2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

# Colibri5: Real-Time Monocular 5-DoF Trocar Pose Tracking for Robot-Assisted Vitreoretinal Surgery

Shervin Dehghani[1], Michael Sommersperger[1], Mahdi Saleh[1], Alireza Alikhani[1,2]
Benjamin Busam[1], Peter Gehlbach[3], Iulian Iordachita[4], Nassir Navab[1] and M. Ali Nasseri[1,2]

***Abstract*— Retinal surgery is a complex medical procedure that requires high precision dexterity to perform delicate instrument maneuvers with sub-millimeter accuracy. Minimizing the manual tremor and achieving precise and repeatable execution of surgical tasks has motivated the development of robotic platforms to overcome the limitations of manual surgery. However, specific tasks, such as instrument insertion through the trocar, are more challenging in robotic surgery than in conventional manual procedures since the robot control is often optimized for navigation inside the eye. This challenges the integration of robotic systems, creating a high cognitive load on the operator and prolonging the surgery time. Moreover, misalignment of the robot's remote center of motion (RCM) and trocar position during the procedure can lead to excessive forces between the instrument and the trocar, potentially causing patient trauma. Precise and rapid localization of the trocars enables the automation of the insertion procedure and dynamic compensation of eye motion.**

**In this work, we present a real-time marker-less method for 3D pose tracking of trocar, achieved with only a single monocular camera. Our experiments show promising results towards real-time trocar pose estimation and tracking, achieving an average error of $3^\circ$ in trocar orientation estimation, with an average processing time of 15 *fps*. This could serve as a foundation to improve robotic systems' automation, integration, and efficiency of robotic systems for retinal surgery. The dataset created for this work is made publicly available.**

***Index Terms*— Computer Vision for Medical Robotics; Vision-Based Navigation; Visual Tracking;**

## I. Introduction

Vitreoretinal surgery is one of the most challenging and delicate surgical procedures, which requires surgeons to possess significant proficiency and exceptional precision in handling microsurgical instruments. The need for experienced retinal surgeons is increasing, as more than 300 million individuals [1] suffer visual impairments due to various retinal diseases, such as retinal detachment, macular holes, diabetic retinopathy, and epiretinal membrane. These diseases are mainly treated by vitreoretinal surgery to preserve or restore vision. Transscleral ports, namely trocars, are placed during these procedures to provide instrument access to the ocular interior Fig. 1. The pivoting point, known as the Remote Center of Motion (RCM) [2], limits the transformations of the instrument. For a precise maneuver and minimizing the applied force on the sclera, a complete understanding of the RCM localization is essential.

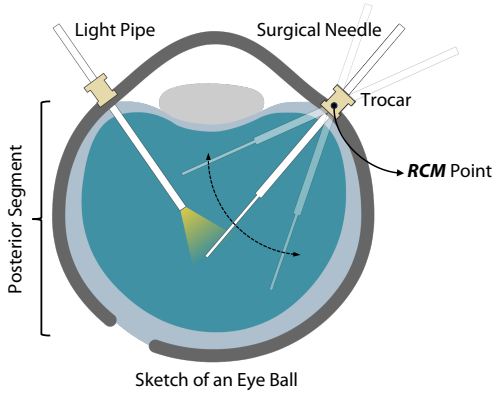


Fig. 1. An overview of vitreoretinal surgery. The light pipe and the surgical needle are introduced into the posterior segment via the trocars. Three trocars are placed in the eye: one for introducing the light pipe, one for the surgical instrument, and the third for docking an infusion line.

In conventional vitreoretinal interventions, surgeons rely on visual and haptic feedback to introduce instruments through the trocars. They approach by visually identifying the trocar and fine-tuning the insertion alignment by sensing the forces during the docking procedure.

In recent years, robotic platforms have made promising strides toward facilitating and improving vitreoretinal procedures that could enable more surgeons to perform such complicated tasks [3]–[6]. Despite the notable technical improvements, robotic systems are not yet fully integrated into surgical workflow due to the limited interfacing and integration capabilities. Specifically, in current robotic setups, clinicians need to invest significant effort in preparing the system and manually positioning the robot to be aligned with the trocar before the main procedure [7]. At the same time, due to the loss of the haptic feedback, the robot cannot compensate for the dynamic changes in the trocar localization, which can lead to excessive force on the sclera. Therefore, in minimally invasive robotic surgery, and particularly in vitreoretinal robotic surgery, the tasks of trocar localization, robot docking, instrument insertion, and continuous localization of the RCM position introduce additional cognitive demands on the surgeon and complicate the integration of robotic systems into the operating room.

Corresponding author: Shervin Dehghani (shervin.dehghani@tum.de)
[1] S. Dehghani, M. Sommersperger, M. Saleh, A. Alikhani, B. Busam, N. Navab and M. Ali Nasseri are with the Department of Computer Science, Technische Universität München, München 85748 Germany.
[2] A. Alikhani and M. Ali Nasseri are with Augenklinik und Poliklinik, Klinikum rechts der Isar der Technische Universität München, München 81675 Germany.
[3]P. Gehlbach is with Wilmer Eye Institute, Johns Hopkins Hospital, Baltimore, MD, USA.
[4]I. Iordachita is with the Laboratory for Computational Sensing and Robotics, Johns Hopkins University, Baltimore, MD, USA.

A significant reason for these challenges is the loss of haptic feedback and the design of the control system of these setups [8]. These must be designed for micron-scale procedures occurring in the limited working volume of the eye. Therefore, inserting the microsurgical instrument into the trocar using these control systems naturally becomes more challenging and time-consuming than the conventional insertion in manual surgery [9].

