2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

# Estimating the Joint Angles of a Magnetic Surgical Tool using Monocular 3D Keypoint Detection and Particle Filtering

Erik Fredin and Eric Diller

***Abstract*— Magnetic surgical tools benefit greatly from real-time pose estimation, as this is essential for controlling them safely and effectively. Current pose estimation methods for surgical tools either focus on rigid tools, or are developed specifically for the da Vinci surgical system. In this work, we use computer vision from a monocular endoscopic camera to estimate the pose of an articulated magnetic surgical tool. In particular, we present a deep 3D keypoint estimation framework and a particle filter to achieve this. The former method can be used for any articulated surgical tool, while the latter method is specific to magnetic tools. We show that the deep 3D keypoint estimation framework estimates the surgical tool's joint angles with an average error of 4.0 degrees and a speed of 29 Hz. In addition, we demonstrate the robustness of the magnetic particle filter and the deep pose estimation method for real-time tool pose estimation.**

## I. Introduction

Minimally invasive neurosurgery has been subject to increasing research in recent years, which promises neurosurgical procedures completed with smaller incisions, shorter hospitalization stays and reduced trauma. We recently proposed a magnetic serial robot (MSR) for minimally invasive neurosurgery [1], consisting of a magnetic end effector attached to a delivery platform (Fig. 1). The tool consists of 2 revolute flexure joints, as well as two on-board magnets. A magnetic field is used to wirelessly move the tools joints, as the tools' magnets align themselves with the external field. A recently proposed magnetic coil system [2] generates such fields without restricting the workspace available to surgeons. This setup enables these tools to have a diameter of only 4 mm, while maintaining full dexterity. However, magnetic actuation of such multi-jointed tools requires feedback on the tools' joint angles ($q_1$ and $q_2$) for safe and effective control [1]. Endoscopic computer vision methods are promising to this end. The delivery platform already includes an endoscopic camera (Fig. 1), and therefore computer vision allows for pose estimation without altering the physical setup.

Pose estimation methods for surgical tools using computer vision have been studied since the 2000s. Early work uses filtering and thresholding techniques to manually extract keypoint and segmentation data [3][4]. However, these vision methods lack robustness in the common situation of self-occlusion, blood or occlusion from tissue. Deep-learning methods on the other hand have been able to track visual features of surgical instruments in images with excellent results [5]. The orientation of rigid surgical instruments has been detected using deep-learning models, yielding average errors from 5-6° [6][7]. However, these methods specifically designed for rigid instruments cannot easily be extended to articulated tools. Allan et al. proposed a gradient-based optimization method for estimating the pose of an articulated tool from monocular images [8]. However, this method is neither accurate or fast enough to suit the requirements for controlling MSRs in real time. Sestini et al. recently achieved a much greater accuracy by using joint angle encoder data to train models in a self-supervised manner [9]. This approach yielded an average angular error 4.8°, and avoids manual labelling of data. MSRs however do not have access to joint angle encoder data, and thus cannot use this method.

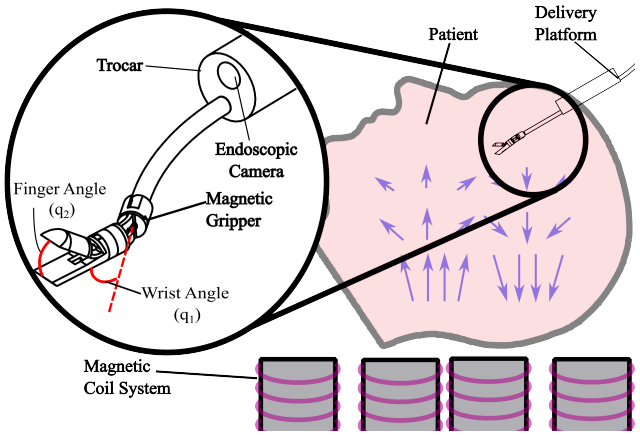


Fig. 1. Vision of in *vivo* pose estimation for magnetically acutated surgical tools. A monocular camera is placed in a trocar to estimate a small 2-DoF surgical tool.

Other studies [10][11] have shown markerless keypoint detection using deep neural networks (DNNs) to be a promising technique for estimating the 3D pose of surgical tools. This poses the problem of re-projecting keypoints detected in a 2D image plane into 3D world space, which is non-trivial for monocular vision. To this end, the human pose estimation (HPE) community often uses DNNs to predict 3D keypoint locations given their respective 2D pixel coordinates [12]. Applying deep 3D keypoint detection to estimating the pose of surgical tools thus has the potential to outperform more standard methods such as stereo triangulation, while also enabling monocular 3D keypoint estimation.

