip position since this information is not openly available to users. Even if this information were available, kinematic calculations of the end effector position suffer from compliance in the cabled joints and links as well as compounded finite uncertainties in sensing.

Computer vision approaches for tracking surgical tools have been developed both by commercial entities as well as research institutions. Stryker Inc. (Kalamazoo, MI) and Northern Digital Inc. (Ontario, Canada) both produce commercial optical tracking systems. These systems utilize LED beacons affixed to the surgical tool for tracking. However, such systems are expensive and require hardware modifications. Research groups have attempted tracking methods such as the use of color marker segmentation via pixel thresholding [6], [7], [8] and color thresholding across the whole image in order to extract the tool shaft [9], [10], [11]. While computationally efficient (15-17 Hz) these methods suffer from color and lighting variation issues. Tool tracking has been performed using geometric constraints in order to identify the tool shaft [12], [13], [14]. Using a Hough transform to identify the lines bordering the tool shaft, the tool tip can then be found along the tool shaft using color thresholding or physical measurements. The highest reported frame rate using this method was 16 Hz with a 2D pixel accuracy of 27.8 pixels [14]. Tool detection has also relied on feature descriptor libraries [15], [16]. Using a physical model of the tool tip, a library of descriptors corresponding to the object can be compiled and used to detect the object.

[1] Rodney Dockter is with the Department of Mechanical Engineering, University of Minnesota, Minneapolis, MN, 55414, USA dockt036@umn.edu

[2] Robert Sweet is an associate professor and urologic surgeon at the University of Minnesota Medical School, Minneapolis, MN, 55414, USA rsweet@umn.edu

[3] Timothy Kowalewski is the Richard and Barbara Nelson assistant professor of Mechanical Engineering, University of Minnesota, Minneapolis, MN, 55414, USA timk@umn.edu

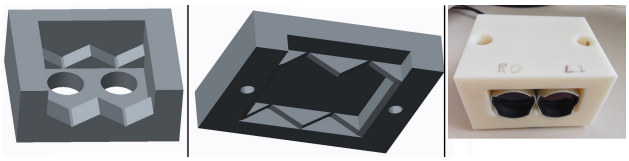

Fig. 1. The experimental webcam mount.

Fig. 2. The da Vinci camera unit and Endoscope.

Such a method is computationally prohibitive for real time tracking (1 FPS) [15].

Recently Richa et al. have had success using a model image of a surgical tool in conjunction with a Sum of Conditional Variance (SCV) metric for finding tools in retinal surgery [17]. The relatively simple shape of retinal surgery needles and smooth background of the field of view allows the use of 'model images', however this would not be a feasible approach given the complex end-effectors found on surgical robots. This group also did not formulate absolute 3D tracking methods and instead utilized only relative pixel disparities.

The literature has indicated a lack of a computer vision solution which can accurately track tools in 3D space, in near real time, which is immune to color changes, and with indifference to tool type. The goal of this work was to design a computer vision solution which can simultaneously provide these requirements at a reasonable cost. The primary contribution of this work is a tracking algorithm which is computationally inexpensive and accurate relative to prior art. An additional benefit is the use of low cost hardware to increase the possibility of adoption in cost-constrained medical simulation applications.

II. Design

A. Hardware Design

Two distinct camera systems were utilized for the development of this algorithm. The first camera unit is a stereo camera mount comprised of two USB Lifecam Studio cameras from Microsoft (Redmond, WA). These provide frame rates of 30 FPS at standard resolutions with slight drops at higher resolutions (up to 1080p). At the time of purchase, each camera cost approximately $50 USD. While webcams suffer from rolling shutter effects, their low-cost availability permits widespread use of this work by other researchers. Webcams allow for convenient algorithm development as well as use in laparoscopic training modules where endoscopes are prohibitively expensive.

A rapid prototype stereo camera mount was designed using computer aided design (CAD) software (Fig. 1) (PTC Creo, Lansing, MI). The camera axes are mounted parallel without allowing rotation or translation in the camera housing. The mount can be 3D printed with standard machines. The top and bottom of the mount are held together by nylon set screws. The interocular distance for the webcam stereo mount is 29.1 mm. The CAD files are available via www.bit.ly/1npTAGr.