In this paper, we propose a general trocar 3D pose tracking method that could be used for both automatic trocar docking and dynamic alignment of the robot's RCM point with the insertion point of the trocar on the sclera. Due to the trocar's symmetrical geometry around its $Z-$Axis, its full 3D pose estimation problem is constrained to 5 degrees of freedom (5-DoF), consisting of three degrees for translation and two degrees for rotation. The method relies only on a monocular camera sensor with an appropriate lens, needless of any other modification of the surgical scene. The camera system can be mounted on the robot [9], on the instrument [10], [11], or be placed statically near the eye. The monocular sensor allows us to avoid introducing stereo cameras or other tracking systems to the scene, requiring an additional calibration stage on site or any further modification in the conventional devices. Notably, depth sensors cannot provide sufficient medical resolution yet [12], highlighting the expedience of the monocular systems.

Our proposed method localizes each trocar in the scene separately and estimates their 3D orientation from RGB images using the state-of-the-art monocular pose estimation methods. Our evaluations show the high precision of the method, and a high inference rate, which indicates the work's potential to be integrated into the surgical setups. To our knowledge, the method proposed in this paper is the first work to demonstrate a monocular 5-DoF trocar tracking from RGB images. We propose this system for vitreoretinal surgery. However, the methodology could also be transferred to other types of robot-assisted minimally invasive surgeries. Furthermore, we provide a public dataset for the monocular 3D pose estimation of the trocar.

## II. Related Works

### A. Trocar Docking and RCM Alignment in Robotic Systems

Multiple works have focused on estimating the trocar position and RCM alignment with different sensing systems. In [13], a force sensor attached to the robot's end-effector is utilized to estimate the instrument-sclera force and optimize the robot's kinematics to minimize this force. Some works utilize a geometric approach [14]–[16] or an external stereo-vision system [17]. Birch *et al.* [11] proposed the development of an instrument with two integrated miniature cameras to detect the trocar position and estimate the robot's optimal RCM point. Later, a monocular vision-based system for 4-DoF detection of trocar was introduced in [9]. A U-Net-like model [18] detects the trocar entry point, and a crop around the detected point is provided to a regression model to estimate the trocar rotation. The loss functions of the second model handle the trocar symmetry-in-variance. This system does not estimate the absolute position of the trocar in the scene, but instead, its rotation in $\mathrm{SO}(3)$ and the homogeneous translation w.r.t. the camera; therefore, the robot can only be instructed to move along the correct trajectory purpose of docking, and the operator should stop the movement when the desired position is reached. For the same reason, this method cannot update the robot RCM point intraoperatively according to trocar movements.

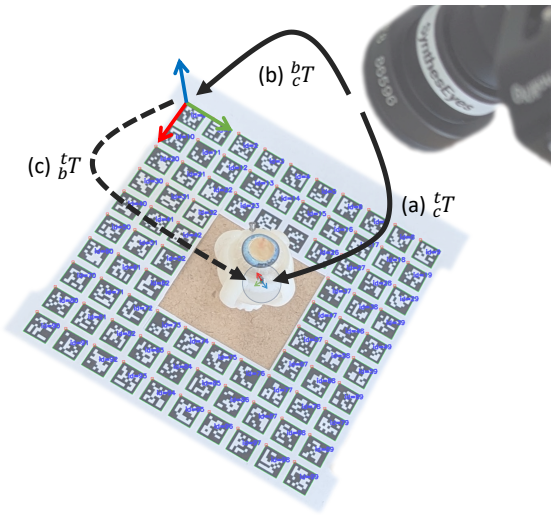


Fig. 2. An overview of the dataset creation. The trocar is placed in a fixed position w.r.t. the Aruco board. The initial ${}^{t}_{c}T$ of a selected frame is calculated with *PnP*; therefore the ${}^{t}_{b}T$ is calculated, which is static among all the captured frames. Using ${}^{t}_{b}T$ and ${}^{c}_{b}T$ in the other frames, ${}^{t}_{c}T$ is calculated.

In another approach proposed in [10], trocars are tracked with 3-DoF with a high accuracy. This work does not estimate trocar rotation, but only their spatial translation w.r.t. the camera, by tracking multiple fixed Aruco markers. This makes the translation of the work into the surgical scene more complex. Moreover, the Aruco markers' relative pose to the trocar(s) are assumed to be static. The trocars can be moving for several reasons, and therefore, mounting the markers robustly towards all of these movements makes the task even more challenging, emphasizing on the importance of implicitly learning the trocar pose, independent of any other proxies in the scene.