As an alternative to direct joint angle estimation using

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada through the Discovery Grant Program 2014-04703 and in part by the Canadian Institutes for Health Research under Grant CPG-158271. (*Corresponding author: Eric Diller.*)

Department of Mechanical and Industrial Engineering, University of Toronto, 5 King's College Rd, Toronto, ON M5S 3G8 ediller@mie.utoronto.ca

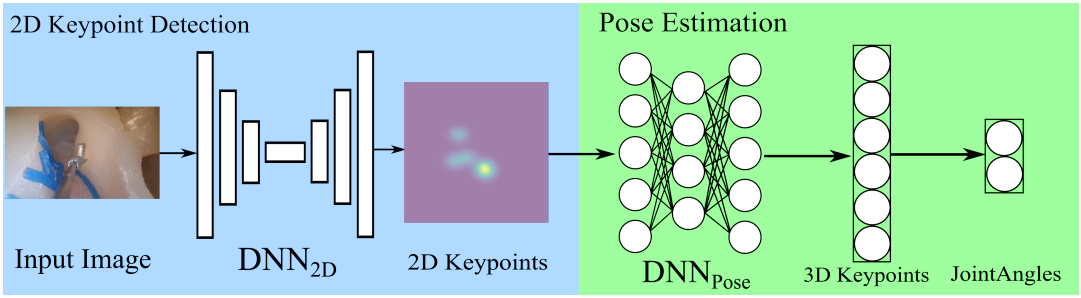


Fig. 2. The proposed deep method for joint angle estimation. $DNN_{2D}$ (architecture based on [13]) detects 2D keypoint probability density maps. These are thereafter converted to 2D coordinates, and converted to 3D world space by $DNN_{pose}$. The architecture of $DNN_{pose}$ was proposed by Martinez et al. [20]. Finally, joint angles are calculated analytically from detected 3D keypoints.

computer vision, state estimators such as Kalman filters or particle filters have been used with great success in surgical robotics [14] [15] [16]. These filters typically use 2D features in the image plane detected using computer vision together with a measurement function to provide a state estimate. This approach is well-suited for monocular vision, in addition to reducing the final joint angle estimates' susceptibility to noise and occasional failures. However, application of these filters to magnetic robot tracking remains limited. Some studies have applied the Unscented Kalman Filter (UKF) [17] to the tracking of magnetic systems [18], [19]. These typically incorporate a mathematical model of the field measured by magnetic sensors. There are few studies modelling the motion of magnetic robots in a process model. One such study models the planar motion of spherical magnetic janus microrobot particles [20] with no articulating joints. So-far, no state estimators have been devised for articulated magnetic tools.

In summary, current pose estimation methods for surgical tools are either tailored to specific tools or rely on joint angle encoder data. Tools that do not have access to such encoder data require alternative methods. In this work, we thus propose two methods for estimating the joint angles of a magnetic surgical tool, presenting the following contributions:

1) A deep method for estimating the 3D pose of a multi-jointed surgical tool from monocular images, demonstrating an accuracy of 3.95 degrees, a speed of 29 Hz and that it outperforms stereo triangulation,
2) A particle filter for estimating the state of magnetic serial robots, demonstrating its robustness in estimating the tool pose in real-time tracking trials.

## II. Deep Pose Estimation (DPE)

Fig. 2 shows the proposed method for estimating the joint angles. This is broken down into two stages: 2D keypoint detection, and pose estimation. The 2D detection stage estimates the pixel coordinates of markerless keypoints in a given monocular image. The markerless keypoints are located at the end of each link of the tool, resulting in a total of 6 keypoints. The probability distribution of each keypoint is predicted as a heatmap. Due to its simplicity and robustness, a model proposed by Xiao et al. [13] for HPE is used to do this. The deconvolutional layers of this model have been modified to match the desired heatmap output size (224x244 pixels). The pose estimation stage regresses the 3D coordinates of the markerless keypoints. The DNN chosen for this was proposed by Martinez et al. [21] for HPE. In the final stage of the method, 3D keypoint locations are used to calculate the tool's joint angles analytically.

### A. 2D Keypoint Detection

The 2D detection model is trained using a large quantity of simulated data and fine-tuned using a smaller amount of real data. Fig. 3 (b) shows an example of a real image. The process for collecting the real training data is described in Section IV. The simulated training dataset is generated using the game engine Unity (Unity Technologies, San Francisco, USA) and its perception package [22]. A domain randomization approach [23] was chosen to facilitate the synthetically-trained model's generalisation to real data. The background, lighting, tool texture and camera position were randomized, and only the rods texture was designed to mimic that of the real dataset. Fig. 3(c) shows an example of a synthetic training image. In total, 20 000 synthetic images were used to train the model. The model was trained on synthetic data for 88 epochs and for 90 epochs on real data.