The second camera unit was a surgical endoscope from a da Vinci S surgical system (Fig. 2). This endoscope provides high definition stereo images to the surgeon for 3D visualization on the master console. The technical specifications for the endoscope are not made available but were empirically observed. The endoscope is capable of resolutions up to 1920x1080 and provides a framerate up to 100 FPS. This setup is relatively similar to the experimental camera setup except for a change in interocular separation. For the endoscope cameras the interocular separation was measured to be 5.1 mm.

For testing and calibration of the endoscope, a Digital Video Interface (DVI) frame grabber was utilized. The frame grabber chosen was the VC200xUSB dual channel DVI box from Electronic Modular Solutions (Wigston, England). This DVI frame grabber allows simultaneous capture from two separate DVI sources.

For benchmarking purposes, a computer running Microsoft Windows 7 was utilized. This PC used an Intel (Santa Clara, CA) Core i7 processor running at 3.0 GHz with 32 GB of RAM.

B. Software and Algorithm Design

The tracking algorithm was developed with the C++ ISO 11 programming language. Additional libraries included the OpenCV library from Willow Garage (Menlo Park, CA) and the Qt library from Digia (Helsinki, Finland).

The flow of information for 3-D tool tracking is functionally decomposed into 3 major steps: 2-D object detection, depth extraction and Cartesian coordinate calculation. While more sophisticated geometric computer vision models do exist in the literature, the data-driven technique for disparity to depth mapping allowed the authors to better understand the mathematical model for the cameras as well as employ more complex models for Cartesian registration. The required input is a captured stereo video frame and the output is Cartesian tool coordinates and time (Fig. 3). While the serial flow of information has the downside of potential errors compounding, the benefit is minimal computational effort as a means to improve frame-rate. Computational effort is decreased since the object recognition can occur simultaneously in each channel and the Cartesian coordinate calculation is reduced to a single linear equation as opposed to a full stereo correspondence method.

A novel approach was identified in which the known geometric constraints of the tool shaft could be exploited using a Hough transform in order to identify lines in the image. The algorithm follows eight high level steps outlined in Algorithm 1. Once the edge gradients are computed using the Sobel operator, the edges are selected using a dynamic threshold. This threshold is set by computing the number

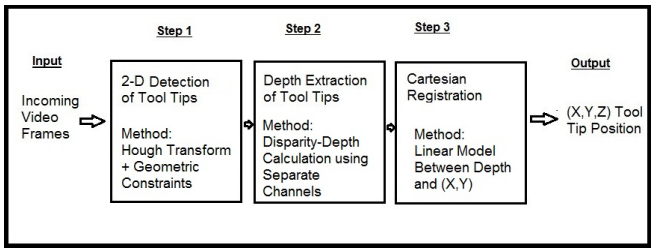

Fig. 3. The 3-step approach for object tracking.

Data: Frame
Result: Cartesian Tool Location
Convert frame to grayscale;
Blur grayscale image;
Sobel edge detection;
Dynamic edge gradient threshold;
Probabilistic hough transform;
for *Each line in PHT line array* **do**
  Extract unique line pair;
  Compute line parameters;
  if *Parameters best match constraints* **then**
    Save line;
  else
    Extract 2 new lines;
  end
end
Calculate endpoints closest to prior location;
Save tool tip location;

Algorithm 1: Detection of surgical tool tips

of edge pixels in the image. If the number of edge pixels is too high (or low), dependent on resolution, the gradient threshold value is then increased (or decreased).

In particular we utilized the probabilistic Hough transform (PHT) designed by Matas et al. [18]. Using the implementation of this algorithm in OpenCV we are able to achieve improved frame rates with the added benefit that the endpoints of the detected lines are determined. Hough transforms have been shown to have severe limitations in in-vivo settings [12]. These limitations include lost lines due to minor occlusions. However, the use of the PHT variation limits the loss of lines due to occlusions along the shaft.