### B. Monocular 6-DoF Pose Estimation

Estimating object translation and rotation in space requires predicting six degrees of freedom (6-DoF). Traditionally, this task is solved with markers for medical setups [19] where objects pose photometric challenges. However, space and hygiene restrictions do not always allow object attachments in sterile environments. Recent markerless methods use the object's appearance to learn a variety of pre-defined [20], [21] or neural [22], [23] representations to estimate the 6-DoF. Depth sensors can help to improve results [24] or overcome missing annotations [25]. Medical objects pose challenges to these methods due to their reflectivity and limited texture. While these can be overcome with image modalities such as polarization [26], [27] (RGBP), they

are not always easily available. A typical monocular RGB pose pipeline consists of two stages: 2D-3D correspondences are established and matched using a consecutive process, typically *PnP* [28]. GDR-Net [29] proposed to regress 6D poses end-to-end, leveraging correspondence-based intermediate geometric features to overcome this two-stage nature. These methods typically fall behind their counterparts that leverage additional depth sensing. However, the coarse-to-fine descriptor design presented by ZebraPose [30] leads to significantly improved results, surpassing previous RGBD methods. While these methods work well even in case of severe occlusions, they are challenged by object symmetries and visual ambiguities [31], and our object of interest, namely the trocar, shows a rotational symmetry. The seminal work SurfEmb [32] uses a contrastive loss to build a descriptor space agnostic to symmetries and multi-modal surface distributions for 2D-3D dense correspondence matching. While this allowed to deal with visual ambiguities, it comes with its own computational complexity, which was elegantly addressed by SC6D [33], which we take as inspiration for our pipeline design.

Among the mentioned works, no general monocular and marker-less trocar pose estimation exists. They are either designed to adjust the robot's RCM point or lack the estimation of some degrees of the 6D pose. This work introduces a general 5-DoF trocar pose estimation with a monocular camera, which could be used for both purposes of trocar docking [9] and interoperative RCM point adjustment [10].

## III. Method

In this section, we describe the 5-DoF estimation of the trocar in four steps: (a) *Dataset Creation*, (b) *Object Detection*, (c) *Pose Estimation*, and (d) *Object Tracking*.

### A. Dataset Creation

An accurate dataset is a critical step for the fully supervised pose estimation. As an initial step of this work, we have created a comprehensive dataset of 5-DoF trocar poses, including trocars in two situations: **(a)** without surgical tool and **(b)** with a surgical tool being mounted. A perfectly mounted surgical tool would be aligned with the trocar pose, and if the model is overfitted to this data, it might learn the tool's orientation instead of the trocar's, and the tool and the trocar are not necessarily aligned during the inference. Therefore, it is essential to have a balanced dataset of both splits. One trocar is mounted on a phantom eye model, and the phantom eye is placed on an Aruco [34] board. We keep the relative pose of the trocar w.r.t. the board static, and images are taken from the camera from different poses. Among all the taken images, we pick one frame, $I$, and run a *PnP* algorithm to estimate the camera to trocar pose (${}^{t}_{c}T_{I}$), as shown in Fig. 3. The pose of the board w.r.t. the camera in $I$ is also estimated by detecting the Aruco markers (${}^{b}_{c}T_{I}$). Having these two transformations, the pose of the trocar w.r.t. the board is calculated as:

$$ {}^{t}_{b}T = {}^{b}_{c}T^{-1}_{I} \cdot {}^{t}_{c}T_{I}. \tag{1} $$

For each of the other frames, as $J$, the ${}^{t}_{c}T_{J}$ is calculated as:

$$ {}^{t}_{c}T_{J} = {}^{b}_{c}T_{J} \cdot {}^{t}_{b}T. \tag{2} $$

In summary, we use the Alg. 1 to create the ground truth for the dataset, generating $R \in \mathrm{SO}(3)$, the 3D translation vector $t$, segmentation mask, and the bounding box of the trocar for each image:

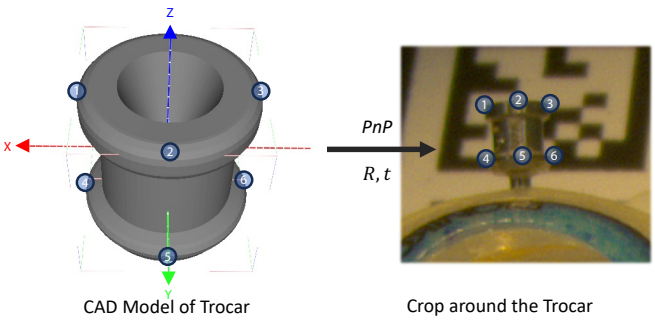


Fig. 3. The correspondence points between the trocar 3D model and the first frame of the dataset. The user chooses the points on the image through the user interface. ${}^{t}_{c}T$ is calculated by correspondence between these 6 points and is tailored to the calculation of ${}^{c}_{b}T$

**Algorithm 1** Real Dataset Creation Algorithm

**Require:** Set of images, Img
**Require:** Trocar CAD model $C$
**Require:** image $I \in$ Img, with 2D-3D correspondence points between the trocar image and the 3D model

$P_2, P_3 \leftarrow \text{GetPointCorrespondances}(I, C)$
${}^{t}_{c}T \leftarrow \text{SolvePnP}(P_2, P_3)$
${}^{b}_{c}T_{I} \leftarrow \text{DetectArucoBoard}(I)$
${}^{t}_{b}T = {}^{b}_{c}T^{-1}_{I} \cdot {}^{t}_{c}T_{I}$
**for** each image $J$ in $Img$ **do**
  ${}^{b}_{c}T_{J} \leftarrow \text{DetectArucoBoard}(J)$
  ${}^{t}_{c}T_{J} = {}^{b}_{c}T_{J} \cdot {}^{t}_{b}T$
  $\text{render}_{J} \leftarrow \text{RenderTrocarWithPose}({}^{t}_{c}T_{J}, C)$
  $\text{segmentation}_{J} \leftarrow \text{GetSegmentation}(J, \text{render}_{J})$
  $\text{boundingBox}_{J} \leftarrow \text{GetBoundingBox}(\text{segmentation}_{J})$
**end for**=0

Having the pose of the trocar and its CAD model, we render the trocar object on the image, which is sufficient to generate the segmentation mask and the bounding box. The dataset's quality is highly dependent on the initial $PnP$ result, so a frame with low ambiguity should be chosen with careful annotation for that purpose.