### B. 3D Pose Estimation

*1) Geometric Loss Function:* $DNN_{pose}$ does not predict 3D keypoint coordinates accurately enough when trained with a conventional $\mathcal{L}_1$ loss. To address this, a geometric loss function is proposed to train the DNN in accordance with the tool's geometry. The motivation for this is illustrated in Fig. 3(a). A link's orientation, denoted $\boldsymbol{l}_x$, is given by two keypoints on its center axis, $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$. Their estimated counterparts, $\hat{\boldsymbol{x}}_a$ and $\hat{\boldsymbol{x}}_b$, make up the estimated link orientation $\hat{\boldsymbol{l}}_x$. As shown in Fig. 3(a), estimating 3D keypoints without considering such geometric constraints leads to estimated link orientations violating them. The estimated link orientation $\hat{\boldsymbol{l}}_x$ points downwards, even though this should not be possible given the tools configuration space. Errors in detected 3D keypoints can thus have an outsized impact on the estimated joint angles. A geometric loss

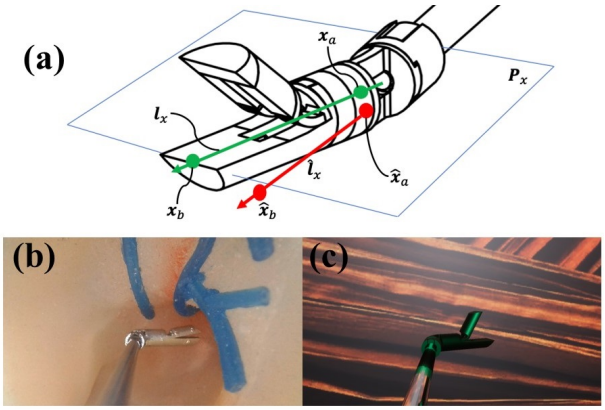


Fig. 3. (a) Illustration of loss function that reinforces geometric constraints of the surgical tool, (b) example of real image of tool, (c) example of synthetic training image.

function is therefore proposed to address this:

$$\mathcal{L}_{geo} = \mathcal{L}_1(\boldsymbol{x}, \hat{\boldsymbol{x}}) + \mathcal{L}_1(\boldsymbol{l}, \hat{\boldsymbol{l}}) + \frac{1}{N}\sum_{i=n}^{N} |\theta_{\hat{\boldsymbol{l}}_n, \boldsymbol{P}_n}|, \quad (1)$$

where $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ are the ground truth and estimated keypoint coordinates, $\boldsymbol{l}$ and $\hat{\boldsymbol{l}}$ are the ground truth and estimated link orientation vectors, and $\theta_{\hat{\boldsymbol{l}}_n, \boldsymbol{P}_n}$ represents the angle that an estimated link $\hat{\boldsymbol{l}}_n$ makes with the plane $\boldsymbol{P}_n$. All terms in (1) use an $\mathcal{L}_1$ loss to compute estimation errors for their respective parameters. The first term regresses the direct output of the network, i.e., the 3D keypoint coordinates, while the second term penalizes incorrect link orientations. The second term is particularly important, since joint angles are determined using estimated link orientations. Finally, the last term computes the angle that estimated links make with the plane that their respective ground-truth link rotates in. This term penalizes keypoint predictions that are inconsistent with the configuration space that a given link is able to rotate in.

*2) Synthetic 3D Data Generation:* While a vast dataset is needed for training $DNN_{pose}$, collecting this experimentally is highly challenging. Therefore, 2D and 3D keypoints pairs are instead generated by simulating the robots kinematics. For a given keypoint $P_i$ with known coordinates with respect to its link's coordinate frame j, the keypoint's 3D coordinates in the camera frame are given by:

$$\boldsymbol{P}_i^C = \boldsymbol{T}_0^C \prod_{n=1}^{j} \boldsymbol{T}_n^{n-1}(q_n)\boldsymbol{P}_i^j, \quad (2)$$

where $\boldsymbol{T}_0^C$ is the robot base to camera transform, $\boldsymbol{T}_n^{n-1}(q_n)$ is the transform from the frame of link n to the previous link $n-1$, $q_n$ is the angle of joint n, and $\boldsymbol{P}_i^j$ is the location of keypoint i with respect to the frame of link j. $\boldsymbol{P}_i^j$ is obtained from the robots CAD model, while $\boldsymbol{T}_n^{n-1}(q_n)$ can be calculated via kinematic modelling. $\boldsymbol{T}_0^C$ is determined using ArUco markers placed in a known configuration on a board mounted onto the rod and an iterative PnP algorithm. Finally, 3D keypoint coordinates in the camera frame are transformed into 2D pixel coordinates via the cameras intrinsic matrix, $\boldsymbol{K}$:

$$\boldsymbol{p}_i = \boldsymbol{K}\boldsymbol{P}_i^C, \quad (3)$$

where $\boldsymbol{p}_i$ is the 2D pixel coordinates of keypoint $P_i^C$ and $\boldsymbol{K}$ is the intrinsic matrix. $\boldsymbol{K}$ is calibrated using Zhang's method [24]. Using this approach, 3D keypoints $\boldsymbol{P}^C$ and their corresponding 2D pixel coordinates $\boldsymbol{p}$ can be generated for a wide array of joint angles $\boldsymbol{q}$. In a real-world setup however, errors occur in calibrating the robot-to-camera transform, $\boldsymbol{T}_0^C$, and estimating 2D keypoints $\boldsymbol{p}$. To account for this, (2) and (3) are modified to add noise to both of these parameters:

$$\boldsymbol{p}_i = \boldsymbol{K}((\tilde{\boldsymbol{T}}\boldsymbol{T}_0^C) \prod_{n=1}^{j} \boldsymbol{T}_n^{n-1}(q_n)\boldsymbol{P}_i^j) + \tilde{\boldsymbol{p}}_i, \quad (4)$$

The added noise increases the likelihood that a subset of the training data reflects the testing environment. This improves the potential for $DNN_{pose}$ to generalize to test conditions. 100 000 training samples were generated using this approach, and the model was trained for 110 epochs with a learning rate of 1e-4.

## III. Particle Filter for Magnetic Serial Robots

Particle filters use a finite set of particles to represent posterior probability density function. In robotics, each particle represents a state hypothesis that is simulated via a motion model of the robot. The probability that a given particle represents the robots true state is thereafter computed via a measurement function and the true sensor measurement.

### A. Motion model

The motion model for the robot is derived by modelling its flexure joints as an overdamped spring-damper system under an external load. The external load for this system is the magnetic torque applied to the tool for actuation, as well as the internal magnetic forces between on-board magnets. Specifically,

$$\boldsymbol{M}_u(\boldsymbol{q})\boldsymbol{u} + \boldsymbol{\tau}_{int}(\boldsymbol{q}) = \boldsymbol{c}\dot{\boldsymbol{q}} + \boldsymbol{k}\boldsymbol{q}, \quad (5)$$

where $\boldsymbol{M}_u(\boldsymbol{q})$ is the current actuation matrix, $\boldsymbol{u}$ are the magnetic coil currents and $\boldsymbol{\tau}_{int}(\boldsymbol{q})$ are the internal forces of the MSR due to its on-board magnets. $\boldsymbol{u}$ is the known control input to the system, while expressions for $\boldsymbol{M}_u(\boldsymbol{q})$ and $\boldsymbol{\tau}_{int}(\boldsymbol{q})$ have previously been derived in [25]. The calibration process for the spring and damping constants $\boldsymbol{k}$ and $\boldsymbol{c}$ is described in Section IV-B. Notably, we did not include an acceleration term in (5). The reason for this is that we assume that the tools acceleration will be comparatively low. While this assumption does not always hold true, we design the particle filter such that this inaccuracy can be compensated for. Expressing (5) in discrete-time yields the particle filters process model:

$$\boldsymbol{q}_{k+1} = \boldsymbol{q}_k + \boldsymbol{c}^{-1}(\boldsymbol{M}_u(\boldsymbol{q}_k)\boldsymbol{u}_k + \boldsymbol{\tau}_{int}(\boldsymbol{q}_k) - \boldsymbol{k}\boldsymbol{q}_k)\Delta t, \quad (6)$$

where $\Delta t$ is the timestep between particle filter iterations.

### B. Importance sampling

Importance sampling quantifies the proximity between 2D keypoints estimated using a measurement function and the corresponding 2D keypoints detected using $DNN_{2D}$. This proximity corresponds to the probability that the particles state is the correct state, also known as that particles weight. (2) and (3) are used as the measurement function, which estimates the 2D keypoints given a particle's state $\boldsymbol{q}$.

### C. Implementation

Importance resampling is commonly used in particle filters to avoid degeneracy, where some particles weights are computed to be near-zero. The particle filter calculates how many weights are effective for each iteration, and applies stratified resampling if the number of effective weights is too low. However, this can lead to the opposite problem, where many or all particles are resampled to have the exact same state and equal weights. In practice, this yields the same result as if a single particle had been used, and is not representative of the probability distribution of the robots state. This problem is particularly prevalent when a low number of particles is used. To counteract this, we add a small amount of gaussian noise $\sim N(0, Q)$ to (6). This way, the particles avoid converging to the same value and are more likely to represent the true distribution of the tools pose. This comes at the expense of the particle filter being more susceptible to temporary measurement noise, however, in practice this can be reduced by setting Q to a low value.

## IV. Datasets & Experiments

### A. Rigid Tool Dataset

A key challenge in testing the deep pose estimation (DPE) method is acquiring data where the tools ground truth joint angles are known precisely. Accurate ground truth is also a challenge in surgical computer vision in general. We address this by 3D printing many different rigid samples of the tool with known joint angles. The rigid tools are painted silver to visually resemble machined surgical tools. The degree to which the joint angles of a 3D printed tool matched that of its CAD model was validated using an Epilog laser scanner. This showed that the joint angles of the rigid tools deviated from their CAD model by at most $\pm$ 2 deg. In order to ensure that the configuration space is represented adequately in the test dataset, each joint angle is increased in steps of 10. This resulted in 190 rigid surgical tools.

Fig. 4(a) shows the setup used for collecting the data. Framing as well as a shroud are used to mount tissue and block external lighting. The rod used to mount the tool includes an insert, so that the tool is unable to rotate around the rods axis. A phantom brain was chosen as background tissue. While a porcine brain strongly visually resembles a human brain, the use of one requires strict ethical approval and guidelines. A phantom brain does not come with these restrictions, and has been used as a background in other studies estimating surgical tool pose using computer vision [6]. The use of an animal cadaver brain is however desirable in future work. In total, 814 training images and 183 test images were collected using this method. The test images features the phantom brain as a background only, while the training data features various backgrounds including the phantom brain. This discrepancy was included to reflect the fact that a surgical workspace may not match the surgical training data perfectly. The DPE method is tested on the images of the right camera. We compare these results to the use of stereo triangulation for estimating 3D keypoints, as this is a well-known method in computer vision for projecting points in images into 3D space.

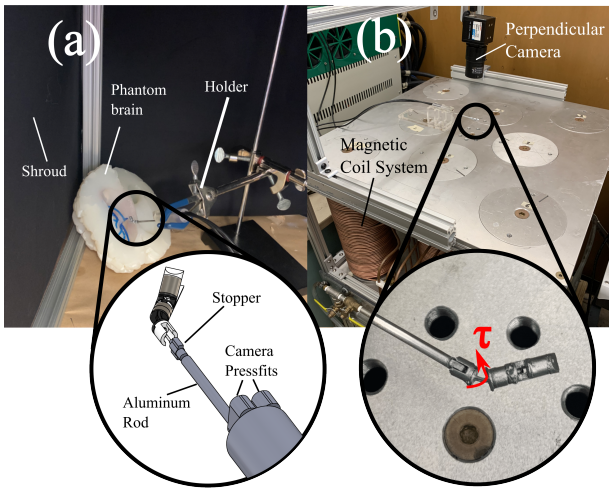


Fig. 4. (a) Setup for data collection (zoom in on trocar, show mounting of gripper), (b) Setup for calibrating the joint stiffness and damping coefficients.

### B. Time-Series Experiment

*1) Data Collection and Training:* To test the performance of both estimation methods on temporal data, three 50-80 second videos of the tool moving under magnetic actuation are collected. Two of the three trials use the phantom brain as a background and a shroud to block external lighting, while the final test video is recorded in our labs environment without a dedicated background. The trials in the phantom brain environment use dim and warm endoscopic lighting respectively, and the speed of the tool is varied across the three trials. Video frames of the tool are recorded synchronously with coil currents used for actuation. DPE by itself, DPE combined with a low-pass filter and the particle filter are tested on these. The low pass filter calculates the average pose from the previous 10 timesteps. The particle filter is initialized with 50 particles and a noise $Q$ of 2 radians. In addition, 877 training images are collected. The training dataset is comprised of each of three settings in the testing videos. The keypoints on these images are annotated by hand, and the real-world images are supplemented by 20k simulated images. $DNN_{2D}$ is trained for 70 epochs on simulated data and finetuned for 130 epochs on the real-world data. A learning rate of 0.0005 and a batch size of 100 were used each time.

*2) Ground-Truth:* In the absence of direct ground truth on the tools' joint angles, the IoU of the estimated and ground-truth segmentation data is computed, an approach to ground-truth previously used to test other pose estimation methods for surgical tools [14], [16]. The estimated tool segmentation is generated using the estimated joint angles and the tools' CAD model, while the ground-truth masks are hand-labelled. While a subset of the video frames are annotated and used to show IoU, inference of both methods is done on all video frames.

*3) Flexure Joint Identification:* To model the behavior of the robots flex joints, $\boldsymbol{k}$ and $\boldsymbol{c}$ in (6) are determined experimentally. Fig. 4(b) shows the calibration setup using the magnetic coil system. $\boldsymbol{k}$ is identified by applying a magnetic torque to a joint angle such that it is in equilibrium. Torques and joint angles are recorded for various equilibrium positions, and the collected data is used together with (5) to identify $\boldsymbol{k}$. This yields $k_1 = 0.0004$ Nm/rad and $k_2 = 0.00035$ Nm/rad. $\boldsymbol{c}$ is determined by placing joint angles at their limit positions and thereafter removing any external forces. The joint angles are measured over time using a perpendicular camera. This data is then used to fit the general solution for an overdamped spring-damper system, yielding $c_1 = 0.00098$ Nm·s/rad and $c_2 = 0.00069$ Nm·s/rad.

## V. Results and Discussion

### A. Rigid Tool Dataset

Fig. 5 shows the box plot data of the rigid-tool dataset. While some test images have high errors, the mean error for each joint angle is 3.1° and 4.8° respectively, while their standard deviations are 2.9° and 4.7°. This suggests that while Fig. 5 shows some samples with high errors, the joint angles were detected much more accurately for most of the dataset. However, a non-negligible number of samples have joint angle errors greater than 10°, with 2 samples having an error of over 20° for the second joint angle. The likely reason why the second joint angle is more error prone is that it is computed from keypoints at the tool's tips. These are more difficult to track, as their position changes considerably across the dataset relative to other keypoints. Detecting these keypoints more robustly would thus likely reduce the occurrences of high joint angle detection errors. For in *vivo* settings, low-pass filters could alternatively be used to reduce this methods susceptibility to occasional tracking failures. This method computes at an average rate of 29.4 Hz on an NVIDIA GeForce RTX4090 GPU, where $DNN_{2D}$ takes by-far the longest (33ms) to process. It it thus fast enough for real-time pose estimation.

Table I shows a comparison of deep pose estimation (DPE) to stereo triangulation. As shown in Table I, the use of $DNN_{pose}$ outperforms stereo triangulation for estimating the 3D pose of the tool. The most likely explanation for the poor performance of stereo triangulation is that, as discussed in Section II, errors in the estimated 3D keypoints can result in outsized joint angle detection errors. DPE mitigates this by using the geometric loss function, penalizing 3D keypoint estimates that are not located on center axis of each of the tool's links. Stereo triangulation would likely improve considerably if 2D keypoints are detected more accurately. This would however likely require tens-of-thousands of training images, rather than the 814 real-world images used in this experiment. Other pose estimation methods in surgical robotics [6], [9] also require tens of thousands of training images. We therefore believe that the proposed DPE method could present a compelling option when training data is scarce.

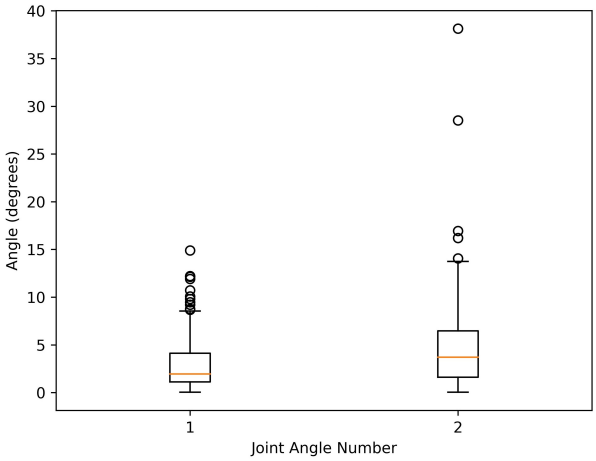


Fig. 5. Joint angle estimation results on the rigid-tool dataset.

TABLE I
Joint Angle Estimation Results

| Method | Mean error (deg) | St. Dev. (deg) |
|---|---|---|
| 2D Detector + triangulation | 18.85 | 17.95 |
| 2D Detector + $DNN_{pose}$ | 3.95 | 4.75 |

### B. Time-Series Data

Fig. 6 shows the IoU over time for unfiltered DPE (red), DPE combined with a low-pass filter (green), and the particle filter (blue). The particle filter seems to outperform the deep pose estimation method for most of the tools trajectory, with the exception of the time stretching from around 14 to 25 seconds. An example of this is shown in the top left of Fig. 6. The pose estimated using the deep pose estimation method (red) seems to match the wrist angle of the tool more closely than the particle filter (blue). The detected keypoints are shown in yellow. These are detected at the ends of each link for the top left example, matching their ground truth positions well. The DPE method benefits from this, and hence yields in a higher IoU for this timestep. In the top right example however, the detected keypoints have considerable errors. Here, DPE performs worse than the particle filter, as demonstrated by the latter's higher IoU. This suggests that the particle filter's use of temporal data combined with the motion model in (6) helps to accurately predict the gripper's pose. These features make the particle filter more robust to temporary measurement noise. In addition, DPE can at times be susceptible to errors stemming from $DNN_{pose}$.

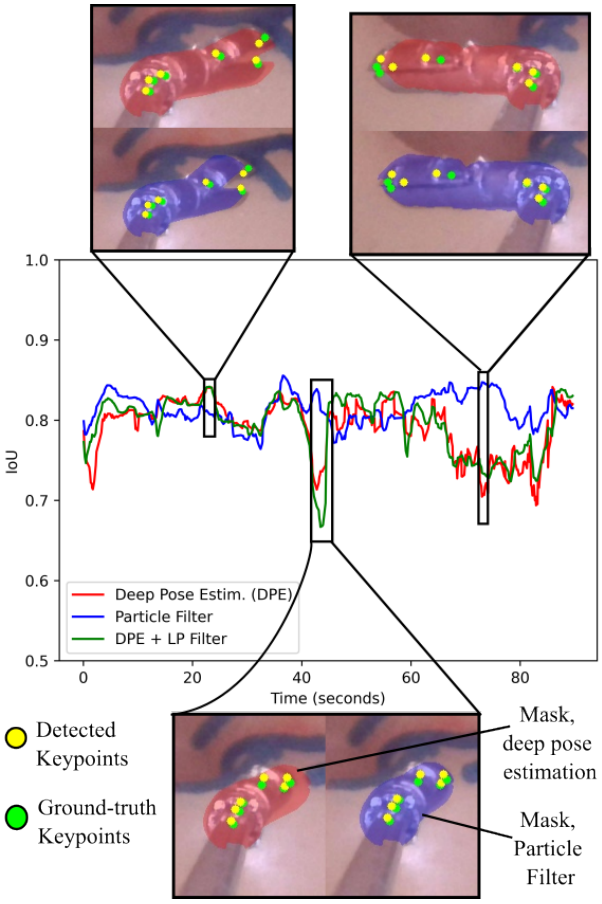


Fig. 6. Trial 2 results: the IoU from re-projecting the estimated tool state onto the image is shown over time for deep pose estimation (DPE, red), DPE and a low pass filter (green) and the particle filter (blue). The IoUs are averaged over 5 timesteps. Re-projected masks from DPE and the particle filter are shown in red and blue respectively, to illustrate how closely these match the tool's true pose. Estimated keypoints are shown in yellow.

This is demonstrated in the bottom example, where detected keypoints are placed closely to link endpoints. Despite this, the tool pose estimated using DPE still has a lower IoU than that estimated using the particle filter. Errors in $DNN_{pose}$ may also explain DPE yields noisier estimates than the particle filter.

The speed of the particle filter is up to 27 Hz, not including the time taken to detect 2D keypoints. In an in *vivo* scenario, multi-threaded programming could be used to run $DNN_{2D}$ and the particle filter on separate threads. This would yield pose updates at a maximum rate of 27 Hz, which is similar to the speed of DPE (29 Hz). Both methods are thus fast enough for controlling the magnetic tool in real-time [1].

Table II shows the tool tracking performance over the 3 recorded trials. While DPE still displays a high level of robustness across the three trials, it is less accurate than the particle filter in all of these. The particle filter estimated the tool's state adequately in Trial 3, where the tool experienced fast motion at certain times. This is despite the tools inertia not being modelled in (6), which could conceivably have lead to errors during high speeds. We believe that importance sampling via keypoint detection compensated for this modelling inaccuracy. If this is true, then the strength of the particle filter is derived from the fact that the both the motion model and keypoint detection can compensate for temporary failures in each other. DPE cannot take advantage of a motion model, and is thus less robust. However, the particle filter in this work is designed specifically for magnetic tools. The main strength of the DPE method is that it can be applied to a wide variety of surgical tools, as long as their articulation can be modelled using homogenous transformations. DPE thus presents a very versatile alternative to the use of a particle filter, as it does not require a motion model of a given tool.

TABLE II
RESULTS FOR 3 DIFFERENT TOOL TRACKING TRIALS

| Trial no. | Trial 1 | Trial 2 | Trial 3 |
|---|---|---|---|
| Lighting | Dim | Warm | Lab lighting |
| Background | Phantom brain | Phantom brain | Lab background |
| Speed | Moderate | Moderate | Fast |
| Avg. IoU DPE | 0.76 | 0.79 | 0.77 |
| Avg. IoU particle filter | 0.80 | 0.81 | 0.79 |

## VI. FUTURE WORK

We would like to acknowledge that our methods in their present form likely cannot estimate the gripper's pose in in *vivo* settings, such as occlusion by grasped tissue. An important subject of future work is thus to adapt our methods to such a setting. In particular, this includes re-training and testing them on images with the presence of blood, occlusion and smoke, as these are very often present in surgical settings [26]. In addition, the cameras used in this work were low-cost endoscopic cameras. For future experiments, we aim to use medical-grade cameras instead. Finally, the tool's delivery platform in this work consisted of a straight rod, rendering the position and orientation of its base-link as fixed relative to the camera. To achieve adequate dexterity for neurosurgery however, the tool must be mounted to a delivery platform consisting of concentric tubes [27]. To control the tool in such a setup, its position and orientation relative to the camera also need to be estimated, increasing the pose estimation problem from 2-DoF to 8-DoF. In real-world surgical settings the trocar would also move freely, meaning that a method for transforming the estimated state from the camera's frame to the coil system's frame will be needed.

The training data required to adapt our proposed methods to these challenges will be substantially greater. Studies for other surgical tools have addressed this by using either improved sim2real transfer [28], [6] or self-supervised learning [9], and we hope to implement similar methods to gather more training data for the magnetic tool.

## ACKNOWLEDGMENT

Supported by Canadian Institutes of Health Research under Grant CPG-158271.

## REFERENCES


[1] C. Forbrigger, E. Fredin, and E. Diller, "Evaluating the feasibility of magnetic tools for the minimum dynamic requirements of microneurosurgery," *IEEE International Conference on Robotics and Automation*, pp. 4703–4709, 2023.

[2] A. Schonewille, C. He, C. Forbrigger, N. Wu, J. Drake, T. Looi, and E. Diller, "Electromagnets under the table: An unobtrusive magnetic navigation system for microsurgery," *IEEE Transactions on Medical Robotics and Bionics*, vol. 6, pp. 980 – 991, 2024.

[3] S. Speidel, M. Delles, C. Gutt, and R. Dillmann, "Tracking of instruments in minimally invasive surgery for surgical skill analysis," *Medical Imaging and Augmented Reality*, pp. 148–155, 2006.

[4] R. Sznitman, A. Basu, R. Richa, J. Handa, P. Gehlbach, R. H. Taylor, B. Jedynak, and G. D. Hager, "Unified detection and tracking in retinal microsurgery," in *Medical Image Computing and Computer-Assisted Intervention*, pp. 1–8, 2011.

[5] X. Du, T. Kurmann, P. L. Chang, M. Allan, S. Ourselin, R. Sznitman, J. D. Kelly, and D. Stoyanov, "Articulated multi-instrument 2-d pose estimation using fully convolutional networks," *IEEE Transactions on Medical Imaging*, vol. 37, pp. 1276–1287, 2018.

[6] M. Yoshimura, M. M. Marinho, K. Harada, and M. Mitsuishi, "Mba-pose: Mask and bounding-box aware pose estimation of surgical instruments with photorealistic domain randomization," in *IEEE International Conference on Intelligent Robots and Systems*, 2021.

[7] M. K. Hasan, L. Calvet, N. Rabbani, and A. Bartoli, "Detection, segmentation, and 3D pose estimation of surgical tools using convolutional neural networks and algebraic geometry," *Medical Image Analysis*, vol. 70, 2021.

[8] M. Allan, S. Ourselin, D. J. Hawkes, J. D. Kelly, and D. Stoyanov, "3-D pose estimation of articulated instruments in robotic minimally invasive surgery," *IEEE Transactions on Medical Imaging*, vol. 37, pp. 1204–1213, 2018.

[9] L. Sestini, B. Rosa, E. D. Momi, G. Ferrigno, and N. Padoy, "A kinematic bottleneck approach for pose regression of flexible surgical instruments directly from images," *IEEE Robotics and Automation Letters*, vol. 6, pp. 2938–2945, 2021.

[10] J. Lu, A. Jayakumari, F. Richter, Y. Li, and M. C. Yip, "Super deep: A surgical perception framework for robotic tissue manipulation using deep learning for feature extraction," in *IEEE International Conference on Robotics and Automation*, pp. 4783–4789, 2021.

[11] Y. Jiang, H. Zhou, and G. S. Fischer, "Markerless suture needle tracking from a robotic endoscope based on deep learning," in *International Symposium on Medical Robotics*, pp. 1–7, 2023.

[12] J. Wang, S. Tan, X. Zhen, S. Xu, F. Zheng, Z. He, and L. Shao, "Deep 3D human pose estimation: A review," *Computer Vision and Image Understanding*, vol. 210, 2021.

[13] B. Xiao, H. Wu, and Y. Wei, "Simple baselines for human pose estimation and tracking," in *European Conference on Computer Vision*, pp. 466–481, 2018.

[14] F. Richter, J. Lu, R. K. Orosco, and M. C. Yip, "Robotic tool tracking under partially visible kinematic chain: A unified approach," *IEEE Transactions on Robotics*, pp. 1653–1670, 2021.

[15] M. Ye, L. Zhang, S. Giannarou, and G.-Z. Yang, "Real-time 3D tracking of articulated tools for robotic surgery," in *Medical Image Computing and Computer-Assisted Intervention*, 2016.

[16] R. Hao, O. Özgüner, and M. C. Çavuşoğlu, "Vision-based surgical tool pose estimation for the da vinci robotic surgical system," in *IEEE International Conference on Intelligent Robots and Systems*, pp. 1298–1305, 2018.

[17] S. J. Julier and J. K. Uhlmann, "New extension of the kalman filter to nonlinear systems," in *Defense, Security, and Sensing*, pp. 182–193, 1997.

[18] J. Montero, M. Gherardini, F. Clemente, and C. Cipriani, "Comparison of online algorithms for the tracking of multiple magnetic targets in a myokinetic control interface," in *IEEE International Conference on Robotics and Automation*, pp. 2770–2776, 2020.

[19] M. Birsan, "Recursive bayesian method for magnetic dipole tracking with a tensor gradiometer," *IEEE Transactions on Magnetics*, vol. 47, pp. 409–415, 2011.

[20] D. Folio and A. Ferreira, "Modeling and estimation of self-phoretic magnetic janus microrobot with uncontrollable inputs," *IEEE Transactions on Control Systems Technology*, vol. 30, pp. 2681–2688, 2022.

[21] J. Martinez, R. Hossain, J. Romero, and J. J. Little, "A simple yet effective baseline for 3D human pose estimation," in *International Conference on Computer Vision*, pp. 2640–2649, 2017.

[22] Unity Technologies, "Unity Perception package." https://github.com/Unity-Technologies/com.unity.perception, 2020.

[23] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *International Conference on Intelligent Robots and Systems*, pp. 23–30, 2017.

[24] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[25] C. Forbrigger, *Tethered Magnetic Serial Robots for Minimally Invasive Neurosurgery*. PhD thesis, University of Toronto, Canada, 2023.

[26] Y. Wang, Q. Sun, Z. Liu, and L. Gu, "Visual detection and tracking algorithms for minimally invasive surgical instruments: A comprehensive review of the state-of-the-art," *Robotics and Autonomous Systems*, vol. 149, 2021.

[27] C. He, R. H. Nguyen, C. Forbrigger, J. Drake, T. Looi, and E. Diller, "A hybrid steerable robot with magnetic wrist for minimally invasive epilepsy surgery," in *IEEE International Conference on Robotics and Automation*, pp. 6830–6836, 2023.

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