The line endpoint information is sorted according to certain known geometric constraints in order to isolate the lines corresponding to the borders of the tool shaft. The particular geometric constraints utilized are unique to this work and are the result of manual frame inspection (Table. I).

Once the correct lines have been isolated, the endpoints of the lines can be utilized to determine which end of the tool shaft contains the tool wrist. For initialization purposes the tool end closest to the center of the image is taken as the tool wrist. After initialization, the endpoints closest to a previous known location are taken to be the wrist. Using the two 'wrist' endpoints, the midpoint of these endpoints is taken as the pixel location of the wrist. In the case of multiple surgical

Constraint	Condition
Lines along the tool shaft are parallel	$\Delta\theta < \theta_{threshold}$
Endpoints of the lines should be near to each other.	$d(p(i)_{1,1}, p(i)_{2,1}) < d_{threshold}$
Length of two lines should be longer than any other set of parallel lines	$d(p(i)_{1,1}, p(i)_{1,2}) > length_{max}$
The resultant endpoints should be 'near' the last known location.	$d(p(i)_{1,1}, p(i-1)_{1,1}) < d_{near}$

TABLE I. Geometric constraints and their implementation in code. d() represents the euclidean distance formula and $p(k)_{m,n}$ represents a point (x,y) within frame k, line m, index n

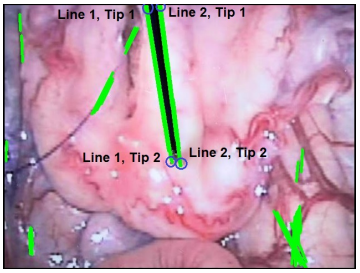

Fig. 4. The four end points of the tool shaft lines are circled.

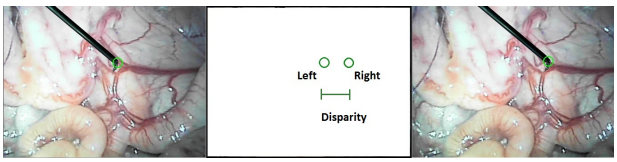

Fig. 5. The tool is localized in separate channels and then the disparity is computed.

tools in the field of the view, the line endpoint information is sorted according to the previously stated known constraints but then separated according to prior spatial information as well as the angle of the tool shaft.

The depth extraction relies on the pixel location of the tool tip detected independently in both stereo channels. Using the right and left pixel location of the tool tips, the disparity can be calculated (Fig. 5). This method of single tool tip disparity calculation requires a simpler calculation than a full stereo correspondence. The tool tip disparity is computed using the Euclidan distance formula. Disparity is then used to determine the depth to the tool tip and Cartesian coordinates.

The formula used to calculate depth from disparity depends greatly on the camera setup and model selected. Several models exist for computing depth. The most ubiquitous of these models is the standard linear transformation outlined in the works of Zhang [19] and Tsai [20]. This method did not result in a sufficiently accurate model for the endoscope optics [21]. The depth model utilized is determined using a planar calibration board and data correlating depth with disparity. The calibration data was used to heuristically

devise a model with the best fit. This empirically determined model allows the use of non-standard camera optics with minimal drop in accuracy.

The final step in the tracking algorithm is the correlation between pixel space coordinates of the tool tip and real world coordinates. The Z_w Cartesian coordinate, determined from the depth extraction method, is used to compute the remaining coordinates, (X_w, Y_w). This Cartesian Z_w model (depth) is a linear combination of disparity, x and y pixel location for the webcam setup (Eq. 1). The Cartesian Z model for the endoscope is a third order polynomial in order to offset disparities (Eq. 2). The offset disparity ($disp_o$) is then used in an exponential function to compute depth (Eq. 3).

$$Z_w = a_1(disp^{a_2}) + a_3 x_p + a_4 y_p + a_5 \tag{1}$$

$$\begin{aligned} disp_o = &\begin{bmatrix} b_0 & \dots & b_n \end{bmatrix} * \\ &\begin{bmatrix} 1 & x & y & x^2 & xy & y^2 & x^3 & x^2y & xy^2 & y^3 \end{bmatrix}^T \end{aligned} \tag{2}$$

$$Z_w = c_0 e^{c_1(disp - disp_o)} \tag{3}$$

The equation coefficients (a_i, b_i, c_i) for this model are empirically determined using a linear regression fit. Separate calibrations are required for the experimental camera setup and the da Vinci Endoscope. This method of Cartesian registration resulted in a simple model with adequate performance.

III. Experimental Design

A series of benchmarking experiments were developed in order to analyze the computational speed of the algorithm, the 2D and 3D tracking accuracy, and the tracking noise for a stationary tool. 'Robustness' was also evaluated based on what conditions result in successfully located tools.

The endoscopic camera was evaluated using a recorded video of a prostate removal procedure. Each metric is evaluated for the experimental webcam setup as well as the endoscope camera.

In order to determine the latency (tracking frame rate), we calculate the time each tracking component takes to complete. The time for each component to complete is computed through the use of timestamps taken at the beginning and end of each component during its thread execution. This occurs at the moment it receives data and returns data, respectively.

The time the second component (2D object detection) takes to complete ($T_{detect,right}, T_{detect,left}$) is computed as the difference in time between when a new right or left frame is received from the capture thread and when the 2-D pixel location of all tool tips has been found. Since each channel's 2D object detection is performed simultaneously, we adopt the worst case scenario and consider the time difference to be equal to the larger of the two channels time difference. The time 3D localization requires ($T_{3DLocate}$) is computed as the difference in time from when the pixel location from the object detection threads is acquired and when the Cartesian coordinates have been calculated.

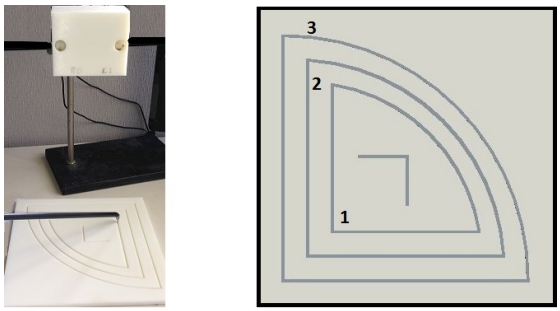

Fig. 6. Known trajectory board below stereo camera and model with numbered arcs.

$$T_{tracking}(ms) = Max(T_{detect,right}, T_{detect,left}) + T_{3DLocate} \tag{4}$$

The second metric to be characterized is the accuracy with which the tracking algorithm can locate the tool tip in 3D Cartesian coordinates. Tracking accuracy is determined by moving the tool tip around a known fixed trajectory. Using the pixel coordinates and disparity found as the tool tip is moved around this trajectory, the world (X_w, Y_w, Z_w) coordinates are calculated. The reprojected world coordinates are then compared with the coordinates of the known trajectory in order to determine the error for each location. The reference trajectory was modeled in Matlab as analytically synthesized, finitely-spaced points.

In order to have a repeatable trajectory with known coordinates, a board was developed using CAD software. This trajectory board contains 3 circumscribed paths located on the same horizontal plane (Fig. 6). Each path is recessed into the board so that the tool tip can rest in the groove and repeatably follow the track. Since the board was designed in CAD and then 3D printed, the geometry of the board is accurately known to within the tolerance of the 3D printer. For this board, a Stratasys Dimension 1200es 3D printer was used (Stratasys, Minneapolis, MN). This printer is capable of resolutions up to 0.254 mm.

The third metric characterized is the overall noise of the tool reprojection. Noise is evaluated by calculating the deviation of a stationary tool tip over time. In order to calculate this metric accurately, the surgical tool is fixed in a clamp with the tool tip in the visible field. The tool tracking algorithm is then run and logs tool position. After the tool tracking is finished, the data is exported for analysis. Using the position data, an average location $(X_{avg}, Y_{avg}, Z_{avg})$ is calculated.

Using the calculated average position, the deviation of each reprojected data point from the average is computed. After all errors have been calculated, an average deviation is determined. This average deviation is considered the noise of the tool tracking algorithm. For any given stationary noise of the tool tracking algorithm, it is not theoretically possible to attain a more accurate overall tracking accuracy. The noise

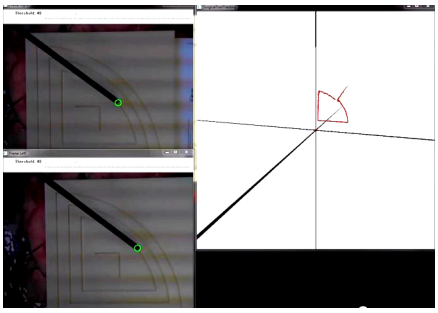

Fig. 7. Tool tracking in 3D using the known trajectory board. 3D position is displayed on the right.

from this algorithm is primarily due to the two stereo object detection channels.

IV. Results

For the webcam camera setup, the algorithm takes an average of 39.9 ms for each frame to be analyzed and the tool position calculated. This results in a working framerate of 25.86 FPS. The camera capture rate is 30 FPS.

In terms of accuracy in tool tip localization, the stationary noise level (1.293 mm) is rather large. Despite this noise level the average tracking error is very low (3.05 mm) considering this is less than the diameter of the tool shaft. The tracking errors are reported for target depths between 200 and 400 mm. The 95^{th} percentile of tool reprojections are within 5.474 mm of the known location. The percentage of time where the surgical tool is located within 8 mm of the known trajectory is 99.4%.

Performance Metric	Webcam	Endoscope
Computation Time (ms)	39.9	33.99
Frame Rate (FPS)	25.86	26.98
Depth Reprojection Error (mm)	4.09	7.89
Localization Noise (Total) (mm)	1.29	7.92
Average 3D Error (mm)	3.05	8.68
Average 2D Error (mm)	1.59	1.88
95^{th} Percentile Error (mm)	5.47	15.06
Percent Within 8 mm	99.4 %	55.22 %

TABLE II. Performance metrics for the webcam setup and the endoscope setup.

For the endoscope camera setup, the average frame rate was found to be 26.98 FPS. It terms of tracking noise, the endoscope setup suffers from substantial noise in the depth extraction model. For a stationary tool, the tracking algorithm returns an average noise of 7.92 mm in tool reprojection.

It terms of tracking accuracy, the overall reprojection error when compared with the known trajectory was found to be 8.68 mm, while this is still less than the diameter of the tool shaft, the performance is inferior to that of the webcam. Similarly, the amount of reprojections within 8 mm of the known trajectory was found to be 55.2 %. This tracking accuracy is significantly lower than the experimental webcam

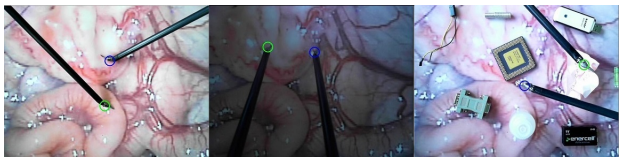

Fig. 8. Tool tracking configurations using the webcam setup (Left Channel).

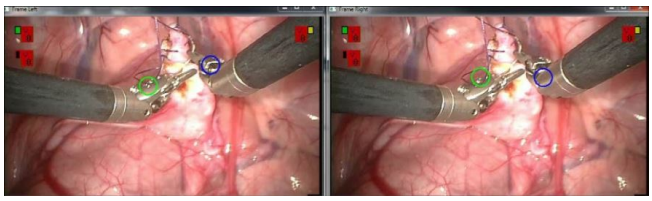

Fig. 9. Tool tracking using a real surgical video (endoscope).

setup. However, it is important to note that the average 2D reprojection errors for the endoscope setup in the X_w and Y_w components were only 0.35 mm and 1.25 mm, respectively.

As indicated in Figure 8 the presence of background objects and poor lighting does not negatively affect the tool detection algorithm using the webcam setup. For the endoscope setup using surgical video (prostate removal), both tool tips were correctly located 51.2 % of the time (54 second video clip) (Fig. 9). Similarly, at least one tool tip was correctly located 88.0 % of the time. While neither tool tip was correctly located 12.0 % of the time, this error is primarily due to tools leaving the field of view and occlusions from electrocautery smoke. The radial distortion within the endoscope also led to localization errors not found in the webcam setup.

V. Conclusion and Future Work

The surgical tool tracking algorithm presented in this work provides promising results in terms of accuracy and primarily in term of speed. For the webcam setup, the tool tracking algorithm met all initial design criteria. The 99.4% successful tool localizations exceeds the previously reported accuracies. Reiter et al. using a manual frame-by-frame analysis, reported a 93 % accuracy over 1600 test frames [15]. That accuracy was reported in 2D pixel space and also functions at a much slower frame rate (1.2 secs/frame). While the experimental setup in the referenced work differs with our work, the reported accuracy provides a baseline for comparison. The working frame rate of 25.86 FPS exceeds the 16 Hz found in prior art [6], [8], [14].

The tool tracking algorithm is agnostic to end-effector type since only the shaft is detected. Therefore any tool tip can be tracked as long as the shaft is consistent. The algorithm also performs well in varied lighting conditions since it does not depend on color. For the endoscopic camera setup, the working frame rate of 26.98 FPS exceeds prior art. The average 2D reprojection errors in the X_w and Y_w components were very good (0.35 mm and 1.25 mm), indicating that the depth extraction model is responsible for most of the error. This is a result of the constrained disparity variation inherent

in a stereo camera with such a narrow interocular separation as well as the uncertainties in camera optics. The range of disparities found in this camera setup was found to be about 10 pixels for a depth range of 200 mm. This results in a high signal to noise ratio.

The primary contribution of this work is the near real time performance. Prior art has not been able to achieve frame rates approaching 30 hz. Our performance enables several applications of tool tracking where real time location updates are necessary. The 2D accuracy is another contribution to the field. The endoscope and webcam camera setups achieved 2D accuracies around 1 mm which has not been previously reported. Prior art has focused on 2D object detection and only relative position information. As such the 3D tracking characterization and absolute Cartesian registration are both unique contributions to the art.

Future work will incorporate the use geometric computer vision methods for 3D coordinate extraction. The use of feature detectors will also be employed in conjunction with epipolar geometry to constrain the search space in the second stereo channel. Other work will include the immediate extension of this tracking algorithm to laparoscopic tools and incorporation into real surgical applications for experimental analysis. Finally we intend to make our work (software and CAD models) available for research purposes to both clinical and engineering researchers (www.bit.ly/1npSXwv).

VI. Limitations

The tool tracking algorithm is not capable of tracking or detecting the configuration of the tool wrist. However, this problem has already been partially resolved by White et al. who developed a mechanism for extracting wrist configuration information from surgical robotic tools [22]. A synthesis of the current tool tracking scheme with an end-effector configuration tool may be explored. The authors do acknowledge that this algorithm will lose the tool position should the end-effector or end of the tool shaft become occluded. These occlusions can include smoke, blood, other tools, and movement outside the field of view.

ACKNOWLEDGMENT

The authors would like to thank Troy Reihsen of the University of Minnesota CREST and SimPORTAL groups and Dr. Art Erdman and Darrin Beekman of the Medical Devices Center for contributing their time and resources.

References

[1] R. Satava, R. Smith, and V. Patel, "Fundamentals of robotic surgery: Outcomes measures and curriculum development," in *Society of Laparoendoscopic Surgeons*, 2012.

[2] T. N. Judkins, D. Oleynikov, and N. Stergiou, "Objective evaluation of expert and novice performance during robotic surgical training tasks," *Surgical Endoscopy*, vol. 23, pp. 590–597, 2009.

[3] N. Stylopoulos, S. Cotin, S. Dawson, M. Ottensmeyer, P. Neumann, R. Bardsley, M. Russell, P. Jackson, and D. Rattner, "Celts: A clinically-based computer enhanced laproscopic training system," in *Medicine Meets Virtual Reality 11: NextMed : Health Horizon*, J. D. Westwood, Ed., 2003.

[4] M. J. H. Lum, D. C. W. Friedman, G. Sankaranarayanan, H. King, K. F. II, R. Leuschke, B. Hannaford, J. RosenRosen, and M. N. Sinanan, "The raven: Design and validation of a telesurgery system," *Internation Journal of Robotics Research*, vol. 00, pp. 1–16, 2009.

[5] M. Chmarra, C. Grimbergen, and J. Dankelman, "Systems for tracking minimally invasive surgical instruments," *Minimally Invasive Therapy*, vol. 16:6, pp. 328–340, 2007.

[6] G.-Q. Wei, K. Arbter, and G. Hirzinger, "Automatic tracking of laparoscopic instruments by color coding," in *Medical Robotics and Computer-Assisted Surgery*, 1997, pp. 357–366.

[7] A. Krupa, J. Gangloff, C. Doignon, M. de Mathelin, G. Morel, J. Leroy, L. Soler, and J. Marescauz, "Autonomous 3-d positioning of surgical instruments in robotized laparoscopic surgery using visual servoing," *IEEE Transactions on Robotics and Automation*, vol. 19, pp. 842–853, 2003.

[8] M. Groeger, K. Arbter, and G. Hirzinger, *Motion Tracking for Minimally Invasive Robotic Surgery*. I-Tech Education and Publishing, 2008.

[9] C. Lee and D. Uecker, "Image analysis for automated tracking in robot-assisted endoscopic surgery," *Pattern Recognition*, vol. 8, pp. 88–92, 1994.

[10] C. Doignon, FlorentNagcotte, and M. D. Mathelin, "Detection of grey regions in color images: Application to the segmentation of a surgical instrumenty in robotized laparoscopy," in *Proceedings of the IEEE/RSJ Conference on INtelligent Robotics and Systems*, 2004.

[11] C. Doignon, P. Graebling, and M. D. Mathelin, "Real-time segmentation of surgical instruments inside the abdomonal cavity using a joint hue saturation color feature," *Real-Time Imaging*, vol. 11, pp. 429–442, 2005.

[12] S. Voros, J.-A. Long, and P. CinQuin, "Automatic detection of laparoscopic images: A first step toward high-level command of robotic endoscopic holders," *The International Journal of Robotics Research*, vol. 26, pp. 1173–1190, 2007.

[13] C. Doignon, F. Nageotte, and M. D. Mathelin, "Segmentation and guidance of multiple rigid objects for intra-operative endoscopic vision," *Dynamical Vision*, pp. 314–327, 2007.

[14] R. Wolf, J. Duchateau, P. Cinquin, and S. Voros, "3d tracking of laparoscopic instruments using statistical and geometric modeling," in *Medical Image Computing and Computer Assisted Intervention*, 2011.

[15] A. Reiter, P. K. Allen, and T. Zhao, "Feature classification for tracking articulated surgical tools," in *Medical Image Computing and Computer Assisted Intervention*, 2012.

[16] Z. Pezzementi, S. Voros, and G. D. Hager, "Articulated object tracking by rendering consistent appearance parts," in *IEEE International Conference on Robotics and Automation*, 2009.

[17] R. Richa, M. Balicki, R. Sznitman, E. Meisner, R. Taylor, and G. Hager, "Vision-based proximity detection in retinal surgery," *IEEE Transactions on Biomedical Engineering*, vol. 59, pp. 2291–2301, 2012.

[18] J. Matas, C. Galambos, and J. Kittler, "Robust detection of lines using the progressive probilistic hough transform," *Computer Vision and Image Understanding*, vol. 78, pp. 119–137, 2000.

[19] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 1330–1334, 2000.

[20] R. Tsai, "A versatile camera calibration techniaue for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses," *IEEE Journal of Robotics and Automation*, vol. 3, pp. 323–344, 1987.

[21] R. Dockter and T. Kowalewski, "A low-cost computer vision based approach for tracking surgical robotic tools," *Journal of Medical Devices*, vol. 7, pp. 1–2, 2013.

[22] L. W. White, T. M. Kowalewski, T. S. Lendvay, and B. Hannaford, "Motion and video capture for tracking and evaluating robotic surgery and associated systems and methods," United States of America Patent 0 253 360 A1, 2012.