A learning approach for suture thread detection with feature enhancement and segmentation for 3D shape reconstruction

Bo Lu[1], X.B. Yu[2], J.W. Lai[1], K.C. Huang[1], Keith C.C. Chan[2], and Henry K. Chu[1]*, *Member, IEEE*

***Abstract*—A vision-based system presents one of the most reliable methods for achieving an automated robot-assisted manipulation associated with surgical knot tying. However, some challenges in suture thread detection and automated suture thread grasping significantly hinder the realization of a fully automated surgical knot tying. In this paper, we propose a novel algorithm that can be used for computing the 3D coordinates of a suture thread in knot tying. After proper training with our dataset, we built a deep learning model for accurately locating the suture's tip. By applying a Hessian-based filter with multiscale parameters, the environmental noises can be eliminated while preserving the suture thread information. A multistencils fast marching method was then employed to segment the suture thread, and a precise stereo matching algorithm was implemented to compute the 3D coordinates of this thread. Experiments associated with the precision of the deep learning model, the robustness of the 2D segmentation approach, and the overall accuracy of 3D coordinate computation of the suture thread were conducted in various scenarios, and the results quantitatively validate the feasibility and reliability of the entire scheme for automated 3D shape reconstruction.**

***Note to Practitioners* – This paper was motivated by the challenges of suture thread detection and 3D coordinate evaluation in a calibrated stereo-vision system. To precisely detect the suture thread with no distinctive feature in an image, additional information, such as the two ends of the suture thread or its total length, are usually required. This paper suggests a new method utilizing a deep learning model to automate the tip detection process, eliminating the need of manual click in the initial stage. After feature enhancements with image filters, a multistencils fast marching method was incorporated to compute the arrival time from the detected tip to other points on the suture contour. By finding the point that takes the maximal time to travel in a closed contour, the other end of the suture thread can be identified, thereby allowing suture threads of any length to be segmented out from an image. A precise stereo matching method was then proposed to generate matched key points of the suture thread on the image pair, thereby enabling the reconstruction of its 3D coordinates. The accuracy and robustness of the entire suture detection scheme were validated through experiments with different backgrounds and lengths. This proposed scheme offers a new solution for detecting curvilinear objects and their 3D coordinates, which shows potential in realizing automated suture grasping with robot manipulators.**

***Index Terms*—Surgical robot, suture thread detection, stereo-vision, 3D coordinates computation.**

I. Introduction

ROBOT-assisted manipulation (RAM) is a widely employed technology in minimally invasive surgery (MIS) [1] that allows surgeons to perform precise operations with high dexterity [2], manipulability [3], and complexity [4]. These manipulating systems are manually controlled by surgeons through a master-slave device, and real-time 2D images are available for monitoring [5]. However, despite assistance from robotic systems, surgeons may feel tired after a long period of operation because staring at the screen for an extended period can cause eye sore and irritation, which in turn may lead to misoperations and secondary trauma to tissues [6]. To improve the performance of RAM, automated manipulation can be incorporated into some low-level operations in surgery [7] by standardizing the procedure of path planning [8] or by improving robustness via visual [9] and force feedback [10]. Reducing the amount of human intervention can improve the overall operating efficiency of RAM.

Surgical knot tying is a popular manipulation that can be performed by robots in an automated or semi-automated manner. The operation begins with suture threading, in which a suture is stitched through the wound automatically by using robotic arms. Sen et al. proposed a multi-throw stitching method based on a sequential convex programming by using a 3D printed suture needle angle positioner [11]. Pedram et al. developed a kinematic model that describes the needle-tissue interaction and can be used for constant curvature needle path planning [12] by using information about the tissue geometry, the surgeon-defined entry/exit points, and the optimization weighting factors.

To tighten a surgical knot, suture loops must be formed with the suture thread. To form a high-quality loop, trajectory planning and the optimization of the control scheme have been studied. Osa et al. proposed a framework that can learn the spatial motion and force information from previous trials for online trajectory planning and automated suture looping in a dynamic environment [13]. Lu et al. generated a space-efficient trajectory for robot-assisted suture looping manipulation [9] and found that the entire operation can achieve sufficient accuracy and robustness by using a visual guide, an online optimizer, and a dynamic LQ controller.

However, the works mentioned in [11] [12] have mainly focused on the stitching part of the process. Afterward, the

[1] B. Lu, H.K. Chu, K.C. Huang, and J.W. Lai are with the Department of Mechanical Engineering, The Hong Kong Polytechnic University, Hong Kong. Email: bo.lu@connect.polyu.hk (B. Lu), henry.chu@polyu.edu.hk (H. K. Chu). *Corresponding author.

[2] X.B. Yu and Keith C.C. Chan are with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong.

[3] This work was supported in part by the Research Grant Council of the Hong Kong Special Administrative Region, China, under Grant 25204016.

long suture thread needs to be trimmed in order to form loops before tying a surgical knot. Owing to its flexible property, the suture thread may form various shapes arbitrarily on the tissue or the skin surface. To automate the knot tying task, the 3D coordinates of the suture thread must be accurately determined to facilitate its grasping operation, the suture thread stitching, and the subsequent looping manipulation using robots.

An RGB-D camera is often used to obtain the 3D information of the suture thread. Mohanarajah used the point cloud by running a dense odometry algorithm in collaborative robots and achieved a 3D visualization of the surroundings [14]. Stereo-camera is another popular paradigm for 3D scene reconstruction. Ren proposed a dual-camera based method that uses markers for surgical tracking [15]. However, these approaches either rely on feature descriptors or markers, which can hardly cope with the detection of a long, flexible, and featureless surgical suture thread.

To address this challenge, the segmentation and stereo-pairs matching of a curvilinear object in a stereo-camera system are necessary. Jackson [16] proposed a stereo-image-based algorithm that detects the suture thread by using a B-spline model. By minimizing the image matching energy, they can track the suture thread in real time. Obara et al. proposed a contrast-independent method that uses the phase congruency tensor to segment the curvilinear object from noisy biomedical images [17]. Benmansour et al. adopted the fast marching method (FMM) to identify the closed contour or surface with two endpoints in both 2D and 3D images [18]. Kaul also adopted the FMM principle to detect both open and closed curves with a single starting input [19].

In recent years, deep learning approaches have been used extensively for accurate object detection. For instance, Cha et al. proposed a vision-based inspection method [20] for detecting cracks on concrete images by using a convolutional neural network (CNN). They later improved this algorithm to the Faster Region-based CNN (Faster R-CNN) for detecting multiple types of damages [21]. Chen et al. combined CNN and Naive Bayes data fusion [22] for inspecting the cracks of a nuclear power plant. Their work mainly focused on detecting crack features on a 2D image where object segmentation is not required for its length evaluation. In this work, a recurrent neural network (RNN) was employed for tip detection in a temporal sequence of 2D images (or to model the temporal dependency) [23], and several new algorithms were implemented for suture length evaluation and 3D shape computation.

In this paper, we present a novel approach for constructing the 3D model of a surgical suture thread in an arbitrary environment. First, to reduce the need for manually clicking on two ends of a suture, we employ a neural network to automatically locate the suture's tip on the image. Given that tips with less distinctive features can easily result in ambiguous matches in the nearby region, a long short-term memory (LSTM) was incorporated to avoid a repeated detection of the same object. To address the challenges in finding flexible and thin sutures for segmentation, a Hessian-matrix-based filter was employed for feature enhancement and noise suppression. Owing to the ambiguity of the Hessian eigenvectors' directions in detecting curvilinear objects [17] [24], a numerical multistencils fast marching method (MFMM) [25] was applied to compute the arrival time map, which determines the minimal arrival time from each pixel in the image to the suture thread's tip. Using this information, the object can be completely segmented along the object contour in a 2D plane. While the method proposed in [18] [26] requires additional information such as the endpoint or the object length, this paper evaluates the end (turning) point by finding the point that takes the longest time to reach from the tip in a closed contour. By combining the map with the proposed front and back propagation, precise key points can be assigned, matched, and visually corrected [27] along the suture thread for the 3D computation on the left and right images. As shown in Fig. 1, by using stereo images as inputs, the 3D coordinates of the suture thread from one end to the other can be evaluated and outputted accordingly.

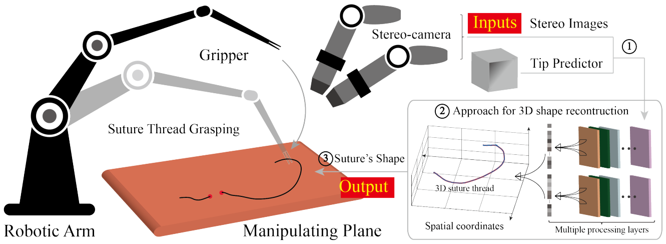


Fig. 1: System illustration for suture thread 3D reconstruction.

The rest of this article is organized as follows. Section II introduces the deep learning model for the suture's tip detection. In Section III, the procedures for highlighting and segmenting the curvilinear object will be proposed. Section IV presents the principle of the MFMM and its implementation for suture detection. Besides, the segmentation of the suture thread, its key point generation, and the stereo matching will be introduced. Section V provides the experimental results of the tip point detection using the deep learning model, the robustness of 2D suture thread segmentation, and the accuracy of computing its 3D coordinates under various environments. Section VI concludes this paper.

II. Initialization using the Deep Learning Approach

A. Deep Learning Model for Suture Thread's Tip Detection

To locate the suture's tip in an image, we used the deep learning-based object detection model proposed by Stewart et al. [23], which uses a CNN ImageNet for feature encoding and an RNN with LSTM units to sequentially decode bounding boxes as shown in Fig. 2.

The advantage of ImageNet lies in its representational power to encode multiple objects, thereby providing enough robustness for extracting a single suture's tip. LSTM can memorize the previous output of the bounding box and then feed this output as input to the next layer for modeling the generation sequence of the next bounding box. Given that the generated sequence will not be duplicately produced in the later stage when conducting object detection, the whole procedure can proceed effectively.

To reduce the computation cost, the image is compressed into 640×480 pixels, and then a new bounding box and the confidence that this box includes the undetected suture thread's tip will be computed as the output in each layer of LSTM.

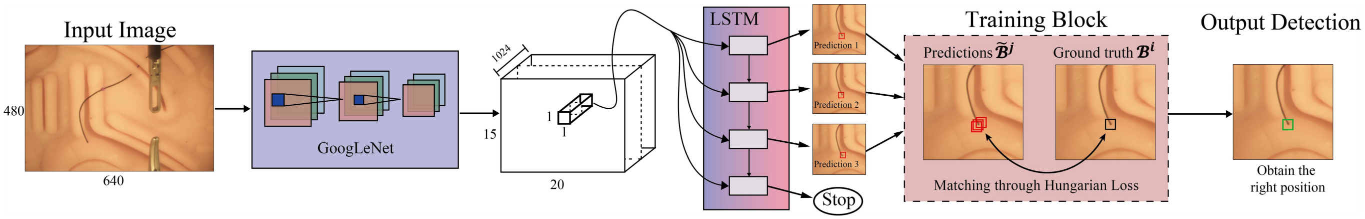

Input Image
480
640
GoogLeNet
1024
15
20
1
1
LSTM
Prediction 1
Prediction 2
Prediction 3
Stop
Training Block
Predictions $\tilde{\mathcal{B}}^j$
Ground truth $\mathcal{B}^i$
Matching through Hungarian Loss
Output Detection
Obtain the right position

Fig. 2: The general framework of the deep learning model. Using our own dataset to train the model, the LSTM can act as a controller for the suture thread's tip detection. With the implementation of the Hungarian Loss, the optimal position of the suture thread's tip can be picked out from the image.

Boxes are produced in a descending confidence order, and each new detection will be conditioned based on the previous one. By using a pre-specified threshold, the detection is terminated when the LSTM is unable to find a new box with a confidence above the threshold value. Until this stage, the output sequence can be collected and presented as a final description of all object instances in the region. In our task, the desired target is a single suture thread's tip. Given that LSTM will output a prediction set of bounding boxes as shown in Fig. 2, we employed an optimizer to find the optimal position of the suture's tip from the detected candidates.

B. Optimization Using Loss Function

By assuming the outputs of the collected sequential candidates as $\mathcal{O}$ which is shown in Fig. 2, we utilized the Hungarian algorithm [23] to model the bipartite matching between the predicted outputs $\mathcal{O}$ and the ground truth $\mathcal{G}$ in polynomial time. The loss function associated with the Hungarian algorithm which models the distance between the predicted outputs $\mathcal{O} = \left\{\tilde{\mathcal{B}}^j | j = 1, 2, ..., N\right\}$ and the ground truths $\mathcal{G} = \left\{\mathcal{B}^i | i = 1, 2, ..., N\right\}$ follows

$$\begin{aligned} L(\mathcal{G}, \mathcal{O}, f) = \nu \sum_{i=1}^{|\mathcal{G}|} l_{pos}(\mathcal{B}^i_{pos}, \tilde{\mathcal{B}}^{f(i)}_{pos}) + \sum_{j=1}^{|\mathcal{O}|} l_c(\tilde{\mathcal{B}}^j_c, y_j) \\ + \sum_{i=1}^{|\mathcal{G}|} \sum_{j=1}^{|\mathcal{O}|} \Delta(\mathcal{B}^i, \tilde{\mathcal{B}}^j) \end{aligned} \tag{1}$$

where the localization accuracy is equal to $l_{pos} = ||\mathcal{B}^i_{pos} - \mathcal{B}^{\tilde{f}(i)}_{pos}||_1$ and represents the displacement between the positions of the ground truth and the candidates. The detection confidence l_c determines the cross-entropy loss on a candidate's confidence to the match of the ground truth, and f denotes the bipartite matching between $\mathcal{O}$ and $\mathcal{G}$ in polynomial time as obtained by the Hungarian algorithm.

III. Image Filter For Curvilinear Object Segmentation

With the detected tip, the following step is to segment the suture thread from the background. In various conditions, an image may contain undesired noises that need to be filtered using proper image filters before the object segmentation.

Several image denoising and enhancement filters have been comprehensively developed, including the smooth filter [28], Gaussian filter [29], max-min filter [30], and weighted median filter [31]. However, these filters do not show satisfactory performance in segmenting curvilinear objects. In order to enhance the curvilinear information, the second derivatives with the Gaussian kernel are implemented [32].

For a stereo camera, the 2D left and right frames can be denoted as $\mathbb{C}_l$ and $\mathbb{C}_r$, and their respective image domains can be denoted as $\Omega_l, \Omega_r \rightarrow \mathbb{R}^+$, where the subscripts l and r denote the left and right cameras, respectively. To distinguish curvilinear features at various conditions, a multiple scale parameter σ_i was implemented.

By taking one camera frame for example, after transforming the color image into grayscale, the local behavior of point $\mathbf{p}_i \in \Omega$ can be determined as its Taylor expansion as

$$\mathbb{C}(\mathbf{p}_i + \Delta\mathbf{p}_i, \sigma_i) \approx \mathbb{C}(\mathbf{p}_i, \sigma_i) + \delta\mathbf{p}_i^T \nabla_{i,\sigma_i} + \delta\mathbf{p}_i^T \mathcal{H}_{i,\sigma_i} \delta\mathbf{p}_i \tag{2}$$

where the expansion approximates the image structure up to the second order derivatives, while ∇_{i,σ_i} and $\mathcal{H}_{i,\sigma_i}$ denote the gradient vector and the corresponding Hessian matrix that are computed at scale σ_i [33]. To highlight the curvilinear information, we processed the image by applying a Gaussian filter with a size of $(3\sigma_i + 1) \times (3\sigma_i + 1)$. The 2D Gaussian filter is defined as

$$G(\mathbf{p}_{gau}, \sigma_i) = \frac{1}{2\pi\sigma_i^2} \cdot \mathrm{e}^{-\frac{||\mathbf{p}_{gau}||^2}{2\sigma_i^2}} \tag{3}$$

where $\mathbf{p}_{gau} = [\mathbf{X}_G, \mathbf{Y}_G]$ is the image coordinates of the filter. Based on Eq. (2), the third term of the Taylor series gives the second derivatives of image $\mathbb{C}$, with $\mathcal{H}$ being the Hessian matrix, which can be expressed as

$$\mathcal{H}(\sigma_i) = \begin{bmatrix} \boldsymbol{I}_{xx}(\sigma_i) & \boldsymbol{I}_{xy}(\sigma_i) \\ \boldsymbol{I}_{xy}(\sigma_i) & \boldsymbol{I}_{yy}(\sigma_i) \end{bmatrix} \tag{4}$$

According to the scale-space theory, the Hessian matrix can be represented by the convolution between the second derivatives of the Gaussian filter and the image as follows

$$\begin{cases} \boldsymbol{I}_{xx}(\sigma_i) = \dfrac{\partial^2}{\partial x^2}\mathbb{C}(\sigma_i) = \sigma_i^\gamma \cdot \dfrac{\partial^2}{\partial x^2} G(\mathbf{p}_{gau}, \sigma_i) \otimes \mathbb{C}(\sigma_i) \\ \boldsymbol{I}_{xy}(\sigma_i) = \dfrac{\partial^2}{\partial x \partial y}\mathbb{C}(\sigma_i) = \sigma_i^\gamma \cdot \dfrac{\partial^2}{\partial x \partial y} G(\mathbf{p}_{gau}, \sigma_i) \otimes \mathbb{C}(\sigma_i) \\ \boldsymbol{I}_{yy}(\sigma_i) = \dfrac{\partial^2}{\partial y^2}\mathbb{C}(\sigma_i) = \sigma_i^\gamma \cdot \dfrac{\partial^2}{\partial y^2} G(\mathbf{p}_{gau}, \sigma_i) \otimes \mathbb{C}(\sigma_i) \end{cases} \tag{5}$$

where $\otimes$ is the image convolutional operator. We also added a parameter γ to normalize the derivatives following the suggestions of Lindeberg [34]. This parameter is restricted as $0 < \gamma < 3$ [35]. To denote the second derivatives of the Gaussian filter, $\boldsymbol{h}$ can be computed with respect to scale σ_i as

$$\boldsymbol{h}(\sigma_i) = \begin{bmatrix} \mathbf{h}_{xx}(\sigma_i) \\ \mathbf{h}_{xy}(\sigma_i) \\ \mathbf{h}_{yy}(\sigma_i) \end{bmatrix} = \begin{bmatrix} \frac{1}{2\pi \cdot \sigma_i^4} \cdot (\frac{\mathbf{X}_G^2}{\sigma_i^2} - 1) \cdot \mathrm{e}^{\frac{-(\mathbf{X}_G^2 + \mathbf{Y}_G^2)}{2\sigma_i^2}} \\ \frac{1}{2\pi \cdot \sigma_i^6} \cdot \boldsymbol{X}_G \cdot \boldsymbol{Y}_G \cdot \mathrm{e}^{\frac{-(\mathbf{X}_G^2 + \mathbf{Y}_G^2)}{2\sigma_i^2}} \\ \mathbf{h}_{xx}^T(\sigma_i) \end{bmatrix} \tag{6}$$

For dark lines on a bright background, the Gaussian kernel returns large positive values across the line and small (positive or negative) values along the line, which can be explained by the eigenvalues of the Hessian matrix [35].

The line structures can be detected by analyzing the Hessian matrix and its eigenvalues. Let the eigenvalues of $\mathcal{H}(\sigma_i)$ be $\lambda_1(\sigma_i)$ and $\lambda_2(\sigma_i)$, and the filtered image $\mathbb{T}(\sigma_i)$ of $\mathbb{C}(\sigma_i)$ can be computed as

$$\mathbb{T}(\sigma_i) = \begin{cases} 0 & \lambda_2(\sigma_i) < 0 \\ \mathrm{e}^{-\frac{\mathcal{R}^2_{b,\sigma_i}}{2\beta^2}} \cdot \left(\boldsymbol{I} - \mathrm{e}^{-\frac{\mathcal{S}^2_{\sigma_i}}{2\mathcal{C}^2}}\right) & otherwise \end{cases} \tag{7}$$

where $\mathcal{R}_{b,\sigma_i} = \lambda_1(\sigma_i)/\lambda_2(\sigma_i)$, $\mathcal{S}_{\sigma_i} = \sqrt{\lambda_1(\sigma_i) + \lambda_2(\sigma_i)}$, β and $\mathcal{C}$ are the threshold parameters that control the sensitivity of the filter to the measures of $\mathcal{R}_b$ and $\mathcal{S}$, respectively [36]. By repeating the above procedures with different scale parameters σ_i, the optimal image can be chosen [34] at the local maxima as follows

$$\mathbb{T} = \max_{\sigma_{\min} \leq \sigma_i \leq \sigma_{\max}} \mathbb{T}(\sigma_i) \tag{8}$$

By setting a proper threshold, $\mathbb{T}$ can be converted into a binary image in which the surrounding noises are initially eliminated.

IV. Suture Thread Segmentation and Key Point Matching

To compute the 3D coordinates of a suture thread, sufficient numbers of key points and their respective stereo pairs should be obtained. Therefore, an accurate description of the suture thread from its tip to the end is required.

To achieve this goal, the minimal action map [18] was computed by using the images $\mathbb{T}$ captured from the camera frame. Within a 2D image, its form of energy E can be expressed as

$$E(\phi) = \int_{\phi} \{\mathcal{P}(\phi(\varepsilon)) + w\} d\epsilon = \int_{\phi} \widetilde{\mathcal{P}}(\phi(\varepsilon) d\epsilon \tag{9}$$

where ϕ denotes the curve within the image domain $\Omega \to R^+$, and $\mathcal{P}$ is the potential image of the input image $\mathbb{T}$. The minimal action map highlights the curvilinear information, and the choice of $\mathcal{P}$ depends on the specific application. In our task, we set $\mathcal{P}$ as

$$\mathcal{P} = (s \cdot \nabla \mathbb{T} + \xi)^{-4} \tag{10}$$

where s is a parameter that highlights the variation along the suture thread boundary, and ξ and ω in Eqs. (9) and (10) are small positive constants that can be tuned by users. The value of $(1/\widetilde{\mathcal{P}})$ represents the traveling speed when crossing each pixel. Therefore, our potential function aims to set the boundary of the suture thread with a high traveling speed, while the background pixels with slow traveling velocities.

By treating the detected tip $\mathbf{p}_t$ as the source point, for any point $\mathbf{p}_i \in \Omega$, the minimal path denotes the trajectory along which the propagation time between $\mathbf{p}_t$ and $\mathbf{p}_i$ obtains the minimal value. Consequently, the curve ε_i between tip $\mathbf{p}_t$ and point $\mathbf{p}_i$ within domain Ω that can minimize the energy function E can be determined as

$$\forall \mathbf{p}_i \in \Omega, \mathcal{M}_{(\mathbf{p}_i)} = \min_{\phi \in Path_{\mathbf{p}_t, \mathbf{p}_i}} \left\{ \int_{\phi} \widetilde{\mathcal{P}}(\phi(\varepsilon) d\epsilon \right\} \tag{11}$$

where $Path_{\mathbf{p}_t,\mathbf{p}_i}$ denotes all paths that connect point $\mathbf{p}_i$ to source point $\mathbf{p}_t$ within Ω. Traveling along a minimal time path, $\mathcal{M}(\mathbf{p}_i)$ denotes the arrival time between $\mathbf{p}_i$ and $\mathbf{p}_t$. When traveling across point $\mathbf{p}_j$, the velocity only depends on the value $(1/\widetilde{\mathcal{P}}_{(\mathbf{p}_j)})$ [37] and satisfies the Eikonal equation as

$$\begin{cases} ||\nabla \mathcal{M}_{(\mathbf{p}_i)}|| = \widetilde{\mathcal{P}}_{(\mathbf{p}_i)}, & \mathbf{p}_i \in \Omega \\ \mathcal{M}_{(\mathbf{p}_t)} = 0 \end{cases} \tag{12}$$

For each point $\mathbf{p}_i$ within the image, the original pixel value can be substituted by its arrival time $\mathcal{M}_{(\mathbf{p}_i)}$. Therefore, the arrival time map of the corresponding input image can be obtained. To achieve this goal, the fast matching method (FMM) developed in [25] was incorporated into our scheme.

A. General Principle and Derivation of FMM

To obtain the arrival time from any point to the source point, the method proposed by Rouy along with the solution of the correct viscosity was implemented [38]. For point $[i,j] \in \Omega$, the equation approximated using the first-order finite difference in a discrete format can be obtained as

$$\begin{aligned} &\max\left\{\frac{\mathcal{M}_{(i,j)} - \mathcal{M}_x}{\triangle_x}, 0\right\}^2 + \\ &\max\left\{\frac{\mathcal{M}_{(i,j)} - \mathcal{M}_y}{\triangle_y}, 0\right\}^2 = \widetilde{\mathcal{P}}^2(i,j) \end{aligned} \tag{13}$$

where $\triangle_x$ and $\triangle_y$ denote the pixel spacing along horizontal and vertical directions within the image. We also have:

$$\begin{aligned} \mathcal{M}_x &= \min\left\{\mathcal{M}_{(i-1,j)}, \mathcal{M}_{(i+1,j)}\right\} \\ \mathcal{M}_y &= \min\left\{\mathcal{M}_{(i,j-1)}, \mathcal{M}_{(i,j+1)}\right\} \end{aligned} \tag{14}$$

The solution of Eq. (13) can be solved as:

1. $\mathcal{M}_{(i,j)} > \max(\mathcal{M}_x, \mathcal{M}_y)$, where $\mathcal{M}_{(i,j)}$ is the maximum solution of the quadratic equation: $\mathcal{M}_{(i,j)} = \frac{(\mathcal{M}_x + \mathcal{M}_y) + \sqrt{2\widetilde{\mathcal{P}}^2(i,j) - (\mathcal{M}_x - \mathcal{M}_y)^2}}{2}$;
2. $\mathcal{M}_x > \mathcal{M}_{(i,j)} > \mathcal{M}_y$, $\mathcal{M}_{(i,j)} = \mathcal{M}_y + \widetilde{\mathcal{P}}(i,j)$; and
3. $\mathcal{M}_y > \mathcal{M}_{(i,j)} > \mathcal{M}_x$, $\mathcal{M}_{(i,j)} = \mathcal{M}_x + \widetilde{\mathcal{P}}(i,j)$.

To finalize the FMM process, all pixels in the entire image domain were labeled with the following sets:

1. ***Alive*** Set $\mathbf{p}_{Alive}$: For points $\mathbf{p}_i$ belong to this set, the arrival times $\mathcal{M}_{\mathbf{p}_i}$ have been computed and fixed;
2. ***Trial*** Set $\mathbf{p}_{Trial}$: Four neighbors around the alive set, and their arrival times may be changed in the later iteration;
3. ***Far*** Set $\mathbf{p}_{Far}$: For those points that have not been reached, and their arrival times have not been calculated.

The general process of FMM is summarized in Algorithm 1. First, the source point is sorted as the alive set. Afterward, the four neighbors of the alive set are treated as the trial set, and their arrival times can be computed. The point with the minimal action $\mathcal{M}$ among the trial set $\mathcal{M}_{\mathbf{p}_{Trial}}$ can be updated to the alive set. Then, loop the above procedure, and one new point can be updated into the alive set in each iteration. This procedure is terminated when the arrival times of all points within the image domain are determined.

However, FMM only makes use of four neighbors but neglects the diagonal information. In this case, the arrival time amp may not be accurate enough. Moreover, the solution

Algorithm 1: Fast Marching Method for the 2D image

```
Data: Image $\mathbb{T}_l$ and $\mathbb{T}_r$, source points $\mathbf{p}_{t,l}$ and $\mathbf{p}_{t,r}$
For each image, the tip of the suture thread is labeled as the alive set, which is initialized as $\mathcal{M}_{\mathbf{p}_t} = 0$;
Four neighbors of $\mathbf{p}_t$ are labeled as the trial set, and their corresponding arrival times $\mathcal{M}_{\mathbf{p}_{Trial}}$ can be computed using Eq. (13);
Set other points as the far set $\mathbf{p}_{far}$: $\mathcal{M}_{\mathbf{p}_{far}} = \infty$;
Pick the local minima of $\mathcal{M}_{\mathbf{p}_{Trial}}$, and sort the corresponding point to the alive set;
Loop the procedure: while $\boldsymbol{p}_{far} \neq \varnothing$ do
    Exam the four neighbors of the all alive points;
    if Neighbors $\notin \boldsymbol{p}_{Alive}$ then
        Update the arrival time $\mathcal{M}_{\mathbf{p}_{Trial}}$ using Eq. (13);
        Move point which has the local minima among $\mathcal{M}_{\mathbf{p}_{Trial}}$ to the alive set $\mathbf{p}_{Alive}$;
Result: The map of arrival time $\mathcal{M}_{\mathbf{p}_l}$ and $\mathcal{M}_{\mathbf{p}_r}$ with respect to the source point
```

of Eq. (13) only uses the first-order approximation, which introduces additional errors when calculating the arrival time map. These shortcomings potentially harm the accuracy of the upcoming suture thread segmentation.

B. Arrival Time Map Computation Using Multistencils FMM

To achieve a higher precision and resolve the aforementioned problems, an enhanced method called multistencils FMM [25] was adopted to compute the arrival time map $\mathcal{M}$ while considering the diagonal information.

Consider that the natural Cartesian coordinate is rotated with an angle α as shown in Fig. 3 (b) and (d). For any point (x, y) in the natural coordinate, its new position $(x^{'}, y^{'})$ in the rotated coordinate can be denoted as

$$\begin{aligned} x^{'} &= xcos\alpha + ysin\alpha \\ y^{'} &= -xsin\alpha + ycos\alpha \end{aligned} \tag{15}$$

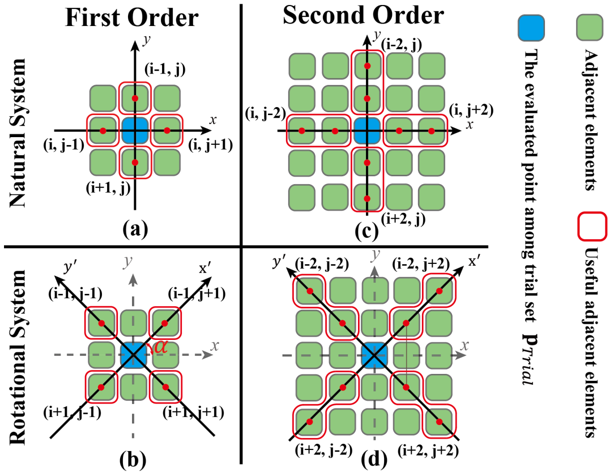


Fig. 3: The normal coordinate (a) using the first-order and (c) second-order approximation; the rotated coordinate (b) using the first-order and (d) second-order approximation.

Using the chain rule, the partial derivatives of the arrival time $\mathcal{M}$ with respect to x and y can be derived as

$$\begin{aligned} \mathcal{M}_x &= \frac{\partial \mathcal{M}}{\partial x} = \frac{\partial \mathcal{M}}{\partial x'} \cdot \frac{\partial x^{'}}{\partial x} + \frac{\partial \mathcal{M}}{\partial y'} \cdot \frac{\partial y^{'}}{\partial x} \\ \mathcal{M}_y &= \frac{\partial \mathcal{M}}{\partial y} = \frac{\partial \mathcal{M}}{\partial x'} \cdot \frac{\partial x^{'}}{\partial y} + \frac{\partial \mathcal{M}}{\partial y'} \cdot \frac{\partial y^{'}}{\partial y} \end{aligned} \tag{16}$$

By substituting Eq. (15) into Eq. (16) and letting the rotational angle be 45°, we can easily obtain

$$\begin{aligned} &\mathcal{M}_x^2 + \mathcal{M}_y^2 = \frac{\partial^2 \mathcal{M}}{\partial x'^2} + \frac{\partial^2 \mathcal{M}}{\partial y'^2} = \mathcal{M}_{x'}^2 + \mathcal{M}_{y'}^2 \\ &\Rightarrow \widetilde{\mathcal{P}}_{(x,y)}^2 = |\nabla \mathcal{M}_{(x',y')}|^2 \end{aligned} \tag{17}$$

It is noticed that Eq. (17) is also the Eikonal equation, which indicates that the $\nabla \mathcal{M}_{(x',y')}$ in the rotational coordinate can be computed by solving the Eikonal equation along the new directions $x^{'}$ and $y^{'}$. Therefore, the arrival time along the diagonal direction can also be solved by following the FMM principle in a rotational system. For point $[i, j] \in \Omega$, we have

$$\sum_{\zeta=1}^{2} \max \left\{ \mathcal{C}_\zeta \cdot \frac{\mathcal{M}_{(i,j)} - \mathcal{M}_\zeta}{\triangle_\zeta}, 0 \right\}^2 = \widetilde{\mathcal{P}}^2(i, j) \tag{18}$$

where $\mathcal{C}_\zeta$ denotes the approximation coefficient, and $\triangle_\zeta$ is the adjacent pixel distance. For a natural system, $\triangle_\zeta$ is equal to 1, whereas for a rotational system, it is equal to $\sqrt{2}$. To further improve the accuracy of the arrival time map, the following second order approximation proposed by [39] was adopted

$$\begin{cases} \mathcal{M}_x = \min\limits_{a_x \in \{1,-1\}} \left[\dfrac{4\mathcal{M}_{(i+a_x,j)} - \mathcal{M}_{(i+2\cdot a_x,j)}}{3} \right] \\ \mathcal{M}_y = \min\limits_{a_y \in \{1,-1\}} \left[\dfrac{4\mathcal{M}_{(i,j+a_y)} - \mathcal{M}_{(i,j+2\cdot a_y)}}{3} \right] \end{cases} \tag{19}$$

The selection of the parameters is summarized as follows:

- Adopting the first-order approximation, $\mathcal{C}_\zeta = 1$, $\mathcal{M}_\zeta$ can be solved by using two adjacent neighbors as shown in Figs. 3 (a) and (b) and by using Eq. (14).
- Adopting the second-order approximation, $\mathcal{C}_\zeta = 3/2$, $\mathcal{M}_\zeta$ can be solved by using Eq. (19), whereas the two-pixel away points are provided as shown in Figs. 3 (c) and (d).

Eq. (18) can then be transformed as

$$\Upsilon_{(1)} \mathcal{M}_{(i,j)}^2 + \Upsilon_{(2)} \mathcal{M}_{(i,j)} + \Upsilon_{(3)} = \widetilde{\mathcal{P}}^2(i, j) \tag{20}$$

where coefficients $\Upsilon_{(1)}$, $\Upsilon_{(2)}$, and $\Upsilon_{(3)}$ are listed in Table I.

TABLE I: Values of three Υ

Coefficient		1st-order	2nd-order
$\Upsilon_{(1)}$	Natural	2	9/8
	Rotated	1	9/16
$\Upsilon_{(2)}$	Natural	$-2(\mathcal{M}_x + \mathcal{M}_y)$	$-\frac{9}{8}(\mathcal{M}_x + \mathcal{M}_y)$
	Rotated	$-\mathcal{M}_{x'} - \mathcal{M}_{y'}$	$-\frac{9}{16}(\mathcal{M}_{x'} + \mathcal{M}_{y'})$
$\Upsilon_{(3)}$	Natural	$\mathcal{M}_x^2 + \mathcal{M}_y^2$	$\frac{9}{4}(\mathcal{M}_x^2 + \mathcal{M}_y^2)$
	Rotated	$\frac{1}{2}(\mathcal{M}_{x'}^2 + \mathcal{M}_{y'}^2)$	$\frac{9}{8}(\mathcal{M}_{x'}^2 + \mathcal{M}_{y'}^2)$

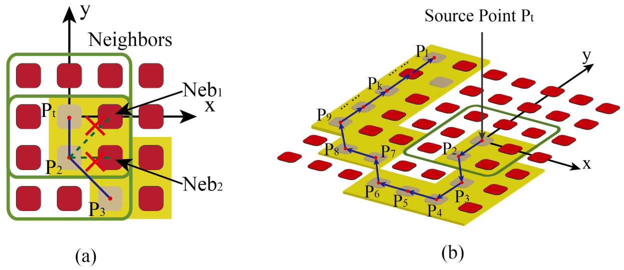
Therefore, the arrival time of $\mathcal{M}_{(i,j)}$ can be computed as the minimal positive solution of Eq. (20) as follows

$$\mathcal{M}_{(i,j)} = \frac{-\Upsilon_{(2)} \pm \sqrt{\Upsilon_{(2)}^2 - 4\Upsilon_{(1)}(\widetilde{\mathcal{P}}^2(i,j) - \Upsilon_{(3)})}}{2\Upsilon_{(1)}} \tag{21}$$

The expansion procedures of FMM and MFMM when calculating the arrival time map are shown in the Supplementary Material. Starting from the source point, FMM only expands in the horizontal and vertical directions, whereas MFMM preserves this property while adding the diagonal expansion. As a result, the narrow band formed by $\mathbf{p}_{Trial}$ is larger in MFMM than that in FMM. When using the arrival time map to segment the suture thread, FMM only gives the propagation path in either the horizontal or vertical direction, whereas MFMM allows the diagonal shape, thereby increasing the precision of reconstructing the 3D shape of the suture thread.

C. Segmentation and Point Matching of the Suture Thread

With the rectified stereo-camera [40], the arrival time map $\mathcal{M}$ and suture thread's tip position $\mathbf{p}_t$ can be obtained. A convenient and accurate stereo matching algorithm was then built for the suture thread's segmentation and matching. Starting from tip point $\mathbf{p}_t$, the potential function gives a larger traveling velocity $(1/\widetilde{\mathcal{P}})$ along the suture contour, and the desired neighbor that connects the tip should be the point that possesses the minimal value $\mathcal{M}_{\min}$.



(a) (b)

Fig. 4: Selection of the suture thread contour point using the minimal increasing principle.

As illustrated in Fig. 4, the yellow color denotes part of the contour area that has a high traveling speed. Starting from the suture thread's tip $\mathbf{p}_t$, its eight neighbors are initially evaluated as shown Fig. 4 (a). The second pixel $\mathbf{p}_2$ was selected based on the minimal increasing principle. It is noticed, Neb_1 and Neb_2 are two neighbors centered at $\mathbf{p}_t$ and $\mathbf{p}_2$. When evaluating the eight neighbors of $\mathbf{p}_2$, Neb_1 or Neb_2 can be selected as connecting elements, which can lead to a trapped zigzag or even a false segmentation result.

To eliminate the potential mistake and the unfavorable zigzag shapes, the remaining neighbors among the adjacent eight candidates should be added to an expandable frozen set $\mathbf{F}_1$. Therefore, in the second iteration, the non-frozen neighbors of $\mathbf{p}_2$ are evaluated and the connecting point $\mathbf{p}_3$ could be selected. During the front propagation segmentation, the contour member $\mathbf{p}_k$ should satisfy

$$\forall \mathbf{p}_i \in \mathbf{N}_{k-1} \ \wedge \ \mathbf{p}_i \notin \mathbf{F}_{k-1}, \mathcal{M}_{\mathbf{p}_k} = \min \left\{ \mathcal{M}_{\mathbf{p}_i} \right\} \tag{22}$$

where $\mathbf{N}_{k-1}$ denotes eight neighbors of contour point $\mathbf{p}_{k-1}$ as identified in the latest iteration, and $\mathbf{F}_{k-1}$ denotes all frozen elements in previous $(k-1)$ steps. Following the above principle, the contour can be sequentially segmented based on the arrival time map $\mathcal{M}$ as illustrated in Fig. 4 (b).

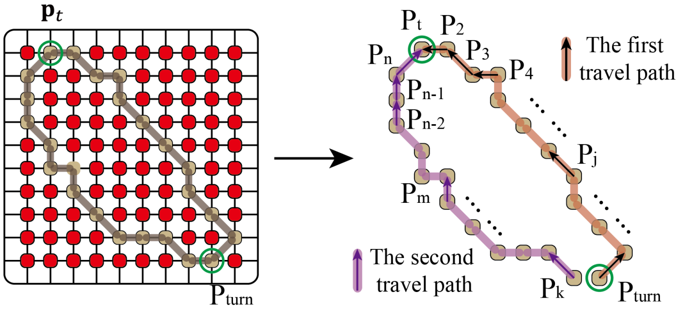


Fig. 5: Illustration of the end (turning point) evaluation.

To locate the other end of the suture thread, a specific stop criterion should be introduced in the iterative computation. By using the contour of a suture thread outlined in Fig. 5, the key points numbered as $\{\mathbf{p}_2, \mathbf{p}_3, ..., \mathbf{p}_j, ...\}$ share one high velocity contour path when traveling to the tip $\mathbf{p}_t$ as denoted by the orange color. Apart from the first path, there also exists a second path with a high traveling speed as indicated by the pink color. The overall contour of the suture thread can be treated as a semi-symmetric object. When computing the minimal increasing manipulation iteratively, there would exist a turning point $\mathbf{p}_{turn}$ after which the following point $\mathbf{p}_k$ holds the inequality $\mathcal{M}_{\mathbf{p}_k} < \mathcal{M}_{\mathbf{p}_{turn}}$. This turning phenomenon happens around the other end. Using this property, point $\mathbf{p}_{turn}$ can be treated as the end position. In the left and right cameras, these segmented points are respectively recorded in point sets $\mathbf{S}_l$ and $\mathbf{S}_r$. Given that the arrival time $\mathcal{M}$ of each selected pixel point cannot strictly obey the rule $\mathcal{M}_{\mathbf{p}_i} > \mathcal{M}_{\mathbf{p}_{i-1}}$ during the front propagation, a tolerance parameter χ was embedded. The final stop criteria can be defined as:

- For current point $\mathbf{p}_i$, if there exists χ members in Set $\{\mathbf{p}_t, \mathbf{p}_2, \mathbf{p}_3, ..., \mathbf{p}_{(i-1)}\}$ whose arrival times are larger than $\mathcal{M}_{\mathbf{p}_i}$, then the current point can be regarded as the turning point $\mathbf{p}_{turn}$ and the front propagation should be terminated.

At this stage, the shape of the suture thread can be initially outlined in the left and right camera frames. Given the asymmetric and irregular behavior of a suture thread, the evaluated turning point $\mathbf{p}_{turn}$ may not be the precise end of the suture thread. Therefore, a back propagation was performed to refine the segmented shape.

First, a RoI was created by using $\mathbf{p}_{turn}$ at the center. By combining the filtered potential image of the suture thread obtained in Section III together with the tip refined algorithm proposed in [27], the accurate end $\mathbf{p}_e$ of the suture thread can be achieved in two camera frames.

Starting from $\mathbf{p}_e$ and evaluating the local minimal value $\mathcal{M}$ of non-frozen neighbors, those elements belonging to the suture thread's contour can be identified via back propagation. The suture's tip $\mathbf{p}_t$ holds the property of $\mathcal{M}_{\mathbf{p}_t} = 0$. Consequently, the back propagation should be terminated when the current arrival time satisfies this condition. By recording the pixels selected in every iteration, a complete one-side contour of the suture thread with a sequence from the tip to end can be obtained in both cameras. The overall workflow of the 2D suture thread segmentation is summarized in Fig. 6.

The 3D suture thread can be regarded as a combination of successive sub-segments. To precisely reconstruct the suture

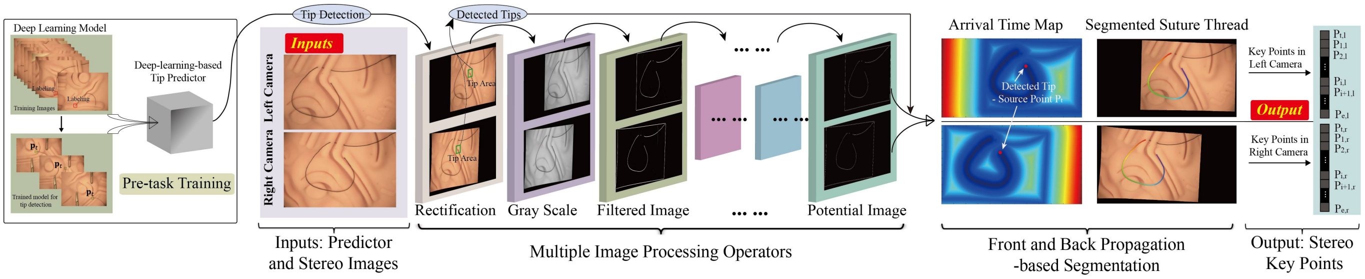

Deep Learning Model
Labeling
Training Images
Labeling
Trained model for tip detection
Deep-learning-based Tip Predictor
Pre-task Training
Tip Detection
Inputs
Left Camera
Right Camera
Detected Tips
Tip Area
Tip Area
Rectification
Gray Scale
Filtered Image
... ...
Potential Image
Arrival Time Map
Segmented Suture Thread
Detected Tip - Source Point
Key Points in Left Camera
Output
Key Points in Right Camera
Inputs: Predictor and Stereo Images
Multiple Image Processing Operators
Front and Back Propagation -based Segmentation
Output: Stereo Key Points

Fig. 6: The workflow of the suture thread segmentation in a calibrated stereo-camera system. The inputs are the stereo images. With the pre-trained tip predictor, multiple image operators and the arrival time based segmentation, the key points of the suture thread in two images can be obtained. They can be utilized to obtain the final output of its 3D coordinates.

thread's shape, sufficient key point pairs that are well aligned in the rectified stereo-camera must be extracted. For any key point $\mathbf{p}_{i,l}$ along the suture thread in the left camera, there should exist one point $\mathbf{p}_{j,r}$ in the right camera to the extent that these two points have the same image row number and vice versa. In other words, i and j should be equal. The total segmented points along the suture thread in both camera frames must also be equal to each other, and we denote this number using the parameter $K_{r,l}$.

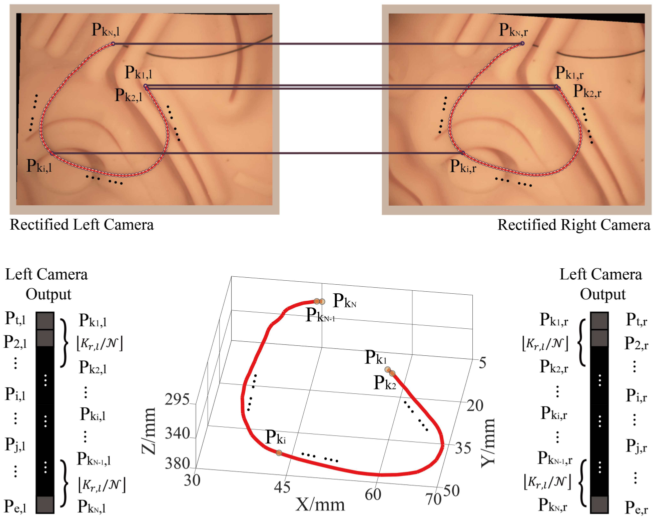


Fig. 7: Key points matching and 3D shape computation of the suture thread based on rectified image pairs in a stereo-camera.

To select $\mathcal{N}$ key points from the segmented points set, the tip point can be regarded as $\mathbf{p}_{k_1,l}$ and $\mathbf{p}_{k_1,r}$. As shown in Fig. 7, we selected one point from $\mathbf{S}_l = [\mathbf{p}_{t,l}, \mathbf{p}_{2,l}, \mathbf{p}_{3,l}, ..., \mathbf{p}_{e,l}]$ and $\mathbf{S}_r = [\mathbf{p}_{t,r}, \mathbf{p}_{2,r}, \mathbf{p}_{3,r}, ..., \mathbf{p}_{e,r}]$ in every $\lfloor K_{r,l}/\mathcal{N} \rfloor$ element. These contour points can be repositioned to the centerline of the suture body by using the refinement method presented in [27]. For the obtained 2D key point pairs $\mathbf{p}_{k_i,l} = [u_{i,l}, v_{i,l}]$ and $\mathbf{p}_{k_i,r} = [u_{i,r}, v_{i,r}]$, their 3D coordinates $\mathbf{p}_{k_i} = [X_i, Y_i, Z_i]$ can be computed as

$$\begin{cases} X_i = u_{i,l} \cdot Z_i / \mathcal{F} \\ Y_i = u_{i,l} \cdot Z_i / \mathcal{F} \\ Z_i = \mathcal{F} \cdot b / (u_{i,l} - u_{i,r}) \end{cases} \tag{23}$$

where $\mathcal{F}$ and b denote the focal and the baseline lengths of the stereo-camera, respectively. Therefore, the 3D coordinates of the suture thread with respect to the camera can be computed and illustrated point by point.

The contents presented in Section IV-C are the stereo matching approach, and its pseudo codes are listed in Algorithm 2.

Algorithm 2: A vision-based algorithm of image segmentation and stereo matching for 3D shape reconstruction of a suture thread in a stereo-camera system

```
Data: Based on stereo images and tip predictor, we obtained
      arrival time map $\mathcal{M}_l$ and $\mathcal{M}_r$, source points $\mathbf{p}_{t,l}$ and
      $\mathbf{p}_{t,r}$. Take one frame as an example and ignore the
      subscript $l$ and $r$.
1  The tolerance, the current iteration number, point, arrival
     time, frozen set, and valid contour point are set as $\chi$,
     $c = 1$, $\mathbf{p}_c = \mathbf{p}_t$, $\mathcal{M}_c = \mathcal{M}_{\mathbf{p}_t} = 0$, $\mathbf{F}_1 = Null$, and $\mathbf{S} = \mathbf{p}_t$;
2  while $I_N < \chi$ do
3      Set the inspect number as $I_N = 0$;
4      Increase iteration number as $c = c + 1$;
5      if $\forall \boldsymbol{p}_i \in \boldsymbol{N}_{c-1} and \notin \boldsymbol{F}_{c-1}$ then
6          Evaluate each patch $\mathcal{M}_{\mathbf{p}_i}$ ;
7          Find the point obtains the minimal arrival → Treat it
             as the current contour point $\mathbf{p}_c$ → Add it to $\mathbf{S}$;
8          Add $\mathbf{p}_c$ and remaining neighbors to $\mathbf{F}_c$;
9          for each $\boldsymbol{p}_j \in \boldsymbol{S}$ do
10             if $\mathcal{M}_{\boldsymbol{p}_j} \geqslant \mathcal{M}_{\boldsymbol{p}_c}$ then
11                 $I_N = I_N + 1$;
12 The last element of $\mathbf{S}$ is $\mathbf{p}_{turn}$ → Adopt the contour closure
     algorithm → The optimized $\mathbf{p}_e$ can be obtained;
13 According to Step 1, reset the corresponding parameters as
     $c = 1$, $\mathbf{p}_c = \mathbf{p}_e$, $\mathbf{F}_1 = Null$, and $\mathbf{S} = \mathbf{p}_e$;
14 while $\mathcal{M}_{\mathbf{p}_c} \neq 0$ do
15     Repeating Step 5∼8 → Iteratively update the point set
         of the segmented suture thread;
16 Set the desired key points number as $\mathcal{N}$;
17 Pick out the selected points in the left and right camera →
     Apply key points optimizer → Locate key points to the
     centerline of the suture thread;
18 Compute the 3D coordinates of these key points wrt. the
     stereo-camera;
Result: The key points coordinates and the 3D shape of the
        suture thread wrt. the camera coordinate.
```

V. Experimental Results and Discussion

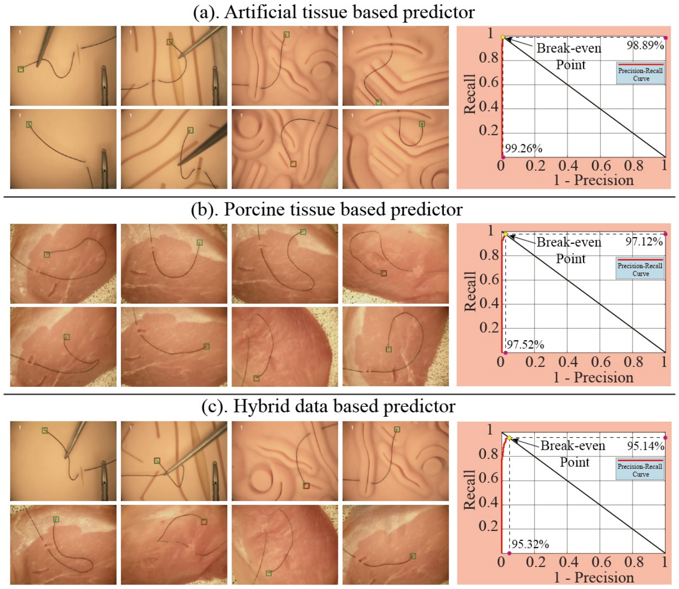
To comprehensively evaluate the feasibility and robustness of our proposed scheme with multiple operators, we performed validations from three aspects. First, we examined the feasibility of detecting the suture thread's tip using the deep learning model. Second, we verified the performance of the 2D segmentation algorithm by using different backgrounds filled with noise. Third, we validated key point matching and 3D shape computation of the suture thread by using the stereo matching algorithm.

A. Performance Validation of the Deep Learning Model

To thoroughly validate the performance of the deep learning-based method, three types of data based on different materials were adopted in our training and testing experiments. To label the ground truth of the suture thread's tip in each image, a bounding box with two pixels that represent the left top and right bottom locations was manually created. By setting the width of the squared bounding box to 32 pixels, all images in the database were labeled with their corresponding bounding boxes that were stored in ***json*** format.

In the first set, we acquired 1278 labeled images on various artificial tissues, of which 923 images were used for training and the remaining 355 images were adopted for testing. We prepared the second set of data by using porcine tissue and obtained 1215 labeled images, among which 972 were used for training and the remaining 243 were adopted for testing. In the last set, we combined the training data from the previous two sets to train a new model for tip prediction and then used the remaining combined images to test the new model.

We set the batch size to 1 and then applied root mean square propagation (RMSprop) as the optimizor with a learning rate of 0.001. We trained the model on a workstation with a Nvidia Quadro P5200 GPU. After 10000 iterations, which were almost 11 epochs, the model converged with high accuracy. Images of the testing results from the three groups are shown in Fig. 8. The processing time for tip detection by using the trained predictor is 0.28 s per testing image.


(a). Artificial tissue based predictor

(b). Porcine tissue based predictor

(c). Hybrid data based predictor

Fig. 8: Typical results and precision-recall curves of the deep learning model for the tip detection based on the background of (a). artificial tissue; (b). porcine tissue; (c). hybrid data.

TABLE II: Performance validation of the deep learning model based on different backgrounds.

Background	Precision	Recall	F1-score
Artificial Tissue	99.63%	98.89%	99.26%
Porcine Tissue	97.93%	97.12%	97.52%
Hybrid Backgrounds	95.51%	95.14%	95.32%

In the above figure, the red squares indicate the prediction candidates, while the green squares denote the final selection of the suture thread's tip location using the Hungarian algorithm. The precision-recall curves are also shown in this figure, and detailed information regarding the precision, recall, and F1-score can be found in Table II. The high-quality performance of this deep learning-based approach confirms that the model has a satisfactory detection performance compared with the other general methods used in object recognition tasks. Compared with common datasets such as MNIST and CIFAR [41] that require more than 10,000 training images for building an object detection model, the deep learning method adopted in this work can train a model for accurate tip detection by using less than 1000 training images.

B. Validation of Suture Thread's Segmentation

To test the robustness of the segmentation algorithm, we examined a suture thread in five different scenarios. We also performed three trials in each scenario, with two trials shown in Fig. 9. In the first scenario, a pure background that shows an obvious color contrast to the suture thread was adopted. By using the arrival time map and the detected tip point that was treated as the source point, the suture thread can be successfully segmented from the background, whereas its key points can be obtained in the image.

To further test the robustness, the color contrast between the suture and the background was decreased, and the environmental disturbance was increased by randomly adding noises to the background in the second and third scenarios. According to the results, the suture thread's shape was successfully segmented by the proposed scheme. In the last two groups, the experiment was performed on an artificial tissue and porcine meat. The results successfully validate the efficacy of our algorithm for the 2D shape segmentation of suture thread.

1) Precision of Exit Point Detection: Detecting the suture's exit point plays an important role in determining the completeness of its 2D segmentation, the precision of the stereo key points matching, and the overall 3D length computation. To validate its accuracy, we compared the detected exit points in the front and back propagation with the manually labeled ones. The detailed results are recorded in Table III.

The turning point was initially figured out by using the proposed stop criterion during the front propagation. The turning point shows larger errors (between 10 and 20 pixels) than the ground truth. The back propagation can then efficiently decrease these errors to around 5 pixels and significantly enhance the output of the 2D suture thread segmentation task.

2) Accuracy of Suture Thread Segmentation: After detecting the two ends and the arrival time map, the complete suture thread can be segmented. To validate the accuracy of the

TABLE III: Experiments of a suture thread's exit point detection among different backgrounds

Backgrounds	$\{\mathcal{R}_{b,\sigma_i}, \mathcal{S}_{\sigma_i}\}$	Exit Position Computation of a Pierced Suture Thread Using the Front and Back Propagation; Unit: Pixel								
		Experiment 1			**Experiment 2**			**Experiment 3**		
		Labeling	Computed Results		Labeling	Computed Results		Labeling	Computed Results	
			Initial	Rectified		Initial	Rectified		Initial	Rectified
Light Color	{1,15}	(1215,514)	(1213,531)	(1216,512)	(1252,924)	(1235,932)	(1251,923)	(525,322)	(520,320)	(521,320)
Dark Color	{2.4,10}	(525,361)	(521,374)	(526,357)	(1381,615)	(1381,602)	(1382,617)	(630,402)	(615,409)	(627,397)
With Noises	{1.5,16}	(1425,442)	(1415,454)	(1421,442)	(372,169)	(362,166)	(375,173)	(512,503)	(512,516)	(516,501)
Artificial Tissue	{2.5,14}	(193,679)	(196,699)	(191,681)	(876,913)	(868,922)	(876,914)	(595,316)	(589,310)	(598,315)
Porcine	{2,6}	(843,766)	(840,763)	(840,767)	(1020,126)	(1023,122)	(1023,124)	(1060,733)	(1043,733)	(1062,729)

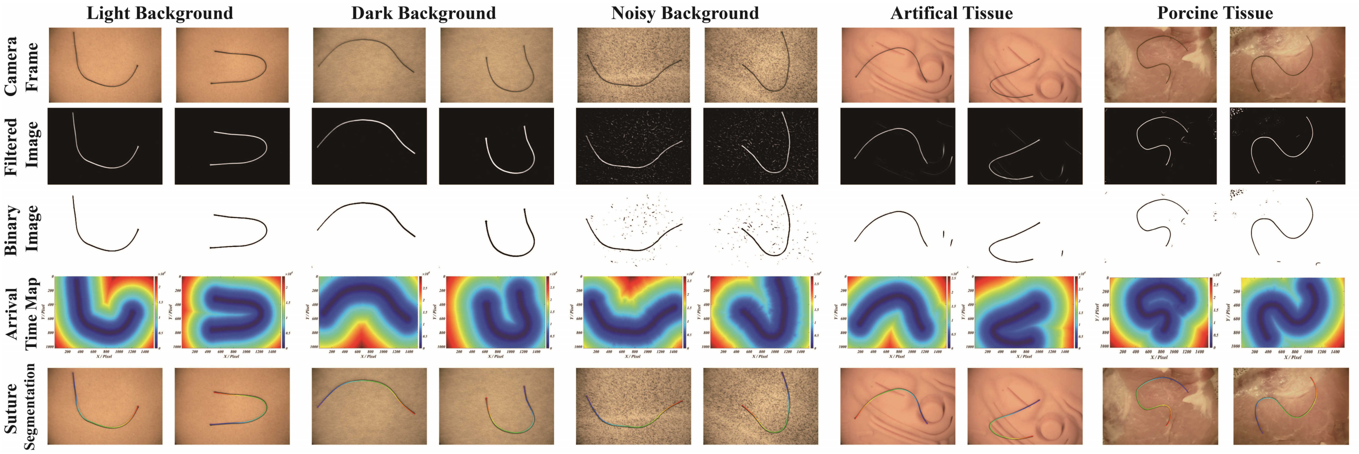


Fig. 9: Experimental results of a suture thread detection with different backgrounds. Pure light and dark backgrounds were implemented in the 1^{st} and 2^{nd} group. In the 3^{rd} trial, noises were randomly added, and the artificial tissue and porcine were adopted in the 4^{th} and 5^{th} group.

TABLE IV: Mean errors of a suture thread's segmentation based on random points validations. Unit: pixel&mm

Group Number	Artificial Tissue 1	Artificial Tissue 2	Porcine with Strong Light	Porcine with Dark Light
#1	4.35&0.22	4.35&0.22	4.81&0.24	4.74&0.24
#2	5.14&0.26	5.14&0.26	4.81&0.24	4.29&0.21
#3	5.27&0.26	5.27&0.26	5.10&0.26	4.49&0.22
#4	4.96&0.25	4.96&0.25	4.33&0.22	4.66&0.23
#5	4.84&0.24	4.84&0.24	5.00&0.25	4.24&0.21
#6	4.72&0.24	4.72&0.24	5.23&0.26	4.11&0.21
#7	4.94&0.25	4.94&0.25	5.83&0.29	3.78&0.19
#8	3.89&0.19	3.89&0.19	4.34&0.22	4.74&0.24
#9	5.03&0.25	5.03&0.25	5.66&0.28	5.03&0.25
#10	5.04&0.25	5.04&0.25	4.66&0.23	5.19&0.26

curvilinear shape segmentation, two artificial tissues and two porcine tissues under different light intensities were examined. The suture thread on the tissues was stitched arbitrarily to form various shapes. Afterward, 10 random key points along the suture were manually selected as ground truth, and their position errors with respect to the segmented suture in 2D were evaluated. For each background, 10 experiments were conducted. The mean errors concerning the units of pixels and millimeters are listed in Table IV, and the error variation in each experiment was computed by using a box plot as shown in Fig. 10. It is noticed that the largest error was around 8 pixels/0.4 mm, and the average error values were all below 6 pixels/0.3 mm. These tests prove that our method can enable an accurate suture thread segmentation with key points generated along the suture for the following 3D reconstruction.

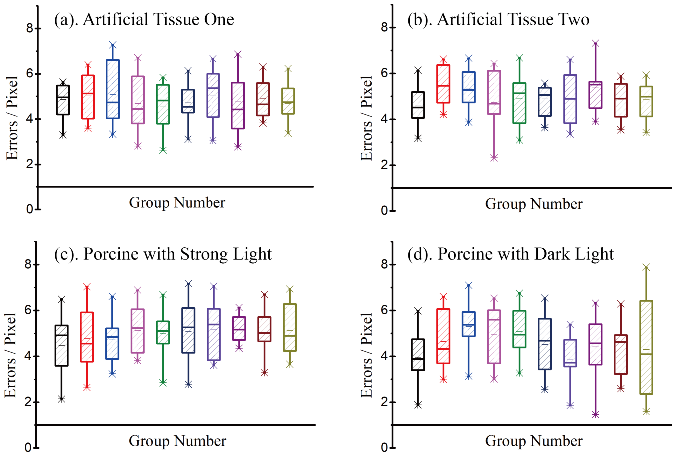


Fig. 10: Box plot of 2D segmentation errors based on random key points validations from a total of 40 experiments.

C. 3D Coordinates Computation of the Suture Thread

1) Evaluating the Consistency of Suture Thread 3D Computation: To examine the 3D computational accuracy of the suture's length, we initially measured the actual length between the tip and exit point. In the first test, we performed suture thread tracking by using the suture to form a semi-heart shape. As shown in Fig. 11, the orientation of the tissue was then changed from -45° to 45° at an increment of 22.5°. By rotating the tissue to different orientations, its corresponding shape and length in 3D can be obtained, and the robustness of

TABLE V: Computed 3D lengths and errors of a suture thread with different orientation angles on the working plane (ground truth length: 85mm).

Orientation	-45°	-22.5°	0°	22.5°	45°
Length(mm)	85.97	86.36	85.87	83.89	84.85
Error(mm)	0.97	1.36	0.87	1.11	0.15
Error ratio	1.14%	1.60%	1.02%	1.31%	0.18%

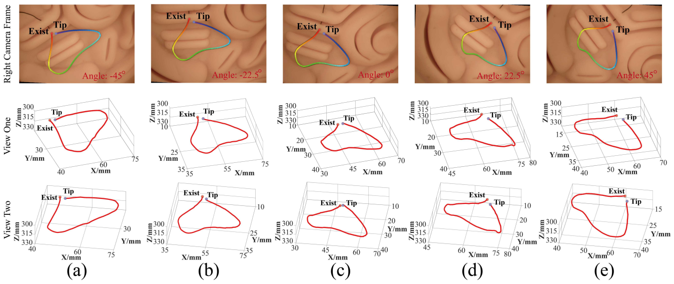


Fig. 11: Computed 3D shape of a suture thread with a rotation angle of (a). -45°; (b). -22.5°; (c). 0°; (d). 22.5°; and (e). 45°. The suture thread was formed in a semi-heart shape.

the vision-based algorithm can be evaluated. In the figure, the segmented sutures were highlighted in colored curves and the 3D shapes were shown from two view angles. The computed lengths are presented in Table V. It can be seen that the detection errors were all below 1.5 mm and the error-to-length percentages were within 2%. Considering the workspace used for the experiments, this amount of error is fairly accurate to achieve a reliable automated tool manipulation for grasping the suture thread.

2) Quantitative Evaluation of Suture Thread's 3D Length Computation: To further validate the performance of the proposed method, we utilized three artificial tissues and two porcine meats as backgrounds. The suture thread was then fixed to different lengths in each experimental set, and six experiments with variant backgrounds and suture's orientations/locations were carried out. Two experimental results from each set are shown in Fig. 12, and the remaining outcomes are presented in the Supplementary Material.

In the first set, the length was fixed at 50 mm. Fig. 12 (a) presents the segmented shape, the arrival time map, and the 3D coordinates of the suture thread with respect to the camera coordinate. To clearly observe its 3D shape, the results are also shown from two view angles. Based on the acquisition of the stereo key point pairs of the suture thread, its spatial coordinates can be obtained by using the triangulation relationship. By comparing its 3D structure to the ground truth, the overall shape appears to be correctly constructed.

In the second and third sets, the suture's lengths increased to 80 mm and 100 mm, respectively. As shown in Fig. 12 (b) and (c), the suture can be successfully detected and its 3D shape is highlighted by using the red curve. The porcine tissues were used in the 4^{th} and 5^{th} sets to validate the practical feasibility of our approach. With different light intensities, it can be noticed that the 3D information of the suture thread can be achieved regardless of its location and orientation.

Moreover, the computed lengths are presented in Table VI. Compared with the ground truth, the maximum error in the first group was 0.90 mm. Increasing the total length also increases the computed errors. With a ground truth of 100 mm in Set 3, the largest error reached 1.74 mm in Experiment 2. Compared with the total length, the error-to-length ratio was only 1.74% in this condition. In all 30 experiments, the maximum value of this ratio was only 1.8%, which happened in Set 1 Experiment 6.

Refer to the workflow in Fig. 6 and Fig. 7, the majority of the computational time was spent on generating the arrival time map. Given the different resolutions of the input camera frames, the time consumption varies from 30 to 120 seconds for each camera. All the other steps in the 3D suture thread reconstruction can be finished within 10 to 20 seconds.

Compared with [42], which focused on catheter tip detection, our work aims to provide a comprehensive solution towards figuring out not only the suture's tip but also its entire shape. Besides, our tip detection error can be limited to within 0.5 mm, which is more accurate compared with the minimal error of 3.7 mm reported in [42]. For the work in [43], they mainly studied the segmentation of the suture thread in a 2D condition. In our work, we extensively considered suture's 3D reconstruction by using a stereo camera, and the physical length of the suture thread in various conditions was evaluated. In [16], the procedure requires a manual selection of one suture's endpoint, and the accuracy in evaluating the length depends on the selected mathematical model to represent the current suture's shape. However, in our work, the computation can be initialized by using a deep-learning model, and we proposed a model-free approach. Besides, the maximal error-to-length ratio is only 1.8% as shown in Fig. 13, which is more accurate than the value of 4.75% reported in [16].

VI. Conclusion

In this paper, we presented a novel stereo vision-based algorithm to reconstruct the 3D coordinates of a suture thread. To automate the procedure, a deep-learning model was implemented to detect the suture thread's tip. An image filter with multiple image preprocessing operators was built to enhance the curvilinear structure of the suture thread while

TABLE VI: Computational results for the overall 3D length of suture threads in different scenarios. Unit: mm.

Experiment	**Ground**	Experiment 1		Experiment 2		Experiment 3		Experiment 4		Experiment 5		Experiment 6		Error Analysis	
Set	**Truth**	Result	Error	Result	Error	Result	Error	Result	Error	Result	Error	Result	Error	Average	Maximal
Set 1	50	49.95	0.05	49.93	0.07	49.79	0.21	50.64	0.64	49.83	0.17	50.90	0.90	0.34	0.90
Set 2	80	79.90	0.10	79.12	0.88	79.89	0.11	79.99	0.01	80.22	0.22	80.47	0.47	0.30	0.88
Set 3	100	100.67	0.67	101.74	1.74	99.89	0.11	101.01	1.01	100.68	0.68	99.24	0.76	0.83	1.74
Set 4	100	99.02	0.98	99.46	0.54	100.29	0.29	98.86	1.14	100.95	0.95	99.92	0.08	0.66	1.14
Set 5	100	99.03	0.97	98.49	1.51	99.39	0.61	98.80	1.20	98.46	1.54	99.71	0.29	1.02	1.54

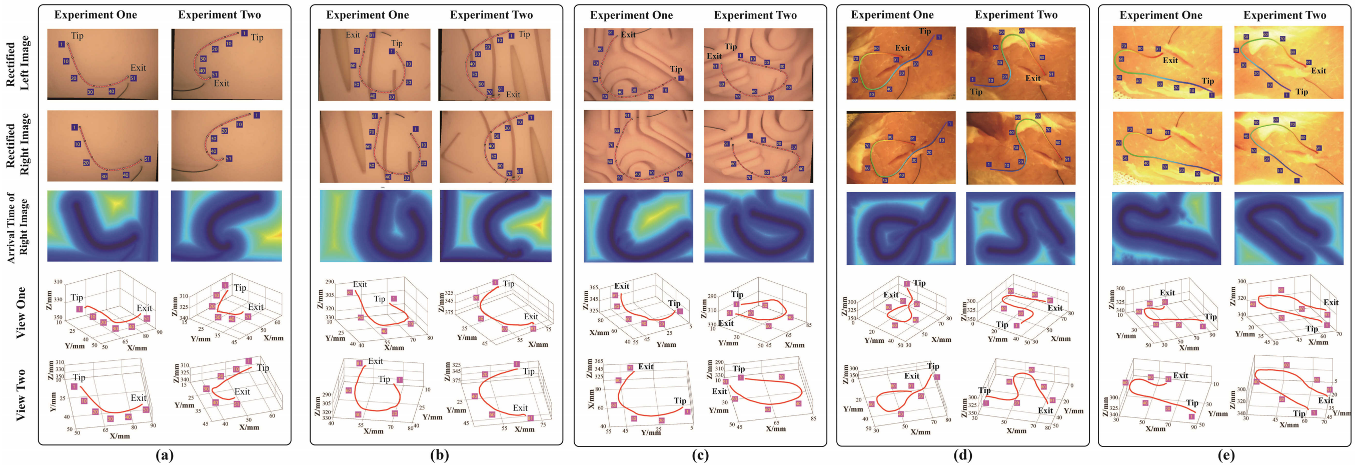


Fig. 12: Typical results of the overall 3D shape computation of the suture thread. Key points correspondences were figured out along the suture thread in the stereo-camera system. The suture length and the background were set as (a). 50mm - artificial tissue 1; (b). 80mm - artificial tissue 2; (c). 100mm - artificial tissue 3; (d). 100mm - porcine tissue with dark light; (e). 100mm - porcine tissue with strong light.

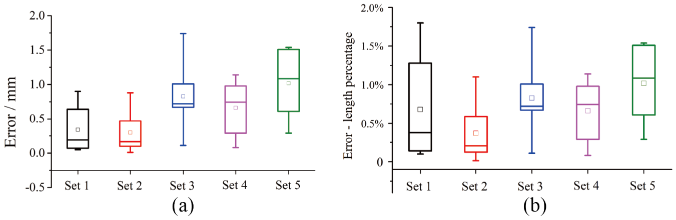


Fig. 13: Box plot of errors for the computed 3D length of a suture thread in various conditions. (a). Absolute errors; (b). Corresponding error-to-length ratios in each experimental set.

eliminating environmental noise. To precisely and completely describe the suture thread, a novel concept based on the arrival time of the propagation which was calculated based on the multistencils fast marching principle was adopted. Based on this knowledge, the suture thread can be segmented by using the front propagation algorithm and efficiently refined through the back propagation approach. Contours of the suture with zigzag or redundant paths can be eliminated, thereby ensuring the precision of key points matching and 3D shape computation in the stereo-camera. The accuracy of the deep learning model for suture's tip detection was examined by using different backgrounds. The performance of the proposed approach in 2D image segmentation, 3D shape reconstruction, and spatial length computation of the suture thread was then comprehensively and successfully validated across various experimental scenarios.

Based on these findings, our approach can be successfully implemented to compute the 3D length and coordinates of a surgical suture thread, thereby reducing the human intervention and facilitate an automated manipulation of the vision-based suture thread grasping task by using a robot, which gives a promising future of achieving a higher level of automation in the task of robot-assisted surgical knot tying.

References

[1] G. I. Barbash and S. A. Glied, "New Technology and Health Care Costs - The Case of Robot-Assisted Surgery," The New England Journal of Medicine, Vol. 363, pp. 701-704, 2010.

[2] K. Moorthy, et al., "Dexterity enhancement with robotic surgery, " Surgical Endoscopy And Other Interventional Techniques, Vol. 18, pp. 790-795, 2004.

[3] R. Konietschke, T. Ortmaier, H. Weiss, G. Hirzinger and R. Engelke, "Manipulability and Accuracy Measures for a Medical Robot in Minimally Invasive Surgery," On Advances in Robot Kinematics, pp. 191-198, 2004.

[4] J. Marescaux, et al., "Transatlantic robot-assisted telesurgery," Nature, Vol. 413, pp.379-380, 2001.

[5] C. Freschi, V. Ferrari, F. Melfi, M. Ferrari, F. Mosca and A. Cuschieri, "Technical review of the da Vinci surgical telemanipulator," The International Journal of Medical Robotics and Computer Assisted Surgery, Vol. 9(4), pp. 396-406, 2012.

[6] R. Berguer, "Surgery and Ergonomics," The Archives of Surgery, Vol. 134(9), pp.1011-1016, 1999.

[7] G. P. Moustris, S. C. Hiridis, K. M. Deliparaschos and K. M. Konstantinidis, "Evolution of autonomous and semi-autonomous robotic surgical systems: a review of the literature," Vol. 7(4), pp. 375-392, 2011.

[8] B. Lu, H. K. Chu and L. Cheng, "Robotic knot tying through a spatial trajectory with a visual servoing system," IEEE/RSJ International Conference on Intelligent Robots and Systems, 2017.

[9] B. Lu, H. K. Chu, K. C. Huang and L. Cheng, "Vision-Based Surgical Suture Looping Through Trajectory Planning for Wound Suturing, " IEEE Transactions on Automation Science and Engineering, Vol. 16(2), pp. 542-556, 2018.

[10] C. R. Wagner, et al., "The Benefit of Force Feedback in Surgery: Examination of Blunt Dissection," Presence: Teleoperators and Virtual Environments, Vol. 16(3), pp. 252-262, 2007.

[11] S. Sen, A. Garg, D. V. Gealy, S. McKinley, Y. Jen, K. Goldberg, "Automating Multi-Throw Multilateral Surgical Suturing with a Mechanical Needle Guide and Sequential Convex Optimization," IEEE International Conference on Robotics and Automation (ICRA), 2016.

[12] S. A. Pedram, P. Ferguson, J. Ma, E. Dutson, and J. Rosen, "Autonomous Suturing Via Surgical Robot: An Algorithm for Optimal Selection of Needle Diameter, Shape, and Path," IEEE International Conference on Robotics and Automation (ICRA), 2017.

[13] T. Osa, N. Sugita, and M. Mitsuishi, "Online Trajectory Planning and Force Control for Automation of Surgical Tasks" IEEE Transactions on Automation Science and Engineering, Vol. 15, No. 2, pp. 675-691, 2018.

[14] G. Mohanarajah, V. Usenko, M. Singh, R. D'Andrea, and M. Waibel, "Cloud-Based Collaborative 3D Mapping in Real-Time With Low-Cost Robots," IEEE Transactions on Automation Science and Engineering, Vol. 12, No. 2, pp. 423-431, 2015.

[15] H. Ren, W. Liu, and A. Lim, "Marker-Based Surgical Instrument Tracking Using Dual Kinect Sensors," IEEE Transactions on Automation Science and Engineering, Vol. 11, No. 3, pp. 921-924, 2014.

[16] R. C. Jackson, R. Yuan, D. Chow, W. S. Newman, M. C. Cavusoglu, "Real-Time Visual Tracking of Dynamic Surgical Suture Threads, "

IEEE Transactions on Automation Science and Engineering, Vol. 15(3), pp. 1078-1091, 2018.

[17] B. Obara, M. Fricker, D, Gavaghan, and V. Grau, “Contrast-Independent Curvilinear Structure Detection in Biomedical Images, ” IEEE Transactions on Image Processing, Vol. 21, No. 5, pp. 2572-2581, 2012.

[18] F. Benmansour and L. D. Cohen, “Fast Object Segmentation by Growing Minimal Paths from a Single Point on 2D or 3D Images,” Journal of Mathematical Imaging and Vision, Vol. 33, pp. 209-221, 2009.

[19] V. Kaul, A. Yezzi, and Y. Tsai, “Detecting Curves with Unknown Endpoints and Arbitrary Topology Using Minimal Paths, ” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 34, No. 10, pp. 1952-1965, 2011.

[20] Y. J. Cha and W. Choi, “Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks, ” Computer-Aided Civil and Infrastructure Engineering, Vol. 32, pp. 361-378, 2017.

[21] Y. J. Cha, W. Choi, G. Suh, and S. Mahmoudkhani, “Autonomous Structural Visual Inspection Using Region-Based Deep Learning for Detecting Multiple Damage Types, ” Computer-Aided Civil and Infrastructure Engineering, Vol. 33, pp. 731-747, 2018.

[22] F. C. Chen and M. R. Jahanshahi, “NB-CNN: Deep Learning-Based Crack Detection Using Convolutional Neural Network and Naive Bayes Data Fusion, ” IEEE Transactions on Industrial Electronics, Vol. 65(5), 2018.

[23] R. Stewart, M. Andriluka, and A. Y. Ng, “End-to-end people detection in crowded scenes,” The IEEE Conference on Computer Vision and Pattern Recognition, pp. 2325-2333, 2016.

[24] X. Qian, M. Brennan, D. Dione, W. Dobrucki, M. Jackowski, C. Breuer, A. Sinusas, and X. Papademetris, “A non-parametric vessel detection method for complex vascular structures,” Medical Image Analysis, Vol. 13(1), pp. 49-61, 2009.

[25] M. S. Hassouna and A. A. Farag, “Multistencils Fast Marching Methods: A Highly Accurate Solution to the Eikonal Equation on Cartesian Domains,” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 29(9), pp. 1563-1574, 2007.

[26] D. Chen, J. M. Mirebeau, and L. D. Cohen, “Vessel tree extraction using radius-lifted keypoints searching scheme and anisotropic fast marching method, ” Journal of Algorithms and Computational Technology, Vol. 10(4), pp. 224-234, 2016.

[27] B. Lu, H. K. Chu, K. C. Huang, and J. W. Lai, “Surgical Suture Thread Detection using a Model-free Approach in a Calibrated Stereo-visual System,” IEEE/ASME Transactions on Mechatronics, doi: 10.1109/TMECH.2019.2942715.

[28] C. Goutte, F. A. Nielsen and K. H. Hansen, “Modeling the hemodynamic response in fMRI using smooth FIR filters,” IEEE Transactions on Medical Imaging, Vol. 19(12), 2000.

[29] G. Deng and L. W. Cahill, “An adaptive Gaussian filter for noise reduction and edge detection,” IEEE Conference Record Nuclear Science Symposium and Medical Imaging Conference, 1993.

[30] P. W. Verbeek, H. A. Vrooman and L. J. Van Vliet, “Low-level image processing by max-min filters,” Signal Processing, Vol. 15, No. 3, pp. 249-258, 1998.

[31] S. J. Ko and Y. H. Lee, “Center weighted median filters and their applications to image enhancement,” IEEE Transactions on Circuits and Systems, Vol. 38(9), pp. 984-993, 1991.

[32] C. Steger, “An Unbiased Detector of Curvilinear Structures,” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, No. 2, pp. 113-125, 1998.

[33] A. F. Frangi, W. J. Niessen, K. L. Vincken and M. A. Viergever, “Multiscale vessel enhancement filtering,” Medical Image Computing and Computer-Assisted Intervention, Vol. 1496, 1998.

[34] T. Lindeberg, “Edge Detection and Ridge Detection with Automatic Scale Selection,” International Journal of Computer Vision, Vol. 30(2), pp.117-156, 1998.

[35] C. Lorenz, I. C. Carlsen, T. M. Buzug, C. Fassnacht and J. Weese, “Multi-scale Line Segmentation with Automatic Estimation of Width, Contrast and Tangential Direction in 2D and 3D Medical Images,” CVRMed-MRCAS’97, LNCS, pp 233-242, 1997.

[36] J. Jin, L. Yang, X. Zhang, and M. Ding, “Vascular Tree Segmentation in Medical Images Using Hessian-Based Multiscale Filtering and Level Set Method,” Computational and Mathematical Methods in Medicine, Vol. 2013, 2013.

[37] D. Adalsteinsson and J. Sethian, “A Fast Level Set Method for Propagating Interfaces,” Journal of Computational Physics, Vol. 118(2), pp. 269-277, 1995.

[38] E. Rouy and A. Tourin, “A Viscosity Solutions Approach to Shape-From-Shading,” SIAM Journal on Numerical Analysis, Vol. 29(3), pp. 867-884, 1992.

[39] J. Sethian, “Level Sets Methods and Fast Marching Methods,” Cambridge University Press, 1999.

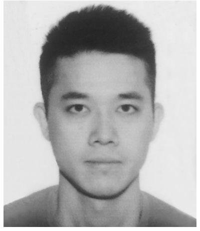
[40] A. Fusiello, E. Trucco, and A. Verri, “A compact algorithm for rectification of stereo pairs,” Machine Vision and Applications, Vol. 12, pp. 16-22, 2000.

[41] W. Rawat and Z. Wang, “Deep convolutional neural networks for image classification: A comprehensive review,” Neural computation, Vol. 29(9), pp.2352-2449, 2017.

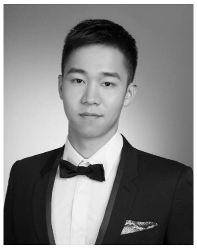
[42] H. Lee, M. Mansouri, S. Tajmir, M.H. Lev, and S. Do, “A deep-learning system for fully-automated peripherally inserted central catheter (PICC) tip detection,” Journal of digital imaging, Vol. 31(4), pp.393-402, 2018.

[43] Y. Hu, Y. Gu, J. Yang, G.Z. Yang, “Multi-stage Suture Detection for Robot Assisted Anastomosis based on Deep Learning” IEEE International Conference on Robotics and Automation (ICRA), 2018.

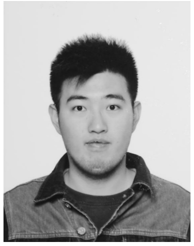
Bo Lu received the B.Eng. degree from the Department of Ship and Offshore Engineering, Dalian University of Technology, Liaoning Province, China, in 2013. He obtained his M.S. Degree (first honor) and Ph.D. Degree from the Department of Mechanical Engineering, The Hong Kong Polytechnic University, Hong Kong, in 2015 and 2019, respectively. He is now a Postdoctoral Research Fellow in the T-stone Robotics Institute, The Chinese University of Hong Kong, N. T., Hong Kong. His current research interests include medical robotics, computer vision, vision-based manipulation, automation and control, trajectory plan, surgical tool detection, surgical procedure analysis and understanding.



X.B. Yu received his B.Eng. degree from Zhejiang Chinese Medical University, Zhejiang, China in 2012 and his M.S. degree from The Hong Kong Polytechnic University, Hong Kong, in 2015. He is a Ph.D. candidate in the department of Computing of The Hong Kong Polytechnic University. His research interest includes big data analytics, machine learning, data mining, artificial intelligence, deep learning, and software engineering.



J.W. Lai received his B.Eng. degree from the Department of Metallurgical Engineering, Wuhan University of Science and Technology, Wuhan, China, in 2016, and his M.Sc. degree from the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong, in 2017. He is now pursuing his Ph.D. degree in The Hong Kong Polytechnic University, Hong Kong. His research interests include continuum robot, soft robot, robotic control, computer vision, and surgical robot system.



K.C. Huang received his B.Eng. degree in the Automation, Department of Mechatronic and Control Engineering from Shenzhen University, Shenzhen, Guangdong, China, in 2014, and the M.Sc. degree in Mechanical and Automation Engineering from The Chinese University of Hong Kong, Hong Kong, China, in 2015. He is currently a Ph.D. candidate in the Department of Mechanical Engineering from the Hong Kong Polytechnic University. His research interests include automated cell patterning with dielectrophoresis.

Keith C.C. Chan received the B.Math. (Hons.) degree in computer science and statistics in 1984 and the M.A.Sc. and Ph.D. degrees in systems design engineering in 1985 and 1989, respectively, all from the University of Waterloo, Waterloo, ON, Canada. He then worked as a Software Analyst for the development of multimedia and software engineering tools with the IBM Canada Laboratory, Toronto, ON. In 1994, he joined the Hong Kong Polytechnic University, Hung Hom, Hong Kong, where he is currently a Professor with the Department of Computing. He has authored or coauthored more than 250 publications. His research has been supported both by government research funding agencies and the industry. His research interest includes big data analytics, computational biology, data mining, machine learning, evolutionary computation, and artificial intelligence.

Henry K. Chu (M'12) received the B.S. degree in mechanical engineering from the University of Waterloo, Canada, and the M.S. and Ph.D. degrees in mechanical and industrial engineering from the University of Toronto, Canada. He was a Post-Doctoral Fellow with the University of Toronto and the City University of Hong Kong, Hong Kong. He is currently an Assistant Professor with The Hong Kong Polytechnic University, Hong Kong. His research interests include robotic manipulation, vision-based control and automation, microsystem design, and tissue engineering.