Vision based relative position estimation in surgical robotics.

MUTHUKRISHNAN, R., KANNAN, S., PRABHU, R., ZHAO, Y., BHOWMICK, P. and HASAN, M.J.

2023

Vision based Relative Position Estimation in Surgical Robotics

[1]Ramkumar Muthukrishnan
School of Engineering
Robert Gordon University
Aberdeen, UK
r.muthukrishnan@rgu.ac.uk

[2] Somasundar Kannan
School of Engineering
Robert Gordon University
Aberdeen, UK
s.kannan1@rgu.ac.uk

[3] Radhakrishna Prabhu
School of Engineering
Robert Gordon University
Aberdeen, UK
r.prabhu@rgu.ac.uk

[4] Yafan Zhao
School of Engineering
Robert Gordon University
Aberdeen, UK
y.zhao@rgu.ac.uk

[5] Parijat Bhowmick
Electronics and Electrical Engineering Department
Indian Institute of Technology
Guwahati, India
parijat.bhowmick@iitg.ac.in

[6] Md Junayed Hasan
National Subsea Centre
Robert Gordon University
Aberdeen, UK
j.hasan@rgu.ac.uk

***Abstract*—Teleoperation-based Robotic-Assisted Minimally Invasive Surgery (RAMIS) has gained immense popularity in medical field. However, limited physical interaction between the surgeon and patient poses a significant challenge. In RAMIS, the surgeon operates the robotic system remotely, which can diminish the personal connection and raise concerns about immediate responsiveness to unforeseen situations. Additionally, patients may perceive RAMIS as riskier due to potential technological failures and a lack of direct surgeon control. Surgeons have identified accidental clashes between surgical instruments and tissues as a critical issue. This work presents a technique that measures the distance between a surgical tool and tissue by extracting feature points from a Static Virtual Marker (SVM) and employing a classic feature detection algorithm Fast Orientedand Rotated Brief (ORB). Using a customized surgical robot and a ROS-based transform measurement system, this approach was successfully validated in the Gazebo simulation environment, offering safer surgical operations.**

***Index Terms*—Image Processing, Robot Operating System, Gazebo simulation, Surgical robot**

I. Introduction

RAMIS, a growing surgical technique, offers several advantages over standard laparoscopy, such as reduced cognitive load, increased dexterity, improved precision, ergonomics, tremor control, movement scaling, and enhanced visual perception [1]. However, these benefits come with limitations on mobility and vision range, making the surgical process challenging. Restricted visibility adds cognitive strain and increases the risk of collisions and tissue injuries [2]. To address this issue, a successful approach is necessary [3],which is presented in this research work. This paper organised as follows: after the introduction section I, related work for this problem is discussed in section Development of algorithm described in three sub section with experiment under section III. Section IV describes the validation method with result and the conclusions are drawnin the section V.

II. RELATED WORKS

Instances with a comparable range of relevance are collected for this research. To begin with, minimise clashes between moving tools and tissue in nasal surgery, Prohibited Region Active Constraints (ACs) utilising Vector Field Inequalities (VFI) and expanded on it by presenting the ACs framework [2]. Later that, a spontaneously annotated training data-set and an enhanced position-prediction deep-learning framework with visual-based marker-less position estimation method has been presented. It quantifies the position of surgical tool shafts using a monocular endoscope [4]. Additionally, an Prohibited Region of Virtual Fixtures (VF). A marker-less instrument tracking approach is used to present an Extended Kalman Filter (EKF) for position calculation, which assures a more reliable application of VF on the instrument by integrating kinematics and vision data [5]. Thereafter, a modified Anchoring Network was created using FR-CNN to identify the tool during the key-hole pro- cedure [6]. For certain surgical tool data sets, a multiple tool tracing architecture based on geometric object descriptors was presented [7]. Moreover, an automated tracking sys- tem for surgical instruments has been presented using an improved TernausNet-16 network architecture [8]. Recently, for automated instrument recognition during operations, a highly optimised InceptionResnetV2 model was created [9]. Instrument collisions are prevented by imposing a repellent force on the surgeon. Although harm to tissues may result from unintentional current flow in the patient's anatomy.