For the purpose of pretraining of the methods in Sec.III-C, as proposed in [9], a set of synthetic datasets is also created, rendering the trocar with random poses on random VOC ( [35]) backgrounds.

### B. Object Detection

Most of the pose estimation works discussed in Sec. II, require a crop of the original image containing the target object as the input. In this regard, the first step towards

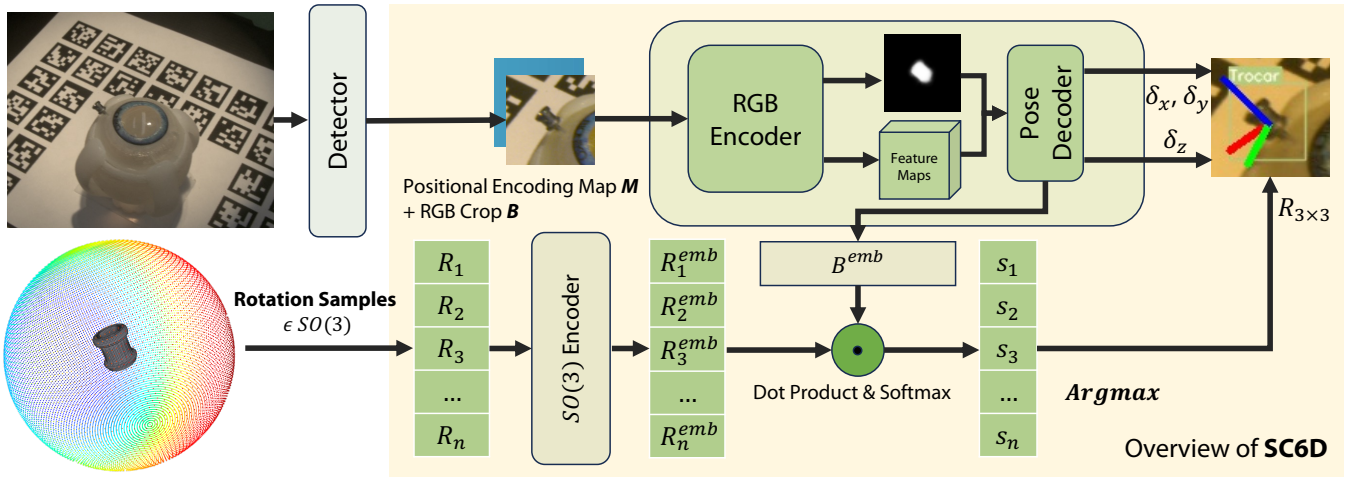


Fig. 4. Overview of the approach. The RGB image is fed into the **Detector** (Yolo7), and the crop around each detected trocar with its positional encoding is passed into the **SC6D** network. A feature map and the input segmentation mask are learned and decoded into 1) rotation embedding $B$, 2) $[\delta x, \delta y]$, and $\delta z$. On the other hand, a fixed set of samples from $\mathrm{SO}(3)$ are embedded in a feature space like $B$, where the closest sample point embedding to $B$ is picked as the best rotation estimation of the input. $\delta x$, $\delta y$, and $\delta z$, together with the position of the crop in the image, are used to estimate the translation vector.

the pose estimation of the trocars in the scene is an object detection model. In this work, we trained a Yolo7 [36] model on our dataset. Any other conventional method is also suitable, and we chose Yolo7 due to its high inference rate. When the trocars in the image are localized, a crop around each of them is created and passed to the pose estimation module, which is further discussed in the next section.

### C. Pose Estimation

Our approach is inspired by the method proposed in [33]. In our case, due to the symmetry of the object around its Z-axis (Fig. 3), the problem is reduced to a 5-DoF pose estimation, so all the possible rotations that have the same $r_z$ are considered as a correct estimation. SC6D learns the rotation $\in \mathrm{SO}(3)$, and a ray that connects the camera's focal point to the center of the object separately, namely $r_c^o$. First, a uniform sampling of the $\mathrm{SO}(3)$, $\mathcal{R}$, is created and fed into a network to create an embedding space of these sample points (Fig. 5). The RGB crop and its positional encoding are concatenated and passed into another encoder-decoder network to create the input's feature maps and segmentation mask. These outputs are also concatenated and converted into a rotation encoding through their pass into the **pose decoder**. This embedding is compared to the embeddings of the points in $\mathcal{R}$ and the closest one is chosen as the rotation of the object. At the same time, the concatenation of the segmentation and the feature maps are also being used to regress to the length of the the ray $r_c^o$ and the object's center $\delta x, \delta y$ from the center of the RGB crop, which together with the length of $r_c^o$ are predicting the objects translation w.r.t the camera. The overview of the approach is shown in Fig. 4.

### D. Object Tracking

Due to the trocar's symmetrical geometry, the network discussed in Sec. III-C is not constrained to estimating one specific pose. Instead, it can estimate multiple equivalent transformations involving rotations around the trocar's $Z$-axis. Since each frame is analyzed separately, these predictions may still vary slightly from one another, even though they are all valid estimations. This prediction variability creates a visually noticeable jittering effect and introduces avoidable noise into the system. To mitigate this issue, a smoothing algorithm is required, as classical methods like the Kalman Filter [37] and its variations or the approach proposed by [38], which jointly addresses rotation and translation. However, to achieve a more robust result prior to the post-processing stage, we modify the network to include the previous frame's rotation estimation online. Specifically, at step $i+1$, where the input embedding is matched to the $\mathrm{SO}(3)$ samples, $\mathcal{R}$ instead of selecting the closest sample point from the entire set of rotation samples, we employ a $k$-Nearest Neighbors ($kNN$) algorithm [39] on the rotation estimate from the previous frame, $r_i$. The $kNN$ algorithm identifies a subset of $\mathrm{SO}(3)$ samples, $\mathcal{R}'_{i+1} \subseteq \mathcal{R}$, and the current embedding is only compared to the embeddings of rotations in this subset (refer to Fig. 5). This approach significantly helps the jittering problem, and it is reasonable

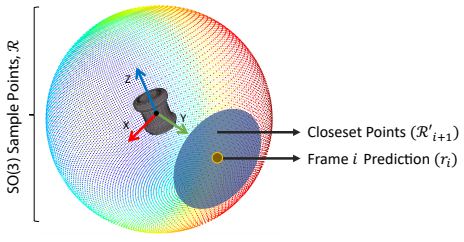


Fig. 5. For a smoother rotation estimation prior to post-processing in $\mathrm{SO}(3)$, for frame $i+1$, instead of choosing the closest point on the sample points to the image embedding, we keep track of the chosen point from the previous frame, $r_i$. For frame $i+1$, instead of searching among all the sample points in $\mathcal{R}$, we search among the $k$ nearest neighbors of $r_i$, denoted as $\mathcal{R}'_i$.

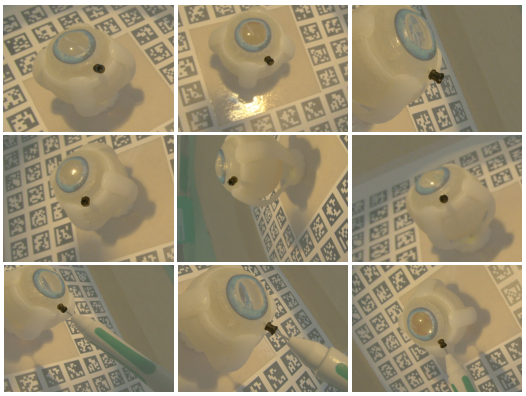

Fig. 6. Qualitative results of the estimations on the test set. The trocar object is rendered with Pyrender according to the estimated pose and overlayed on the original image.

since the rotation between consecutive frames does not change drastically. By adjusting the value of $k$ in the $kNN$ algorithm, we can control the level of smoothing applied to the estimations.

A Kalman Filter is applied to the results of the object detector, and they are tracked and distinguished from each other by Sort algorithm [40].

## IV. Materials

For the creation of the dataset, a BIONIKO eye model with flex orbit holder was placed on an Aruco board and a microsurgical trocar (23G from RETILOCK) was inserted into the phantom. For the acquisition of the dataset images and inference, a Basler acA4024-29$\mu$ sensor (Basler AG, Ahrensburg, Germany) with an Edmund Optic (Edmund Optics Inc. NJ, USA) lens (6mm, f8) was used. The frames are acquired at a 20 *fps* rate with 4K resolution. The camera was calibrated separately for each scene by a checkerboard calibration method using MATLAB (R2022b), which provided a reprojection error of $0.3$ pixels. 800 images are captured for an instance with the surgical tool inserted into the trocar and another 800 set without the surgical tool. All the images are resized to $1280 \times 960$ pixels. The data is split with a $4:1$ ratio for training and testing purposes. The segmentation masks are generated by Pyrender [1], and since Pyrender does not support camera distortion, all the images are undistorted after being acquired. Since the pose estimation requires trocar crops as input, including multiple trocars in the training set does not enhance the performance of the pose estimation network. However, including some samples with multiple trocars for the object detection model is crucial. We generated specific data solely for fine-tuning the object detection model to effectively identify multiple trocars, but in this dataset, the trocar poses were not generated. The dataset and the helper functions are available on our public repository [2]. The Yolo7 [3] model is trained on the original size and the crops are resized to $256 \times 256$ pixels for the training of the pose estimation network [4]. Models are written with PyTorch version 1.12.0 and trained on Nvidia RTX A5000 GPU.

## V. Experiments and Results

To calculate the pose estimation error, for each frame of the test set ${}^{t}_{c}T_{gt}$ is provided from the dataset, and ${}^{t}_{c}T_{prediction}$ is calculated. From these two poses, we define $T_{error}$ as ${}^{t}_{c}T_{gt} \cdot {}^{t}_{c}T^{-1}_{prediction}$, which is the relative pose between the ground truth pose and the prediction pose. For a zero error, $T_{error}$ should be equal to $I_4$. For the rotation error, due to the object symmetry, we only measure the $r_z$ from $T_{error}$, with the following equation:

$$R_{\text{error}} = \arccos(r_z \times \vec{k}). \tag{3}$$

The results of the rotation error are plotted in Fig. 7. The results show an average of $3.06 \pm 2.36$ degrees error. This is a huge improvement over the only similar work, [9], where the authors claim an $80\%$ accuracy for estimating the error $< 10^\circ$ and $94\%$ for $< 15^\circ$. These numbers are improved to $97\%$ and $99\%$. Moreover, on $89\%$ of the test set, an accuracy of $< 5^\circ$ is reached. For the translation error, on the other hand, [10] have evaluated their method extensively. They have reported an impressive RMSE of the trocar localization of $1.24$mm. This has been achieved by localization of the Aruco markers in the scene. In our case, we have analyzed the translation error of each axis separately (Fig. 8), while they are also calculated differently due to the methodology. The absolute error for each of the $X$, $Y$ and $Z$ axes are $4.59 \pm 3.41$ mm, $3.23 \pm 1.83$ mm and $1.52 \pm 1.21$ mm respectively. The performance on the $Z$-axis is considerably better than the other two since it is being learned implicitly. Some samples of the rendered trocar with the predicted pose on the test set are illustrated in Fig. 6.

Such methods should also have high-speed competence to be integrated in the surgical scenes. We measured the end-to-end inference time on 200 samples containing 0, 1, or 2 trocars (Fig. 5). On average, the method provides 15 *fps* performance on the A5000 GPU. The performance can be improved further with the optimization techniques proposed in [41], [42].

## VI. Discussion and Future Work

The quality of the dataset affects the results, and creating a more accurate dataset, as proposed in [12], would eventually lead into a more accurate system. However, in this study, the results are only comparing the predictions to the dataset ground truth, and both are being evaluated from the same domain, which causes a level of uncertainty in the results. For this reason, a study like [10] is needed to evaluate the effectiveness of the system by an external sensing source, such as the force sensors and external tracking devices, rather than the same modalities being used for the training. The method is capable of estimating the pose of the trocar while

[1]https://github.com/mmatl/pyrender

[2]https://github.com/shervn/5dof-trocar-pose-dataset

[3]https://github.com/WongKinYiu/yolov7

[4]https://github.com/dingdingcai/SC6D-pose

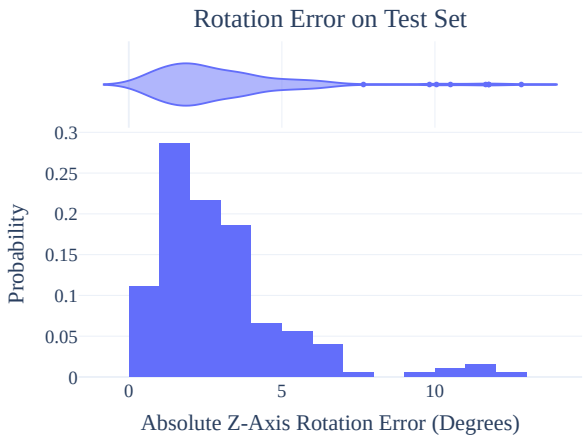


Fig. 7. Z Rotation Error Plot. The data is from 199 samples and only one prediction result is removed due to a wrong bounding box detection.

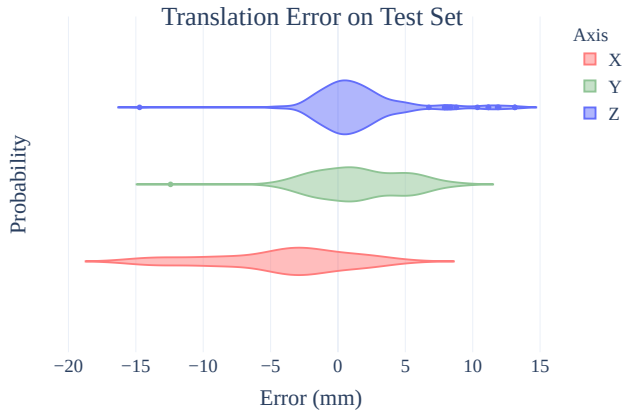


Fig. 8. Translation Error Plot. The data is from 199 samples and only one prediction result is removed due to a wrong bounding box detection. For a better visualization, on this plot the raw values are plotted instead of the absolute error values which are discussed in Sec. V

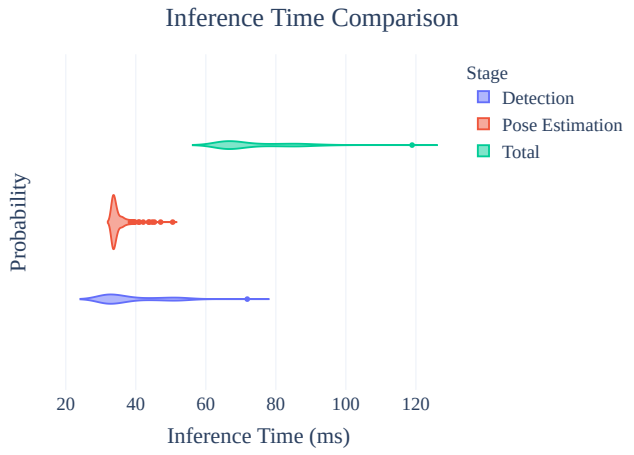


Fig. 9. Comparison of the inference time during the different stages of the end-to-end algorithm. The performance can be improved by 20 30% by optimizing the networks and implementations for inference.

it is being manipulated with the surgical tool, as shown in the supplementary video, but for the evaluation of such accuracy, the Aruco board is not relevant since the pose of the trocar w.r.t. the board is not static anymore. The external sensor is also required for the evaluation of these scenarios.

As shown in [9], when the camera is mounted on the robot, with a hand-eye calibration [43], the robot end-effector could also be localized, and the control of the robot is then automatable. Therefore, future work integrates the method into the robotic setup [3] and evaluates the precision of the system's precision utilizing external sensing tools, as force sensors.

The proposed approach mainly focuses on robotic surgery and the optimization of the robot's kinematics. The same method applies to evaluating surgeon performance while performing a manual vitreoretinal surgery by measuring the interactions between the hand-held instrument and the trocars. The eye sclera, a texture-less object, could also be tracked by tracking the 5-DoF pose of multiple trocars attached to the sclera for further evaluations and procedures.

Moreover, the primary emphasis in vitreoretinal surgery lies in the surgeon's concentration on the main target area, such as the retina, whether it is performed manually or with robotic assistance. Consequently, it is conceivable that the interactions between the trocar and the instruments may receive less attention as the surgeon's focus is directed elsewhere. Having the trocar pose estimated, along with the tools and trocars interaction, the sonification of such data can increase the surgeon's awareness without distracting them from the main task, as shown in [44]. Sonification in the field of ophthalmology has been evaluated before, [45], [46], which implies the high chance of acceptance of such an approach among the medical community.

## VII. Conclusion

This study introduced a novel monocular 5-DoF multi-trocar tracking system, eliminating the need for markers, depth or stereo cameras, and other complex tracking systems. We also created a trocar pose dataset, publicly available for the research community. Our findings demonstrate a notable level of precision, surpassing comparable research efforts. Our results indicate that translation estimation remains a primary challenge, whereas rotation estimation exhibits remarkable reliability. Furthermore, our method shows a high inference speed, allowing it to be integrated into real-time applications. Future research endeavors will attempt to refine the translation accuracy and optimize clinical suitability with extended evaluations utilizing external sensors.

## VIII. Acknowledgment

The authors wish to thank SynthesEyes[1] for providing the experimental setup for the dataset creation.

[1]https://syntheseyes.de

## REFERENCES


[1] I. F. de Oliveira, E. J. Barbosa, M. C. C. Peters, M. A. B. Henostroza, M. N. Yukuyama, E. dos Santos Neto, R. Löbenberg, and N. Bou-Chacra, "Cutting-edge advances in therapy for the posterior segment of the eye: Solid lipid nanoparticles and nanostructured lipid carriers," *International Journal of Pharmaceutics*, p. 119831, 2020.

[2] R. H. Taylor, J. Funda, D. D. Grossman, J. P. Karidis, and D. A. LaRose, "Remote center-of-motion robot for surgery," Mar. 14 1995, uS Patent 5,397,323.

[3] M. A. Nasseri, M. Eder, S. Nair, E. C. Dean, M. Maier, D. Zapp, C. P. Lohmann, and A. Knoll, "The introduction of a new robot for assistance in ophthalmic surgery," in *Eng. Med. Biol. Soc. (EMBC), 2013 35th Annu. Int. Conf. IEEE*. IEEE, 2013, pp. 5682–5685.

[4] G.-Z. Yang, J. Cambias, K. Cleary, E. Daimler, J. Drake, P. E. Dupont, N. Hata, P. Kazanzides, S. Martel, R. V. Patel *et al.*, "Medical robotics-regulatory, ethical, and legal considerations for increasing levels of autonomy," *Sci. Robot*, vol. 2, no. 4, p. 8638, 2017.

[5] B. Lu, H. K. Chu, K. Huang, and L. Cheng, "Vision-based surgical suture looping through trajectory planning for wound suturing," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 2, pp. 542–556, 2018.

[6] E. Vander Poorten, C. N. Riviere, J. J. Abbott, C. Bergeles, M. A. Nasseri, J. U. Kang, R. Sznitman, K. Faridpooya, and I. Iordachita, "Robotic retinal surgery," in *Handbook of Robotic and Image-Guided Surgery*. Elsevier, 2020, pp. 627–672.

[7] K. Faridpooya, S. H. van Romunde, S. S. Manning, J. C. van Meurs, G. J. Naus, M. J. Beelen, T. C. Meenink, J. Smit, and M. D. de Smet, "Randomised controlled trial on robot-assisted versus manual surgery for pucker peeling," *Clinical & experimental ophthalmology*, vol. 50, no. 9, pp. 1057–1064, 2022.

[8] M. Zhou, Q. Yu, K. Huang, S. Mahov, A. Eslami, M. Maier, C. P. Lohmann, N. Navab, D. Zapp, A. Knoll, and M. Nasseri, "Towards robotic-assisted subretinal injection: A hybrid parallel–serial robot system design and preliminary evaluation," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 8, pp. 6617–6628, 2020.

[9] S. Dehghani, M. Sommersperger, J. Yang, M. Salehi, B. Busam, K. Huang, P. Gehlbach, I. Iordachita, N. Navab, and M. A. Nasseri, "Colibridoc: An eye-in-hand autonomous trocar docking system," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 7717–7723.

[10] J. Birch, L. Da Cruz, K. Rhode, and C. Bergeles, "Trocar localisation for robot-assisted vitreoretinal surgery," *International Journal of Computer Assisted Radiology and Surgery*, pp. 1–8, 2023.

[11] J. Birch, King's College London, K. Rhode, King's College London, C. Bergeles, King's College London, L. Da Cruz, and Moorfields Eye Hospital, "Towards localisation of remote centre of motion and trocar in vitreoretinal surgery," pp. 33–34. [Online]. Available: https://www.ukras.org/publications/ras-proceedings/UKRAS21/pp33-34

[12] P. Wang, H. Jung, Y. Li, S. Shen, R. P. Srikanth, L. Garattoni, S. Meier, N. Navab, and B. Busam, "Phocal: A multi-modal dataset for category-level object pose estimation with photometrically challenging objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 21 222–21 231.

[13] A. Ebrahimi, N. Patel, C. He, P. Gehlbach, M. Kobilarov, and I. Iordachita, "Adaptive control of sclera force and insertion depth for safe robot-assisted retinal surgery," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 9073–9079.

[14] J. Smits, D. Reynaerts, and E. V. Poorten, "Setup and method for remote center of motion positioning guidance during robot-assisted surgery," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 1315–1322.

[15] C. Gruijthuijsen, L. Dong, G. Morel, and E. V. Poorten, "Leveraging the fulcrum point in robotic minimally invasive surgery," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2071–2078, 2018.

[16] L. Dong and G. Morel, "Robust trocar detection and localization during robot-assisted endoscopic surgery," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 4109–4114.

[17] B. Rosa, C. Gruijthuijsen, B. Van Cleynenbreugel, J. V. Sloten, D. Reynaerts, and E. V. Poorten, "Estimation of optimal pivot point for remote center of motion alignment in surgery," *International Journal of Computer Assisted Radiology and Surgery*, vol. 10, no. 2, pp. 205–215, 2015. [Online]. Available: https://doi.org/10.1007/s11548-014-1071-3

[18] 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: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 2015, pp. 234–241.

[19] M. Esposito, B. Busam, C. Hennersperger, J. Rackerseder, A. Lu, N. Navab, and B. Frisch, "Cooperative robotic gamma imaging: Enhancing us-guided needle biopsy," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part II 18*. Springer, 2015, pp. 611–618.

[20] B. Busam, H. J. Jung, and N. Navab, "I like to move it: 6d pose estimation as an action decision process," *arXiv preprint arXiv:2009.12678*, 2020.

[21] F. Li, I. Shugurov, B. Busam, M. Li, S. Yang, and S. Ilic, "Polarmesh: A star-convex 3d shape approximation for object pose estimation," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4416–4423, 2022.

[22] F. Li, H. Yu, I. Shugurov, B. Busam, S. Yang, and S. Ilic, "Nerf-pose: A first-reconstruct-then-regress approach for weakly-supervised 6d object pose estimation," *arXiv preprint arXiv:2203.04802*, 2022.

[23] H. Chen, F. Manhardt, N. Navab, and B. Busam, "Texpose: Neural texture learning for self-supervised 6d object pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 4841–4852.

[24] Y. He, H. Huang, H. Fan, Q. Chen, and J. Sun, "Ffb6d: A full flow bidirectional fusion network for 6d pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3003–3013.

[25] F. Manhardt, G. Wang, B. Busam, M. Nickel, S. Meier, L. Minciullo, X. Ji, and N. Navab, "Cps++: Improving class-level 6d pose and shape estimation from monocular images with self-supervised learning," *arXiv preprint arXiv:2003.05848*, 2020.

[26] D. Gao, Y. Li, P. Ruhkamp, I. Skobleva, M. Wysocki, H. Jung, P. Wang, A. Guridi, and B. Busam, "Polarimetric pose prediction," in *European Conference on Computer Vision*. Springer, 2022, pp. 735–752.

[27] P. Wang, L. Garattoni, S. Meier, N. Navab, and B. Busam, "Crocps: Addressing photometric challenges in self-supervised category-level 6d object poses with cross-modal learning," 2022.

[28] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Ep n p: An accurate o (n) solution to the p n p problem," *International journal of computer vision*, vol. 81, pp. 155–166, 2009.

[29] G. Wang, F. Manhardt, F. Tombari, and X. Ji, "Gdr-net: Geometry-guided direct regression network for monocular 6d object pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 16 611–16 621.

[30] Y. Su, M. Saleh, T. Fetzer, J. Rambach, N. Navab, B. Busam, D. Stricker, and F. Tombari, "Zebrapose: Coarse to fine surface encoding for 6dof object pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 6738–6748.

[31] F. Manhardt, D. M. Arroyo, C. Rupprecht, B. Busam, T. Birdal, N. Navab, and F. Tombari, "Explaining the ambiguity of object detection and 6d pose from visual data," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6841–6850.

[32] R. L. Haugaard and A. G. Buch, "Surfemb: Dense and continuous correspondence distributions for object pose estimation with learnt surface embeddings," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 6749–6758.

[33] D. Cai, J. Heikkilä, and E. Rahtu, "Sc6d: Symmetry-agnostic and correspondence-free 6d object pose estimation," in *2022 International Conference on 3D Vision (3DV)*. IEEE, 2022, pp. 536–546.

[34] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.

[35] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International journal of computer vision*, vol. 111, pp. 98–136, 2015.

[36] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," *arXiv preprint arXiv:2207.02696*, 2022.

[37] R. E. Kalman, "A new approach to linear filtering and prediction problems," 1960.
[38] B. Busam, T. Birdal, and N. Navab, "Camera pose filtering with local regression geodesics on the riemannian manifold of dual quaternions," in *Proceedings of the IEEE international conference on computer vision workshops*, 2017, pp. 2436–2445.
[39] R. O. Duda, P. E. Hart *et al.*, *Pattern classification and scene analysis*. Wiley New York, 1973, vol. 3.
[40] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 3464–3468.
[41] M. Sommersperger, J. Weiss, M. A. Nasseri, P. Gehlbach, I. Iordachita, and N. Navab, "Real-time tool to layer distance estimation for robotic subretinal injection using intraoperative 4d oct," *Biomedical Optics Express*, vol. 12, no. 2, pp. 1085–1104, 2021.
[42] J. