2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
September 27 - October 1, 2021. Prague, Czech Republic

2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) | 978-1-6654-1714-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/IROS51168.2021.9636441

Localization and Control of Magnetic Suture Needles in Cluttered Surgical Site with Blood and Tissue

Will Pryor†, Yotam Barnoy†, Suraj Raval, Xiaolong Liu*, Lamar Mair, Daniel Lerner, Onder Erin, Gregory D. Hager, Yancy Diaz-Mercado, Axel Krieger

***Abstract*— Real-time visual localization of needles is necessary for various surgical applications, including surgical automation and visual feedback. In this study we investigate localization and autonomous robotic control of needles in the context of our magneto-suturing system. Our system holds the potential for surgical manipulation with the benefit of minimal invasiveness and reduced patient side effects. However, the nonlinear magnetic fields produce unintuitive forces and demand delicate position-based control that exceeds the capabilities of direct human manipulation. This makes automatic needle localization a necessity. Our localization method combines neural network-based segmentation and classical techniques, and we are able to consistently locate our needle with 0.73 mm RMS error in clean environments and 2.72 mm RMS error in challenging environments with blood and occlusion. The average localization RMS error is 2.16 mm for all environments we used in the experiments. We combine this localization method with our closed-loop feedback control system to demonstrate the further applicability of localization to autonomous control. Our needle is able to follow a running suture path in (1) no blood, no tissue; (2) heavy blood, no tissue; (3) no blood, with tissue; and (4) heavy blood, with tissue environments. The tip position tracking error ranges from 2.6 mm to 3.7 mm RMS, opening the door towards autonomous suturing tasks.**

***Index Terms*— Computer Vision for Medical Robotics; Needle Localization, Image Segmentation; Magnetic Manipulation; Autonomous Control; Surgical Robotics; Suture Needle**

I. Introduction

A surgical environment contains many unstructured elements that make localization of surgical tools difficult. In such conditions, the localization task is the surgical equivalent of “finding a needle in a haystack”, and accurate localization is necessary for manual, tele-operated robotic, and autonomous surgical tasks. This broad localization concept is useful for aiding the visualization of small surgical tools during both manual and robotic surgery, as well as for surgical operations with wireless magnetic robot systems.

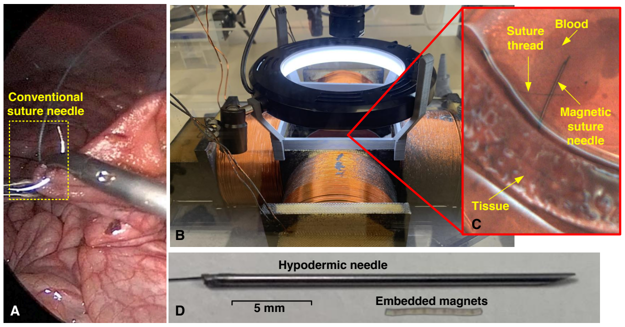


Fig. 1: Illustration of suture tasks in a minimally invasive surgery and our proof-of-concept magnetic suture system. (A) Anastomosis of the small intestine by using a conventional suture needle. (B) The physical magnetic suture system used in this study. (C) Magnetic suture needle in a simulated cluttered surgical site with blood and tissue. (D) The magnetic suture needle used in this study.

Magnetic robotic systems use magnetic fields to rapidly convey force and torque to rigid magnetic bodies. These magnetic bodies effectively become wireless end effectors of the magnetic systems. Since these magnetic end effectors can exert torques and forces, their usage towards multi-degree-of-freedom, minimally invasive surgical operations has been explored with many applications, including biopsy, targeted drug delivery, and capsule endoscopy [1]–[3]. Additionally, magnetic manipulation removes the need for on-board power supplies, thus reducing surgical tool size. Implementing magnetic suturing operations may reduce invasiveness, increase patient comfort, reduce hospital stays, and hasten recovery.

The nonlinear nature of magnetic actuation systems makes direct human manipulation of magnetic robots challenging. Even worse, excessive use of magnetic force or torque could cause tissue damage or complications. To prevent undesired outcomes and assist surgeons with manipulating magnetic robots, reliable localization algorithms combined with closed-loop feedback systems could be integrated, moving the process closer to autonomous suturing tasks by using visual servoing via camera imaging [4].

For a reliable autonomous suturing task, two sub-algorithms are crucial: (1) an accurate localization/sensing of the state of the suturing needle, and (2) a control algorithm for calculating the desired forces and torques on the suturing needle. Inaccurately detecting the state of the suturing needle may deceive the control algorithm and result in undesired or unsafe forces or torques being applied. Accurate localization

This paper was supported by the University of Maryland Medical Device Development Fund (MDDF) and the National Institute of Biomedical Imaging and Bioengineering (NIBIB) of the National Institutes of Health under award number R01EB020610.

† Will Pryor and Yotam Barnoy contributed equally to this paper.

* Corresponding author: Xiaolong Liu, xiaolong@jhu.edu

Will Pryor, Yotam Barnoy and Gregory D. Hager are with the Department of Computer Science, Johns Hopkins University, Baltimore, MD 21211 USA.

Suraj Raval, Daniel Lerner and Yancy Diaz-Mercado are with the Department of Mechanical Engineering, University of Maryland, College Park, MD 20742 USA.

Xiaolong Liu, Onder Erin and Axel Krieger are with the Department of Mechanical Engineering, Johns Hopkins University, Baltimore, MD 21211 USA.

Lamar O. Mair is with Weinberg Medical Physics, Inc., North Bethesda, MD 20852 USA.

and control with high closed-loop feedback rates in cluttered environments remains a challenge. Many localization practices are limited to occlusion-free environments, which oversimplify clinical scenarios [5].

Our localization goal is to track the three components of the needle's state (x, y, θ) rapidly in visually complex and cluttered environments for real-time closed-loop control applications. One known localization method makes use of feature-detection approaches to extract specific features of the needle, such as the boundaries between different-colored sections [6]. Such feature-detection is more prevalent in the broader field of surgical tool tracking [7], as more complex surgical tools have more easily identifiable natural features [8]. Our needle has few natural features, and direct deep learning methods to predict needle position typically produce a bounding box or region containing the tool, rather than our desired precise tool pose [9], [10]. In segmentation studies, color information is often used to separate the needle from the background [11], [12]. However, during surgical procedures, blood may discolor the needle, and specular reflection from tissue may mimic metallic reflection from the needle. In these cases, color information is insufficient to distinguish needle pixels from background pixels. Other work uses machine learning methods, such as random forests [13] or neural networks [14], to learn segmentation models that incorporate information from neighboring pixels. However, these works result in segmentation masks with considerable noise and generally rely on the distinct shape of a curved needle. Numerous features in the environment incidentally resemble our straight needle and can therefore cause false detections. The conventional design for segmentation using neural networks relies on an encoder-decoder architecture, where the image is first forced into an embedding space using convolution layers, and then extracted into a segmentation mask. This is the framework for several popular architectures such as Deeplab [15] and U-Net [16].

Applying neural networks to real-time control requires low latencies, and thus minimal end-to-end neural network forward propagation times. One way to speed up computation is to replace the usage of full convolutions, which are relatively expensive to perform, in the encoder. Dilated convolutions have been attempted [17]. Recently, depthwise-separable convolutions with separate depth and space convolution stages have been used by [18] and LWANet [19]. These operations greatly decrease the number of required operations, driving latencies lower. LWANet utilizes MobileNetV2 [20], which also makes use of depthwise-separable convolutions, as its encoder, in addition to making use of other efficient elements such as attention fusion blocks.

In this paper, we present a novel needle localization system feeding low-latency neural network-based segmentation into a classical localization method utilizing RANSAC [21], thus improving on our previously presented localization [22]. Our neural network emphasizes speed, accuracy, and a detailed output which combines a high-resolution segmentation mask and additional information in the form of a classification output. The proposed system maintains accurate localization in the presence of blood and tissue occlusion, while maintaining real-time performance. We present experiments demonstrating the improvement on prerecorded videos and in a needle control task. The presented localization algorithm, combined with a real-time closed-loop feedback control scheme, brings us closer to true magnetic suturing operations.

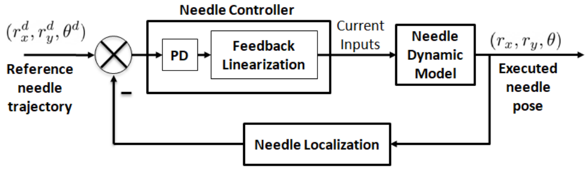


Fig. 2: Magnetic suture needle control scheme.

II. Magnetic Suture System Setup and Needle Control Scheme

Our physical magnetic suture system shown in Fig. 1B was previously presented in [5], [22]. Here, we implement an updated needle design that relies on a hypodermic needle filled with NdFeB micromagnets. Specifically, we consider the problem of controlling the motion of a 22 gauge hypodermic needle (ID = $0.413\,\mathrm{mm}$, OD = $0.7176\,\mathrm{mm}$, length = $23.5\,\mathrm{mm}$, EXEL International 22Bx1) with internally embedded NdFeB permanent magnets ($0.3\,\mathrm{mm}$ diameter, $0.5\,\mathrm{mm}$ length, 42 inserted magnets, axially magnetized). The NdFeB magnets were sealed in place with glue. The needle is submerged in a viscous medium with simulated blood and tissue, inside a Petri dish (diameter = $85\,\mathrm{mm}$). Imaging was performed at 60 frames per second (FPS) using a FLIR Blackfly camera (BFS-U3-13Y3C-C) with a resolution of 1280 × 1024 pixels. During activation, maximum currents of $10\,\mathrm{A}$ were measured by current transducers (CR5411S-30 AC/DC Hall Effect Current Transducer, CR Magnetics, Inc.). All the experiments were run on a GeForce RTX 2060, with an average localization and control loop time of around 50 milliseconds. The workspace was illuminated by a ring light mounted on a custom 3D-printed adapter.

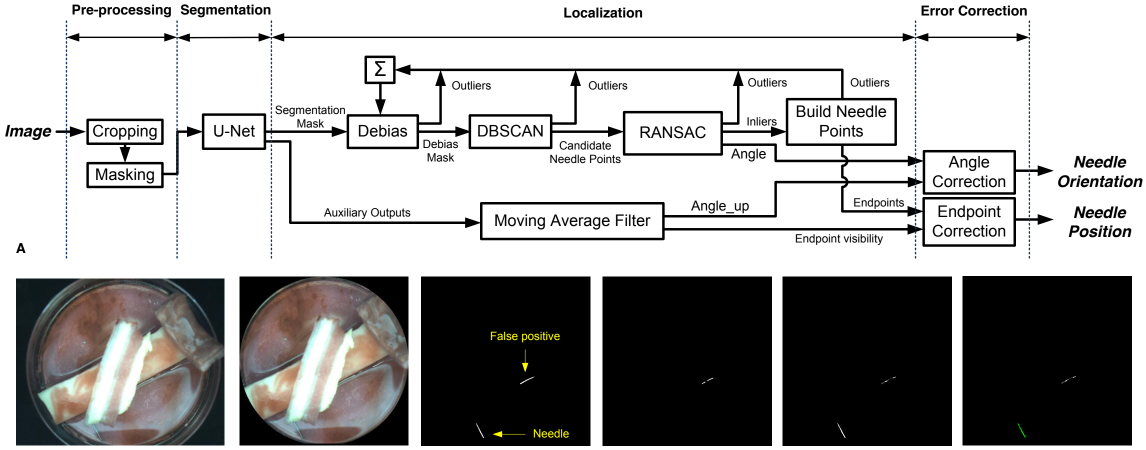
We designed a closed-loop needle control scheme as shown in Fig. 2, which consists of a real-time needle localization method for dealing with cluttered surgical environments (Sec. III), a needle controller, and a nonholonomic model of the needle for computing the desired current inputs of the electromagnets and actuating the needle to follow a reference trajectory (Sec. IV).

III. Needle Localization

Our needle localization aims to accurately detect the position and orientation of the magnetic suture needle in cluttered simulated surgical environments in real time. The framework (Fig. 3A) consists of a pre-processing step, a neural network based segmentation step, a localization step, and an error correction step.

A. Needle Segmentation

The speed and accuracy of the needle segmentation are equally crucial for the success of magnetic suture needle control. Although LWANet [19] targets fast test performance



B: Original input image C: Pre-processed image D: Segmented image E: Bias F: Debias G: Clustered

Fig. 3: Framework of magnetic suture needle localization.

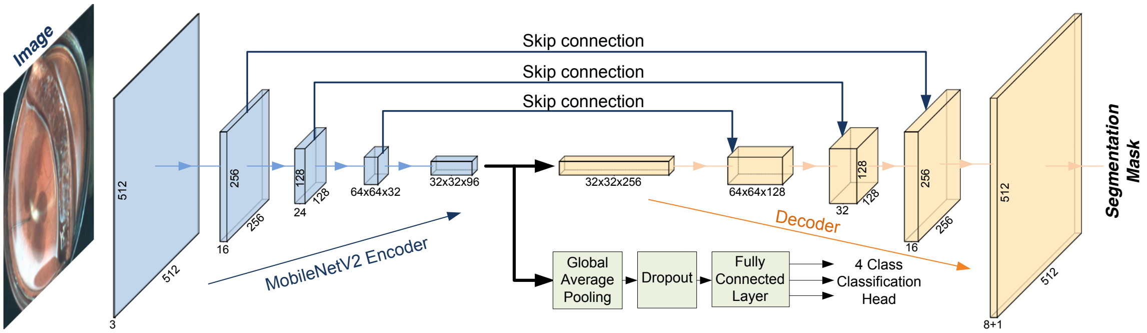


Fig. 4: U-Net architecture for training the segmentation mask and needle tip classification.

with its MobileNetV2 [20] encoder, we found that LWANet is highly sensitive to hyperparameters, slow to train, and reduces segmentation resolution too substantially for use. Since the MobileNetV2 architecture [20] by itself utilizes depthwise-separable convolutions which require far fewer GPU operations, we designed a U-Net [16] architecture as shown in Fig. 4 to achieve faster training speed, better training stability, low network latency and sufficient segmentation accuracy at high resolution (512×512 pixels).

Our network consists of five MobileNetV2 [20] *encoder* blocks, each of which halves the width and height of the image, but increases the number of features to levels prescribed by MobileNetV2 (Fig. 4). Following the encoder blocks is an equal number of *decoder* blocks, each of which doubles the width and height while decreasing the number of features. Each decoder block consists of 2 repetitions of {convolution, batch normalization, rectified linear unit (ReLU)}. A skip connection connects each encoder block to the corresponding decoder block, allowing the decoder to consider information from multiple resolution levels. The final layer adds a convolution reducing the number of features to one. After applying a sigmoid function, this yields our segmentation mask for either ‘needle’ or ‘background’ (Fig. 3D).

In order to perform suturing tasks, we need the full needle orientation. To robustly localize the needle during partial occlusion we need to know if the needle is obscured on one or both of its ends. Therefore, we require both the segmentation mask and some data from the neural network. We could train a separate neural network for this classification task, but that would entail running two neural networks on each frame, increasing our latency. Additionally, it is likely that information useful for segmentation will also inform classification. We therefore connect the output of the encoder to an additional classification ‘head’. We use global average pooling to interpret each feature by itself, then add a fully connected layer with a sigmoid per output. We train the network to produce additional information about the needle: *angle_up* and *angle_left*, which assist in localizing the angle of the needle, and *tip_visible* and *tail_visible*, which assist in localizing the position of the needle during partial occlusion. These values are described further in Sec. III-D.

We train the network’s segmentation head using dice loss [23], emphasizing small classes (such as our needle) relative to the whole image and binary cross-entropy loss for the classification head. We combine the two (equal weights) into one loss, using a learning rate of 0.0001 and a Root Mean Square Propagation optimizer running for 20 epochs.

B. Pre-processing

Images are collected at 60 FPS and for each sampled frame (Fig. 3B), the image is cropped to 1024×1024 pixels (Fig. 3C), which is the approximate size of the Petri dish in our sample dataset (see Sec. V-A). The image is then masked by a circle with the actual radius of the dish. Both the imaging cropping and masking are centered at the dish center, which is manually calibrated together with the radius. Images

Algorithm 1 Construction of final set of needle points

```
Sort clusters in decreasing order by number of points
$needle\_points := largest\_cluster$
for $cluster$ sets do
    if $cluster \neq largest\_cluster$ then
        $d := max(||x_c - x_n||),\ \forall x_c \in cluster,\ \forall x_n \in needle\_points$
        if $d < 1.1 \times needle\_length$ then
            Append $cluster$ to $needle\_points$
return $needle\_points$
```

are then downsampled to 512×512 pixels, and normalized per channel of RGB to a mean of $(0.485, 0.456, 0.406)$ and a standard deviation of $(0.229, 0.224, 0.225)$, for compatibility with pretrained weights distributed by PyTorch [24]. These weights are the result of training a MobileNetV2 network [20] on the ImageNet dataset.

C. Localization

The needle localization method begins by running the preprocessed image through the segmentation network. We then apply a *debiasing* step to correct detectable false negatives. The segmentation's most common failure mode is to falsely identify a linear section of an obstacle boundary as a needle. Such false positives are consistent in location throughout a single trial of the needle control and are typically shorter than a full needle. When the needle is fully visible the localization method successfully identifies it even when a false needle is also present because the longest needle-like object is identified. However, when the needle is partially occluded (as shown in Fig. 3B), the false needle appears longer than the unoccluded portion of the real needle and the false needle is incorrectly identified, as shown in Fig. 3D. To address this, we maintain a *bias* image, which identifies pixels that are consistently classified as needle pixels by the segmentation but are filtered out as false positives by the post-processing steps. For each sample the false positive image is generated by subtracting the predicted needle pixels, as identified by the localization framework, from the segmented image (Fig. 3D). The *bias*, shown in Fig. 3E, is computed as the pixel-wise exponential moving average of the false positive image. Pixels where the bias term is greater than 0.1 are subtracted from the segmentation mask in the *debiasing* step.

The points in the debiased segmentation mask (Fig. 3F) are clustered using the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [25] clustering algorithm. Clusters with too few points are discarded as outliers, as are points identified as outliers by DBSCAN. We compute the threshold N_{min} for discarding a cluster as the square of the needle width. We observe that the segmentation mask reliably produces clusters whose width is approximately equal to the needle width, and this threshold ensures the line detection will never receive a cluster whose longer direction is along the needle width. If no clusters remain, localization terminates and returns *No Detection*.

The set of points belonging to non-discarded clusters is passed to the random sample consensus (RANSAC) method [21] for line detection. Points identified as outliers by RANSAC are removed from their respective clusters. If fewer than N_{min} points are identified as inliers, the localization terminates and returns *No Detection*. Otherwise the final set of needle points is constructed according to Algorithm 1. The result is illustrated in Fig. 3G. Finally, the two extreme points in the set of needle points along the angle returned by RANSAC are taken to be the needle endpoints.

D. Error Correction

The localization procedure identifies the needle angle and visible endpoints, but two ambiguities remain. First, RANSAC cannot tell the difference between the needle angle $0 \leq \theta < \pi$ and the corresponding angle $\theta' = \theta + \pi, \pi \leq \theta' < 2\pi$. Accurate needle control requires knowing the correct orientation of the needle. Second, the identified endpoints may be the true endpoints of the needle or they may be the points at which the needle disappears under an obstacle. Proper occlusion handling requires resolving this ambiguity. We resolve both ambiguities using the outputs from the segmentation network described in III-A.

angle_up and *angle_left* are used to resolve the angle ambiguity. When the detected angle is closer to 0 or π than to $\pm\pi/2$, *angle_left* is used to resolve the ambiguity, otherwise *angle_up* is used. While either of these values on their own are mathematically sufficient, the network's accuracy on *angle_up* suffers when the needle is near horizontal, and likewise for *angle_left* when the needle is vertical. Choosing which to consider based on the needle angle avoids using the network output in cases where it is less accurate.

tip_visible and *tail_visible* are used to account for occlusion. When these values indicate both endpoints are visible, the center is computed based on the average of the endpoints. However, when only one is visible, the center is computed based on the location of the visible endpoint, the predicted angle, and the known needle dimensions. This allows the needle localization to accurately identify the needle location even with significant occlusion. In the final case, both endpoints are occluded and we compute a best guess at the needle position based on the average of the endpoints.

IV. Needle Motion Control

This section addresses generation of coil currents to allow the needle to execute a desired motion plan. In particular, we design the motion of the needle in a way that would enable the needle to perform a series of suture patterns, such as ligation and running sutures. We first describe how the needle's motion is affected by the coil currents.

A. Nonholonomic Model

As detailed in [22], and included here for completeness, we assume a dipole model of interaction between the needle and the magnetic fields generated by the coils. Under the dipole model, the magnetic field $\mathbf{B}$ generated by the k^{th} coil at the location of the needle's center of mass, $r = [x, y]^T$ is

$$\mathbf{B}_k = -\frac{m_k}{\delta_k^3}(\hat{r}_k - 3\hat{d}_k\hat{d}_k^T\hat{r}_k)I_k$$

where m_k is a constant related to the magnetic properties of the coil, r_k is the center location of the coil, $\hat{r}_k = r_k/\|r_k\|$ is a unit vector. The vector that points from the coil's center to the needle's center of mass is $d_k = r - r_k$, the distance between the coil and the needle is $\delta_k = d_k/\|d_k\|$, $\hat{d}_k = d_k/\delta_k$ is a unit vector, and I_k is the input current at the coil.

The torque and force experienced by the needle due to the magnetic fields generated by the coils is given by the cross product between the needle's magnetic moment vector ($\mathbf{M}(\theta) = Mh$, where M is a constant related to the needle's magnetic properties and $h = [\cos\theta, \sin\theta]^T$ is a unit vector pointing in the direction of the tip) and the coil's magnetic field, and the gradient of the magnetic potential, respectively,

$$\tau = \sum_{k=1}^{4} \mathbf{M}(\theta) \times \mathbf{B}_k, \qquad F = -\sum_{k=1}^{4} \nabla \left(\mathbf{M}(\theta)^T \mathbf{B}_k\right).$$

The needle motion is assumed to be first-order due to laminar fluid flow from low Reynolds number and negligible inertia terms [26], resulting in

$$\dot{r} = \tfrac{1}{c_t} F, \qquad\qquad \dot{\theta} = \tfrac{1}{c_r}\tau$$

where the coefficients c_t and c_r map forces and torques to translational and rotational motion (obtained empirically).

We require that the needle's translational motion to be kinematically constrained along the length of the needle, ensuring the force is applied along the tip of the needle [22]. The nonholomic model becomes

$$\begin{bmatrix} \dot{r} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} hh^T \frac{F}{c_t} \\ \frac{\tau}{c_r} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} g(r,\theta) I = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$

where needle's linear and angular velocity are given by

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = g(r,\theta) I = \sum_{k=1}^{4} g_k(r,\theta) I_k$$

and where if S is a skew-symmetric rotation matrix,

$$g_k(r,\theta) = \begin{bmatrix} \frac{3Mm_k}{c_t} \frac{\left(2\hat{d}_k^T hh^T \hat{r}_k + \hat{r}_k^T \hat{d}_k - 5\left(\hat{d}_k^T h\right)^2 \hat{d}_k^T \hat{r}_k\right)}{\delta_k^4} \\ \frac{Mm_k}{c_r} \frac{\left(h^T S \hat{r}_k - 3h^T S \hat{d}_k \hat{d}_k^T \hat{r}_k\right)}{\delta_k^3} \end{bmatrix}.$$

Thus, the $g_k(r,\theta)$ term, computed in real-time, when multiplied by the respective coil currents, will generate the desired forces and torques, which are directly mapped to linear and angular velocities of the needle. Given a desired linear and angular velocity commands (v, ω), the input currents can be found by employing a pseudo-inverse of the motion vector field relating these, i.e.,

$$I = g(r,\theta)^T \left(g(r,\theta) g(r,\theta)^T\right) \begin{bmatrix} v \\ \omega \end{bmatrix}.$$

B. Tip-Tracking Control

In this work, we are interested in accurately controlling the tip of the needle through the dish. This will be important to ensure the tip penetrates tissue at the desired location as well as avoiding inadvertently damaging surrounding tissue. To this end, we now derive a tip-tracking controller to allow the tip of the needle to track a desired reference trajectory.

Let $r_{tip} = [x_{tip}, y_{tip}]^T$ be the position of the tip relative to the center position of the needle, r, which is provided by the localization algorithm. As the needle is rigid, the location of the tip is related to the center via

$$r_{tip} = r + \frac{\ell}{2} \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}$$

where ℓ is the length of the needle. Differentiating these expressions, we get that

$$\dot{r}_{tip} = \dot{r} + \frac{\ell}{2} \begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix} \omega = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} v + \frac{\ell}{2} \begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix} \omega$$

which suggests a unique mapping

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\frac{1}{\ell/2}\sin\theta & \frac{1}{\ell/2}\cos\theta \end{bmatrix} \dot{r}_{tip}.$$

Given a differentiable desired reference path $r_{des}(t)$, we employ a PD control strategy, with the gains being tuned separately for categories A,B,C and D, to enable tip tracking

$$\dot{r}_{tip} = k(r_{des}(t) - r_{tip}) + \dot{r}_{des}(t).$$

C. Motion Design

The reference desired path $r_{des}(t)$ is designed to test the controller and localization algorithms at different regions within the Petri dish and under various visual conditions as described in the next section. The path is designed to achieve a back-and-forth motion of the tip, e.g., as required for a running suture. The path is parameterized in time as

$$r_{des}(t) = \left\{ \tfrac{(t - t_{i\;\;1})}{(t_i - t_{i\;\;1})}(q_i - q_{i-1}) + q_{i-1}, \quad t \in [t_{i-1}, t_i) \right.$$

where $q_0, \ldots, q_m$ are a sequence of points that define the shape of the path, $t_i = t_{i-1} + \|q_i - q_{i-1}\|/v_{des}$ with $t_0 = 0$, and v_{des} is the desired speed of the reference point through the path. The derivative of the path is given by

$$\dot{r}_{des}(t) = \left\{ v_{des} \tfrac{(q_i - q_{i\;\;1})}{\|q_i - q_{i\;\;1}\|}, \quad t \in [t_{i-1}, t_i) \right.$$

We set $v_{des} = 0.2mm/s$. We choose this speed to take into account the effects of static and dynamic friction that the needle experiences during its motion. The points $q_0, \ldots, q_m$ are selected to mimic a running suture pattern penetrating a tissue segment in the center region of the Petri dish, making three passes through the tissue. The tissue thickness used in the experiments is around 5 mm. The reference path and tracking performance are illustrated in Fig. 8.

V. Experimental Setup

We created a simulated surgical site in the Petri dish using fake blood (Liquid Latex Fashions Inc.) and synthetic tissues (Abdominal Tissue Plate, SynDaver Inc.) to collect datasets for training the segmentation network and test the localization method and the needle control performance.

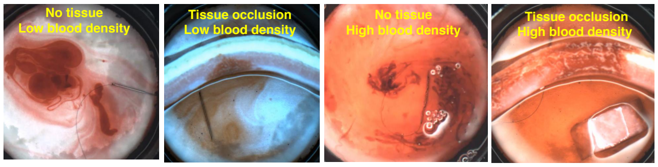


Fig. 5: Example datasets for training segmentation network.

A. Dataset Preparation for Training Segmentation Network

Various simulated surgical site conditions were set up by combining categories of different blood densities, with or without tissue occlusion, different fluid viscosity, and different illumination conditions (Fig. 5). For each condition, we recorded the magnetic needle motion created using manual joystick control to execute a random path [5].

Each training video was recorded at 60 FPS, then 75% of them were downsampled to 4 FPS due to lack of internal variation. The needle location and orientation in each training dataset frame were manually annotated and converted to ground truth masks. To maximize the variety of training data, we augmented it with random rotations from 0 to 2π and hue-saturation-luminance modifications of ±7.8%, ±11.7%, and ±7.8% respectively. In total we trained on 37,990 frames.

B. Evaluation of Segmentation Network

For the purpose of evaluating performance of the segmentation network and the localization method, we classified our dataset into 4 exclusive categories: **A**: no/low blood without tissue (2509 frames); **B**: heavy blood without tissue (3495 frames); **C**: no/low blood with tissue (11812 frames); **D**: heavy blood with tissue (13288 frames).

To reliably evaluate the performance of our segmentation network, we chose to use leave-k-out cross-validation We divide our 26 training videos into 3 groups and leave out between 7 and 10 videos for testing each time (the exact number being balanced based on number of total frames), training on the remaining videos (between 16 and 19). This gives a good sense of the performance of the network in completely unseen environments. We ran our experiments with a batch size of 4 on a GeForce RTX 2080. We report average statistics across all folds of the cross-validation.

VI. Results

A. Segmentation Performance

Fig. 6 shows the Intersection-over-Union (IoU), precision, and recall of the segmentation component and the accuracy, precision, and recall of the classification component. All results are averaged over leave-k-out folds, where the network is tested on completely unseen data.

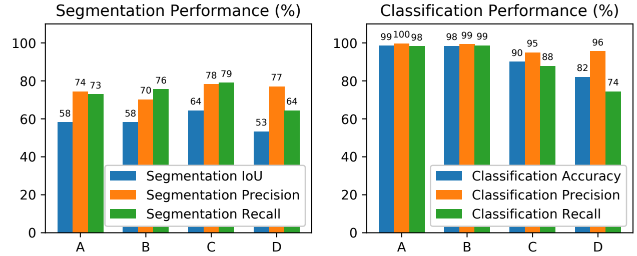


Fig. 6: Segmentation performance and classification performance. Categories A, B, C, D are defined in Sec. V-B.

Note that while precision and recall tend to be competitive, IoU varies between 53% and 64% on unseen videos, which appears low. For example, our encoder network achieved 74.7% IoU on the ImageNet dataset [20]. However, we expect some error is caused by the constraints of our annotation technique: we do not have pixel-perfect mask annotations, but rather extrapolate rectangular-needle ground-truth masks based on human location/orientation estimates and per-video tissue locations. Additionally, our masks cannot account for shifts in tissue location over the video, refraction through the medium changing the apparent size of the needle, heavy glare from the medium, or motion blur. Furthermore, we note that our localization process is far more sensitive to recall than to precision, since the pose detection steps effectively filter out a large degree of false positives but are less successful at compensating for false negatives.

The difference between categories A and B (see V-B) is within 5% on all three segmentation metrics, suggesting the network is well-equipped to deal with occlusion. The network also performs as well in category C, with occlusion but no blood, indicating that it does not rely on the fully visible needle to categorize needle pixels. As expected, category D performance is lower due to the difficulty of combining both blood and occlusion.

Classification performance on needle angle and tip/tail visibility is shown in Fig. 6, right. Categories A and B show highly similar results with near 100% accuracy, suggesting that the network successfully learns to identify the needle tip. In contrast with segmentation, performance decreases for categories C and D. This may indicate that the presence of blood impedes the classification's ability to identify the tip, as that information is required for all classification outputs.

B. Localization Performance

We evaluated the localization performance on a 6-core Intel® Core™ i7-9750H CPU and NVIDIA GeForce RTX 2080 Max-Q GPU. The average time per frame for the entire localization procedure was 22.37ms, with a 99th percentile time of 31.04ms, giving an average frame rate of 44.7 FPS.

Fig. 7 shows the performance of the localization in the four testing categories. The misdetection rate is divided into *No Detection* and *Incorrect Detection*. Both errors can be corrected for rather easily using filtering over time.

The *Incorrect Detection* rate is negligible in all but the first category, where a single video in the dataset contributes 86% of the incorrect detections. In this video, which is not well represented in the training data, the needle moves to the very edge to the dish, causing it to be difficult to detect even for humans. If this case is excluded the incorrect detection rate is only 0.22%, similar to the other cases. The *No Detection* rate is negligible in the first category. In the presence of tissue (category B), the *No Detection* rate increases significantly to 18%, and heavy blood (category C) moderately increases the rate to 8.2%. With heavy blood and tissue (category D) the *No Detection* rate is at its highest at 29%. This suggests

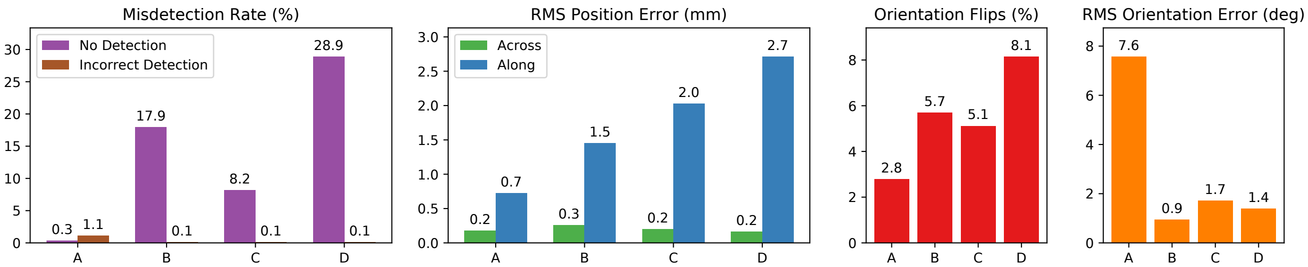


Fig. 7: Localization performance in Categories A-D (see Sec. V-B). The RMS Orientation Error represents error after correcting flips.

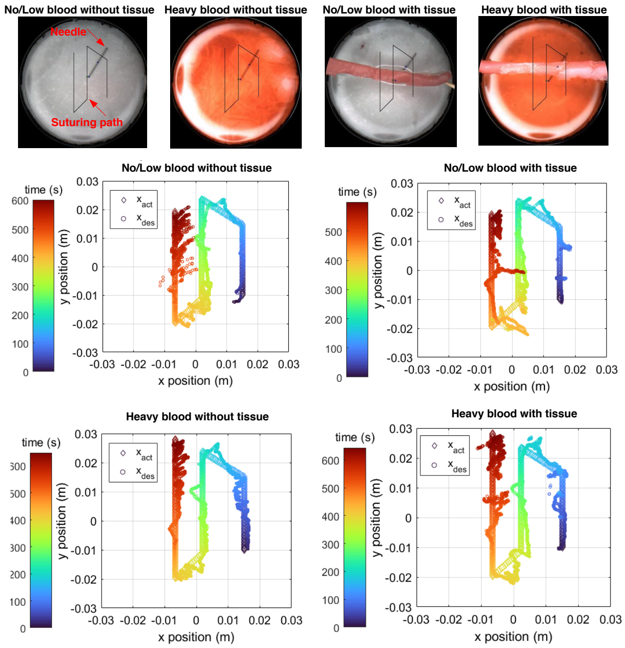


Fig. 8: Experimental results of needle tip-tracking control performing a running suture path using the proposed localization algorithm.

that both occlusion and heavy blood impact the detection's success rate separately, and those errors compound when both conditions are present.

Position errors are divided into "along needle" and "across needle", parallel and perpendicular to the needle direction, respectively. Note that our method is significantly more accurate in the "along needle" direction: the root-mean-square (RMS) error in the across direction is significantly below 1mm in all cases, indicating a high reliability in correctly identifying a sufficient number of needle pixels to allow RANSAC to locate the needle line. However, the RMS error along the needle has higher variance, indicating a difficulty in determining the precise location of the needle. As the cases increase in difficulty, the RMS error increases, representing also an increase in the rate of classification error, since the classification outputs are used to determine the full extent of the needle when the tip or tail is occluded.

Needle flips induce many orientation errors, where the predicted orientation is off 180° from the correct orientation. This distinct failure mode can be detected and mitigated by assuming the orientation has not changed more than 180° from the previous frame, and so we report the rate of orientation flips separately from the flip-corrected orientation error. Orientation flips occur in approximately 3% of frames in the first category, 6% with either blood or occlusion, and reach a maximum of 8% with blood and occlusion. As with the RMS error across the needle, this likely indicates the network's failure to identify the needle tip in more difficult cases. The RMS orientation error, which is computed on the flip-corrected orientation, is below 2° in Categories A, B, and C. The high value for Category A is caused by the same video which caused errors in *Incorrect Detection*. Excluding this video, the RMS orientation error for category A is 0.81°. Aside from this outlier, the network is highly successful at detecting orientation once flips are corrected. This again indicates that our method can successfully detect the needle line, and that the greatest source of error is determining the needle's location along that line. Using a network which classifies the needle tip and tail separately from the body, rather than encoding this information in auxiliary outputs, may improve flip and occlusion handling. However, robustly detecting true needle endpoints from potentially noisy segmentation masks would require additional complexity.

We also compared the proposed needle localization method to an unpublished iteration of the method presented in our previous work [22]. This method can only localize the suturing needle with no suture thread in the clean Petri dish with plain white background. Across 2,000 frames in those conditions the previous method achieves a RMS error of 6.18mm, compared to only 0.61mm with the proposed method. With the suture thread present, the previous method reports "no detection" in 68% of frames compared to 0.64% for the proposed method, and on the remaining frames achieves an RMSE error of 11.7mm compared to 0.86mm for the proposed method. It is clear that the previous method cannot localize the needle in any environment more complex than the clean, empty Petri dish.

C. Motion Control Performance

We experimentally validated the use of the proposed localization method and the controller under 4 types of environments: No/low blood without tissue, Heavy blood without tissue, No/low blood with tissue, and Heavy blood with tissue. The path tracking performance of the needle-tip for these conditions is provided in Fig. 8. RMS errors in position tracking are found to be 3.0, 2.6, 3.7, 2.9 mm for the aforementioned 4 conditions, respectively. One of the main reasons for such errors is the nonlinear Coulomb surface friction. Additionally, the dipole model used to compute the magnetic fields and control efforts is not accurate enough for the locations far from the center of the dish. We believe these two elements are the major contributors to the tracking errors we observe during the experiments.

VII. Conclusion

We present a novel needle localization method, based on a U-Net modified for speed and additional output combined with error-correcting classical methods, and a corresponding control method which uses our localization method to navigate a needle in a lab environment. While our control method is tailored to our specific magnetic suturing setup, our needle localization method is applicable to any other similar contexts.

Our needle localization method is able to localize the needle in over 70% of frames even in the most challenging environment we considered, under the presence of blood and tissue. For this challenging situation, the localization RMS error is 2.7 mm. For white background with no blood and the tissue, the average RMS localization error is 0.73 mm. Overall average of RMS localization error is 2.17 mm. The needle motion control shows 2.6 to 3.7 mm RMS errors in the desired tip position over the course of a running suture path. The reliable localization and control method proposed in this study would pave the way towards autonomous wireless suturing tasks. In the future we will extend the system to 3D by adding additional magnetic z-coils and 3D localization and will incorporate model prediction (e.g. Kalman observer) to handle more occlusions for real surgical applications.

Acknowledgments

We thank Margaret Pryor, who assisted with data labeling.

References

[1] M. Sitti, H. Ceylan, W. Hu, J. Giltinan, M. Turan, S. Yim, and E. Diller, "Biomedical applications of untethered mobile milli/microrobots," *Proceedings of the IEEE*, vol. 103, no. 2, pp. 205–224, Feb 2015.

[2] C. Heunis, J. Sikorski, and S. Misra, "Flexible instruments for endovascular interventions: Improved magnetic steering, actuation, and image-guided surgical instruments," *IEEE Robotics Automation Magazine*, vol. 25, no. 3, pp. 71–82, Sep. 2018.

[3] X. Liu, G. J. Mancini, Y. Guan, and J. Tan, "Design of a magnetic actuated fully insertable robotic camera system for single-incision laparoscopic surgery," *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 4, pp. 1966–1976, Aug 2016.

[4] O. Erin, D. Antonelli, M. E. Tiryaki, and M. Sitti, "Towards 5-dof control of an untethered magnetic millirobot via MRI gradient coils," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 6551–6557.

[5] L. O. Mair, X. Liu, B. Dandamudi, K. Jain, S. Chowdhury, J. Weed, Y. Diaz-Mercado, I. N. Weinberg, and A. Krieger, "MagnetoSuture: Tetherless Manipulation of Suture Needles," *IEEE Transactions on Medical Robotics and Bionics*, vol. 2, no. 2, pp. 206–215, May 2020.

[6] M. Najafi, P. Abolmaesumi, and R. Rohling, "Single-Camera Closed-Form Real-Time Needle Tracking for Ultrasound-Guided Needle Insertion," *Ultrasound in Medicine & Biology*, vol. 41, no. 10, pp. 2663–2676, Oct. 2015.

[7] A. Reiter, P. K. Allen, and T. Zhao, "Feature Classification for Tracking Articulated Surgical Tools," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2012*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, vol. 7511.

[8] M. Alsheakhali, A. Eslami, and N. Navab, "Detection of articulated instruments in retinal microsurgery," in *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*, Apr. 2016, pp. 107–110, iSSN: 1945-8452.

[9] A. Nakazawa, K. Harada, M. Mitsuishi, and P. Jannin, "Real-time surgical needle detection using region-based convolutional neural networks," *International Journal of Computer Assisted Radiology and Surgery*, vol. 15, no. 1, pp. 41–47, Jan. 2020.

[10] H. Sugimori, T. Sugiyama, N. Nakayama, A. Yamashita, and K. Ogasawara, "Development of a Deep Learning-Based Algorithm to Detect the Distal End of a Surgical Instrument," *Applied Sciences*, vol. 10, no. 12, p. 4245, Jun. 2020.

[11] P. Sundaresan, B. Thananjeyan, J. Chiu, D. Fer, and K. Goldberg, "Automated Extraction of Surgical Needles from Tissue Phantoms," in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, Aug. 2019, pp. 170–177.

[12] S. Speidel, A. Kroehnert, S. Bodenstedt, H. Kenngott, B. Müller-Stich, and R. Dillmann, "Image-based tracking of the suturing needle during laparoscopic interventions," in *Medical Imaging 2015: Image-Guided Procedures, Robotic Interventions, and Modeling*, Orlando, Florida, United States, 3 2015, p. 94150B.

[13] Y. Chen, M. M. Marinho, A. Nakazawa, K. Deie, K. Harada, M. Mitsuishi, and Y. Kurose, "Towards robust needle segmentation and tracking in pediatric endoscopic surgery," in *Medical Imaging 2018: Image-Guided Procedures, Robotic Interventions, and Modeling*. Houston, United States: SPIE, Mar. 2018, p. 96.

[14] J. Chang, "Robust needle recognition using Artificial Neural Network (ANN) and Random Sample Consensus (RANSAC)," in *2012 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, Oct. 2012, pp. 1–3.

[15] L. Chen, G. Papandreou, and et al., "Rethinking atrous convolution for semantic image segmentation," *CoRR*, vol. abs/1706.05587, 2017. [Online]. Available: http://arxiv.org/abs/1706.05587

[16] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, 2015, pp. 234–241.

[17] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation," *arXiv:1606.02147 [cs]*, Jun. 2016, arXiv: 1606.02147. [Online]. Available: http://arxiv.org/abs/1606.02147

[18] T. Lei, W. Zhou, Y. Zhang, R. Wang, H. Meng, and A. K. Nandi, "Lightweight v-net for liver segmentation," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 1379–1383.

[19] Z.-L. Ni, G.-B. Bian, Z.-G. Hou, X.-H. Zhou, X.-L. Xie, and Z. Li, "Attention-Guided Lightweight Network for Real-Time Segmentation of Robotic Surgical Instruments," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, May 2020, pp. 9939–9945.

[20] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.

[21] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

[22] M. Fan, X. Liu, K. Jain, D. Lerner, L. Mair, I. Weinberg, Y. Diaz-Mercado, and A. Krieger, "Towards autonomous control of magnetic suture needles," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 10 2020, pp. 2935–2942.

[23] C. H. Sudre, W. Li, and et al., "Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, 2017, pp. 240–248.

[24] A. Paszke, S. Gross, and et al., "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, 2019, pp. 8024–8035.

[25] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. AAAI Press, 1996, pp. 226–231.

[26] M. Salehizadeh and E. Diller, "Two-agent formation control of magnetic microrobots in two dimensions," *Journal of Micro-Bio Robotics*, vol. 12, no. 1-4, pp. 9–19, 2017.