III. PROPOSED METHOD

In this research, a new approach is developed to prevent ion between surgical tool and organ by estimating the distance between them for RAMIS is presented. The contributions are

- Created a new Static Virtual Marker (SVM) inspired by the April tag.
- Developed a algorithm frame structure with integration of classic feature detection method for estimating the distance between the tissue and the surgical tool.

A algorithm flowchart for easier comprehension is illustrated in Fig. 1. In initiation part, the node (algorithm) begins with extracting each frame from live video, when it is activated. This live video is obtained from the camera in the simulation environment. It then begins with the initial stage of processing.

A. *First phase of processing*

Each frame is transformed from RGB to HSV in the initial stage of processing. The RGB colour model was developed by combining the shades of red, green and blue. Hue, saturation, and value or brightness, are combined to form HSV. Hue, which may be characterised as an angle with a range of $0 - 360^o$, represents the character and quality of a colour. Colour intensity is referred to as saturation. The contrast of the colour, commonly referred to as brightness or represents value. According to this, switching from RGB to HSV frames allows for improved processing to recognise objects in various lighting situations, shadows, etc. In order to create a binary frame, the thresholding approach was used to choose the colours of the objects (both tissue and instrument) from the HSV frame. It depicts the background scene as black, with the targeted items as white. By capturing the tissue and instrument areas alone and maintaining them in a first and second loop for subsequent processing, binary frames are supported to calculate the first and second contours of the targeted item. If the tissue area is bigger than 50 pixels under the first loop, the tissue location is calibrated for further processing. From the centre of mass, the centroid coordinates for the tissue, X_c and Y_c, have been determined in;

$$X_c^1 = \frac{\sum_{i=1}^{K_1} X_i^1}{K_1}, \tag{1}$$

$$Y_c^1 = \frac{\sum_{i=1}^{K_1} Y_i^1}{K_1}, \tag{2}$$

where K_i is the total number of pixels that match the tissue, and X_i and Y_i are the x and y coordinates of the i^{th} pixel on the screen plane. In this research, an eight-bit RGB camera with an 640 x 480 resolution was employed in the simulation environment. As a result, wherever the targeted item (tissue) presents inside the screen plane, its location is shown in pixels. The pixel-to-centimetre conversion is obtained using:

$$S_{sw_1} = \frac{1 - T_{sw_{(pixel)}} * 2.54}{100} = 0.0245cm, \tag{3}$$

$$S_{sh_1} = \frac{1 - T_{sh_{(pixel)}} * 2.54}{100} = 0.0245cm. \tag{4}$$

The characters S_{sw_i} and S_{sh_i} represent the width and height of the screen, respectively. T_{sw} and T_{sh} are the positions of the targeted item (tissue) in the width and height directions of the screen. The x and y axes coordinates are calibrated from the Screen Plane (SP) to the Global Plane (GP), which are determined below:

$$S_{sw_{(SP-GP_1)}} = -(0.0036*(S^2_{sw_1}))+(0.7938*(S_{sw_1}))-0.0403, \tag{5}$$

$$S_{sh_{(SP-GP_1)}} = -(0.0068*(S^2_{sh_1}))+(0.0395*(S_{sh_1}))-0.4439, \tag{6}$$

The location of the targeted item in terms of screen width and height is calibrated to the GP's x and y axes coordinates, which are denoted by $S_{sw_{(SP-GP_i)}}$ and $S_{sh_{(SP-GP_i)}}$, respectively. Following that, in the last phase of processing, the calibrated position of the tissue is used. Before proceeding to standard feature detection processing, under the second loop. It checks the instrument region, if the area of the instrument is greater than 100 pixels, a Static Virtual Marker (SVM) is overlayed on the instrument by obtaining the centroid coordinates of the targeted instrument, it has been determined in;

$$X_c^2 = \frac{\sum_{i=1}^{K_2} X_i^2}{K_2}, \tag{7}$$

$$Y_c^2 = \frac{\sum_{i=1}^{K_2} Y_i^2}{K_2}. \tag{8}$$

B. *Second phase of processing*

Simple definitions of image feature points include dark spots in bright areas and bright spots in dark areas, or contour points. These features are highly significant in the picture. Finding points for features is accomplished using the Fast Oriented and Rotated Brief (ORB) [1] method with a fast approach. Fast's fundamental concept is to identify standout points. To perform this, first it compares a point with its surroundings, and if it differs significantly from the majority of them, choose it as a feature point.

The ORB algorithm combines the Binary Robust Independent Elementary Features (BRIEF) descriptor and the FAST key-point detector. The scale pyramid picture is transformed to prepare it for feature extraction by employing the ORB method. Generally, it takes a 16-pixel round with the digits 1 to 16 marked circularly which is created by the FAST method to locate the key-points that start with random points on the initial inspection. After locating the key points, Harris Corner was utilised to determine the optimum N points. To locate the crucial location, an intensity centroid is employed. The picture patch's grey value, referred to as the centroid, serves as the weight centre [1]. The following defines a moment of image patching;

$$m_{pq} = \sum_{xy} x^p y^q I(x, y). \tag{9}$$

Following the discovery of the key-point [1], the centroid intensity method of orientation searches is used and described as follows;

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}}\right), \tag{10}$$

$$\theta = \arctan\left(m_{01}, m_{10}\right), \tag{11}$$

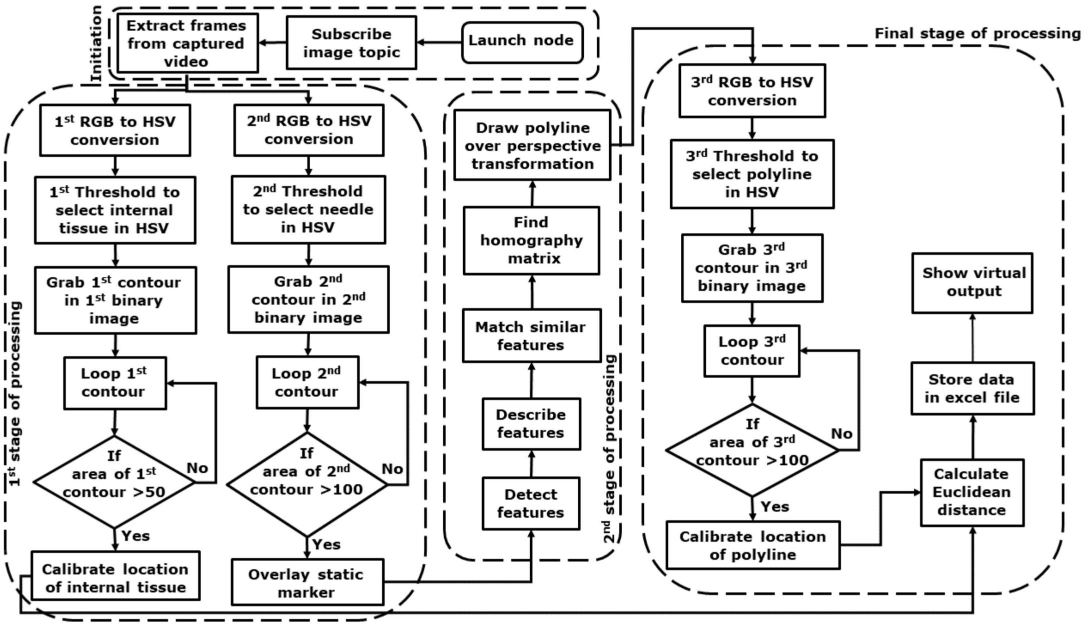
Initiation
Extract frames from captured video
Subscribe image topic
Launch node
1st stage of processing
1st RGB to HSV conversion
1st Threshold to select internal tissue in HSV
Grab 1st contour in 1st binary image
Loop 1st contour
If area of 1st contour >50
No
Yes
Calibrate location of internal tissue
2nd RGB to HSV conversion
2nd Threshold to select needle in HSV
Grab 2nd contour in 2nd binary image
Loop 2nd contour
If area of 2nd contour >100
No
Yes
Overlay static marker
2nd stage of processing
Detect features
Describe features
Match similar features
Find homography matrix
Draw polyline over perspective transformation
Final stage of processing
3rd RGB to HSV conversion
3rd Threshold to select polyline in HSV
Grab 3rd contour in 3rd binary image
Loop 3rd contour
If area of 3rd contour >100
No
Yes
Calibrate location of polyline
Calculate Euclidean distance
Store data in excel file
Show virtual output

Fig. 1. Algorithm flow chart of proposed approach with three stage of processing for overlaying the Virtual Static Marker and developing Virtual Dynamic line for euclidean distance estimation.

where $C(x,\, y)$ is the object's centroid, m_{00} is moment level zero (the object's region), and m_{10} and m_{01} are moment level one. The fundamental concept is to arbitrarily choose numerous sets of points near the feature points. They are contrasted in terms of their gray-scale values to create a binary feature description. The BRIEF approach is used to acquire binary descriptors following the acquisition of key-points. The initial pixel and second pixel in each sample pair are compared using BRIEF [1]. Initial pixels have a value of 1 if they are brighter than second pixels; otherwise, they have a value of 0. The binary test τ is defined below:

$$\tau(p; x, y) = \begin{cases} 1, p(x) < p(y) \\ 0, p(x) \geq p(y) \end{cases} \tag{12}$$

where $p(x)$ and $p(y)$ represent the intensity values at pixel x and y, is used to carry out this operation. There will be 256 pairings created after repeating this method. A binary descriptor has 32 dimensions when 256 bits are split by 1 byte. To create a binary vector, choose at random n sets of points $(x_i,\, y_i)$. A "descriptor" refers to a vector that identifies an image feature. An example of a BRIEF descriptor is to find the feature descriptors direction. The description now includes the direction data for the feature point represented in (11). Identify a 2 x n matrix Q at any location $(x_i,\, y_i)$ as follows:

$$f_n(p) = \sum_{1 \leq i \leq n} 2^{i-1} \tau\left(p; x_i, y_i\right), \tag{13}$$

where $(x_i,\, y_i)$ are the pixel test point [1]. The guided matrix is created as follows via conversion from the picture patch orientation θ to the associated rotation matrix R_θ:

$$Q = \left[\begin{array}{c} x_1, x_2, x_3, \ldots, x_n \\ y_1, y_2, y_3, \ldots, y_n \end{array} \right], \tag{14}$$

$$Q_\theta = R_\theta Q, \tag{15}$$

The feature descriptor is:

$$g_n(p, \theta) = f_n(p) \mid (x_i, y_i) \in Q_\theta. \tag{16}$$

The correlation between two features in the input and goal images is measured by descriptor distance. Since a binary string serves as the description of the ORB feature, the hamming distance metric is used. Finding the feature point in the target picture with the shortest Hamming distance for every feature point in the reference image will determine if the two feature points are matched. The k-Nearest Neighbours (KNN) technique is used to quickly produce a large number of feature sets in order to increase the effectiveness of two supplied feature sets. Let $p_{ti} = p_{t1}, p_{t2}, \ldots, p_{tn}$ represent the collection of feature points in the goal picture I_{t+1}, and let $p_{ri} = p_{r1}, p_{r2}, \ldots, p_{rn}$ represent the collection of feature points in the template picture I_t.

The KNN method is then merged with the brute-force matchers for $k = 2$, which results in the identification of two closest neighbours in the train picture for each descriptor in the query picture. By using a straightforward selection approach, it aids in the removal of outliers, just as the ratio test does. If a possible set is accepted or rejected, it depends on the ratio between the distances of the nearest and second-nearest matching features. The match is excluded from further investigation if the ratio is greater than the predetermined threshold, which is typically 0.75 chosen for better performance. The rest of the matches are inliers and are arranged according to how far apart the descriptions are from one another. The Random Sample Consensus (RANSAC) method is used to get the best homography transformations between the feature points while keeping the most appropriate pairings and removing other excessive pairings [10]. The homography matrix H between p_{ri} and p_{ti} is written as follows:

$$\begin{bmatrix} u_i' \\ v_i' \\ 1 \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix} \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix}, \tag{17}$$

where $u_i^{'}$, $v_i^{'}$, and $(u_i,\, v_i)$ are the corresponding pixel locations for features $p_{ti},\ p_{ri}$ and typically one of the nine matrix components is assigned a preset unity value $(H_{33} = 1)$, leaving the rest of the eight components unknowable. It results in the decision that a minimum of four inlier sets $(m1, m2)$ are necessary.

The upcoming stages will provide more information on the RANSAC method. After setting the total count of optimal pairings to zero, it randomly select four pairs to use as the first inputs in the posture prediction system for matrix H calculation [10]. For the rest $(h - 4)$ matching sets, the prediction error Δd_j $(j = 1, 2, \ldots, h - 4)$ is calculated as follows;

$$\Delta d_j = \left\| \begin{bmatrix} u_j' \\ v_j' \\ 1 \end{bmatrix} - \mathbf{H} \begin{bmatrix} u_j \\ v_j \\ 1 \end{bmatrix} \right\|. \tag{18}$$

The j^{th} matching pair is identified as an inlier if $\Delta d_j > 3$; if not, it is eliminated as an outlier. Additionally, to boost outlier elimination, the robust RANSAC method is employed. To discover the appropriate features in the train picture for the collection of features in the query picture, such as corners. A poly-line drawn by utilising the perspective transformation, if the homography H is successfully acquired. This process aids for detecting the features of an object.

C. Final phase of processing

After creating a poly-line over the perspective transformation of detected features, it proceeds to the final stage of processing. Here, it repeats the same method that are followed in the first stage of processing, like converting RGB to HSV frame, selecting the colour of the poly-line from HSV frame, obtaining binary frame with the aid of the threshold method, and grabbing the poly-line by considering it as a 3rd contour to retain in the 3rd loop for further processing. If the area of the poly-line is larger than 100 pixels, then it starts to calibrate the location of the poly-line, just like calibrating the location of tissue in the SP, which is performed in first stage of processing. Similarly, if area of the poly-line over than 100 pixels then location of poly-line is required to be calibrated, which are determined as follows;

$$X_c^3 = \frac{\sum_{i=1}^{K_3} X_i^3}{K_3}, \tag{19}$$

$$Y_c^3 = \frac{\sum_{i=1}^{K_3} Y_i^3}{K_3}, \tag{20}$$

$$S_{sw_2} = \frac{1 - P_{sw_{(pixel)}} * 2.54}{100} = 0.0245cm, \tag{21}$$

$$S_{sh_2} = \frac{1 - P_{sh_{(pixel)}} * 2.54}{100} = 0.0245cm, \tag{22}$$

$$S_{sw_{(SP-GP_2)}} = -(0.0036*(S^2_{sw_2}))+(0.7938*(S_{sw_2}))-0.0403, \tag{23}$$

$$S_{sh_{(SP-GP_2)}} = -(0.0068*(S^2_{sh_2}))+(0.0395*(S_{sh_2}))-0.4439, \tag{24}$$

where P_{sw} and P_{sh} stand for the positions of the poly-line over the targeted item in the directions of screen width and height. Here, targeted item is SVM, which is overlayed on the instrument. Thereafter, euclidean distance is estimated between calibrated location of tissue and instrument, which are determined as follows;

$$X_e = ((S_{sw_{(SP-GP_2)}}) - (S_{sw_{(SP-GC_1)}})), \tag{25}$$

$$Y_e = ((S_{sh_{(SP-GP_2)}}) - (S_{sh_{(SP-GC_1)}})), \tag{26}$$

$$e_{disp_{(SP)}} = \sqrt{(X_e)^2 + (Y_e)^2}, \tag{27}$$

where X_e and Y_e denote the x and y axes of the SP, and e_{disp} represents the Euclidean distance between the tissue and the surgical instrument. The virtual output is shown in Fig. 2.

D. Experiment

To test the efficacy of the proposed approach, a customised surgical robot model with one Patient Side Manipulator (PSM) and one Endoscopic Camera Manipulator (ECM) was developed and imported to the Gazebo simulation platform with the setup environment shown in Fig. 2. An abundance of blur filters were applied to test the area of contour range and available feature points on surgical instrument in dynamic frames, which are shown in the Table. I.

TABLE I
COMPARISON BETWEEN FEATURE POINTS AND CONTOUR RANGE UNDER DIFFERENT BLUR SCENES

Type of Blur	Detected features count	Area of contour
Erosion	7	221
Dilation	2	219
Gaussian	7	215
Median	3	215
Bilateral	3	218

Through this test, the obtained number of feature points of surgical instruments in dynamic frames is unfortunately lower than anticipation for each blur filter. Nevertheless, the area of contour range of the surgical instrument is not less than 200 pixels in each blur filter. This is a significant justification to develop and overlay SVM on the contour of the surgical instrument. It is a non-scale-changeable SVM, which helps to obtain a greater number of feature points. Finally, this different approach succeeded in estimating the distance between tissue and SVM; the whole algorithm is explained in the previous section and final output shown in Fig. 2 with simulation environment.

IV. VALIDATION AND RESULT

A comparison study between the proposed work and recently published research has to be carried out in order to verify the suggested work. The instructions provided to the surgical robot for movement trajectory are special, though. As

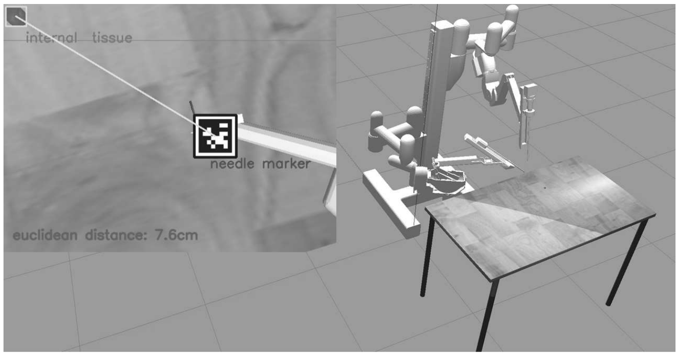

Fig. 2. Customised surgical robot model in simulation environment with proposed approach.

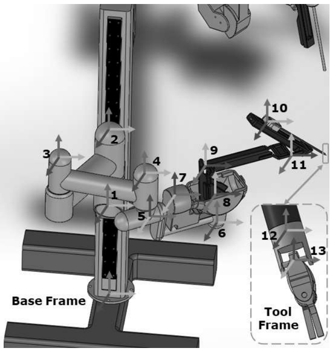

Fig. 3. PSM with coordinate frame.

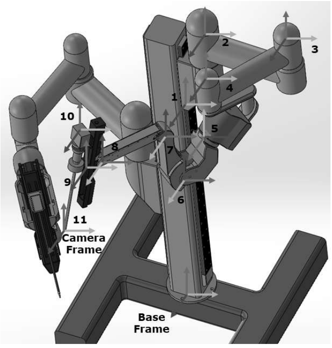

Fig. 4. ECM with coordinate frame.

a result, the proposed work's data are entirely distinct from those of earlier research. Therefore, forward kinematics was chosen to validate the proposed approach. Here, the Denavit-Hartenberg (DH) technique is used to determine each joint's location and angle. To begin with, a DH chain diagram is drawn (shown in Fig. 5) based on a group of coordinate frames aligned over each joint of the customised surgical robot, as shown in the Fig. 3 and Fig. 4. Thus, DH parametric values are easily obtained and deployed in the DH table for PSM, shown in Table. II. These are the DH parameters α_{i-1}, a_{i-1}, θ_i and d_i, which represent link twist, link length, joint angle, and link offset. In addition, Reference Frame, Target frame and Base Frame are represented by the abbreviations of RF, TF and BF, which are used in Table. II and cross multiplication of each matrices.

TABLE II
DH-TABLE FROM RF TO TF

Link	Joint-Type	a_i	α_i	d_i	θ_i
$RF \rightarrow BF$	Fixed	0	0	0	0
$BF \rightarrow PSM_1$	Prismatic	l_1	0	d_1	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$PSM_{12} \rightarrow TF$	Rotation	0	$\frac{\pi}{2}$	0	θ_{13}

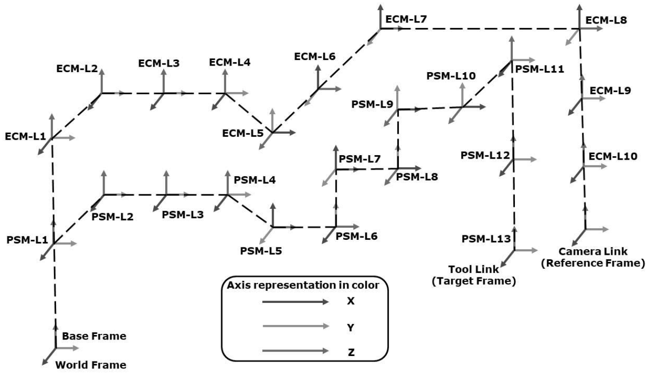

Fig. 5. DH-chain diagram for customised surgical robot from base frame to target frame and reference frame.

From DH parametric values in the DH table, matrices for each joint of PSM are extracted. Thereafter, Homogenous Transformation Matrices (HTM) are carried out by performing cross-multiplication of each joint of matrices from reference (camera) to target frame (surgical instrument), has determined below. Moreover, ECM joints are maintained in a standard posture to obtain a stable camera perspective. Thus, it is not necessary to obtain matrices for each joint in ECM. Various transformations between individual frames are defined by;

$$T_{BF}^{RF} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{28}$$

$$T_{PSM_1}^{BF} = \begin{bmatrix} 1 & 0 & 0 & l_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{29}$$

$$\vdots$$

$$T_{TF}^{PSM_{12}} = \begin{bmatrix} c\theta_{TF} & 0 & -s\theta_{TF} & 0 \\ s\theta_{TF} & 0 & c\theta_{TF} & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{30}$$

where $sin\theta_i$ and $cos\theta_i$ are represented by the abbreviations $s\theta_i$ and $c\theta_i$, respectively. Additionally, d_i represents the prismatic-joint and l_i denotes the link-length. The transformation between TF and RF has determined as;

$$T_{TF}^{RF} = T_{BF}^{RF}(q_0) * T_{PSM_1}^{BF}(q_1) * \ldots * T_{TF}^{PSM_{12}}(q_{13}). \tag{31}$$

When it comes to the joint i (i = 1, 2, …, n), the joint variable is q_i (revolute or prismatic joint) and the final output of the cross multiplication as follows;

$$T_{TF}^{RF} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & p_x \\ R_{21} & R_{22} & R_{23} & p_y \\ R_{31} & R_{32} & R_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{32}$$

where the rotational elements of final output matrix are represented by (k and j =1, 2, and 3), respectively, and by R_{kjs}. Location vector elements in x, y and z axes are represented by p_x, p_y, and p_z.

The values from the proposed approach (SP) are in centimetres. However, values from the (GP) are in metres, which are obtained by performing forward kinematics. Those values are calibrated and converted from metre (m) to centimetre (cms), has been determined as follows;

$$p_x(cms) = \frac{0.099 - p_x(m)}{100}, \tag{33}$$

$$p_y(cms) = \frac{0.043 - p_y(m)}{100}, \tag{34}$$

$$e_{disp_{(GP)}} = \sqrt{(p_x)^2 + (p_y)^2}. \tag{35}$$

Moreover, there is a likelihood of making mistakes because it is a hugely cumbersome task to derive the whole robot kinematics calculation manually or through coding. To troubleshoot this issue, the Robot Operating System (ROS) presented a package named Transform Measuring System (TF). It enables the performance of forward and inverse kinematics for any sort of robot with respect to the number of joints by simply choosing the target and reference frame.

This type of absolute data is required to validate the proposed approach. Therefore, the effectiveness of the proposed approach is compared with absolute data, as depicted in the Fig. 6. The red plot represents the absolute value, which is obtained by performing forward kinematics, and the blue plot denotes the estimated value using the proposed approach. However, there are continuous buffering during simulation, which results in minor gaps between absolute and estimation value, shown in Fig. 6. It could be resolved in future by utilising high computing system. To evaluate the accuracy range of the proposed approach, the coefficient of determination (R^2) is used to analyse values between absolute and estimation using

$$R^2 = \left(1 - \frac{\sum_{i=1}^{n}(f_i - y_i)^2}{\sum_{i=1}^{n}(y_i - y^{'})^2}\right) * (100). \tag{36}$$

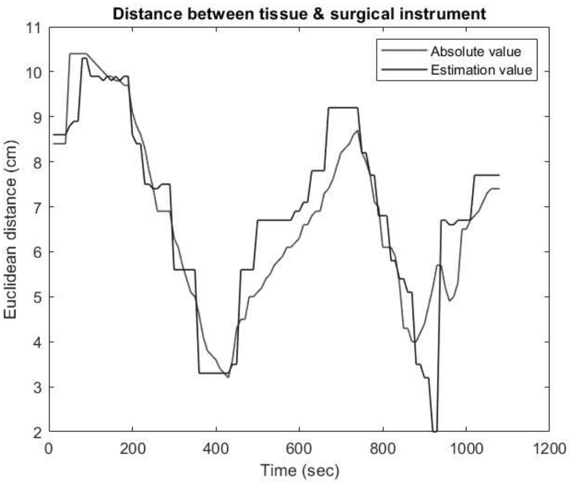

Fig. 6. Comparison between absolute and estimation value for euclidean distance.

Here $y^{'}$ is the average (mean) absolute value, f_i represents the estimated value, y_i represents the absolute value, and n signifies the number of values. Through this calculation, the overall accuracy is 78%. Moreover, this research work was completely performed on a virtual machine-based Ubuntu 18.04 version with 16 GB of RAM and a 6-core processor. Python 2.7 was used as a programming language, and Gazebo 9.0 with ROS 1 was used for simulation.

V. Conclusion

In this work, presented a different approach to estimate the distance between surgical instrument and tissue by extracting features from the SVM using the classic ORB-based features detector. The proposed approach is tested and validated with a developed customised surgical robot model in a Gazebo simulation environment by comparing it with a ROS-based transform measuring system. Through this, the proposed approach performed well in terms of precision in the simulation environment, and the accuracy range is 78%. However, the proposed approach might faces issues in real world. The future plan for this research is to estimate the stitching-pulling and knot-tying forces by developing a neural network based solution.

References

[1] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International conference on computer vision*, pp. 2564–2571, Ieee, 2011.

[2] M. M. Marinho, B. V. Adorno, K. Harada, and M. Mitsuishi, "Active constraints using vector field inequalities for surgical robots," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5364–5371, IEEE, 2018.

[3] A. Banach, K. Leibrandt, M. Grammatikopoulou, and G.-Z. Yang, "Active contraints for tool-shaft collision avoidance in minimally invasive surgery," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 1556–1562, IEEE, 2019.

[4] M. Yoshimura, M. M. Marinho, K. Harada, and M. Mitsuishi, "Single-shot pose estimation of surgical robot instruments' shafts from monocular endoscopic images," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9960–9966, IEEE, 2020.

[5] R. Moccia, C. Iacono, B. Siciliano, and F. Ficuciello, "Vision-based dynamic virtual fixtures for tools collision avoidance in robotic surgery," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1650–1655, 2020.

[6] B. Zhang, S. Wang, L. Dong, and P. Chen, "Surgical tools detection based on modulated anchoring network in laparoscopic videos," *IEEE Access*, vol. 8, pp. 23748–23758, 2020.

[7] M. Robu, A. Kadkhodamohammadi, I. Luengo, and D. Stoyanov, "Towards real-time multiple surgical tool tracking," *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, vol. 9, no. 3, pp. 279–285, 2021.

[8] T. Cheng, W. Li, W. Y. Ng, Y. Huang, J. Li, C. S. H. Ng, P. W. Y. Chiu, and Z. Li, "Deep learning assisted robotic magnetic anchored and guided endoscope for real-time instrument tracking," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3979–3986, 2021.

[9] J. Jaafari, S. Douzi, K. Douzi, and B. Hssina, "Towards more efficient cnn-based surgical tools classification using transfer learning," *Journal of Big Data*, vol. 8, pp. 1–15, 2021.

[10] C. Sun, X. Wu, J. Sun, N. Qiao, and C. Sun, "Multi-stage refinement feature matching using adaptive orb features for robotic vision navigation," *IEEE Sensors Journal*, vol. 22, no. 3, pp. 2603–2617, 2021.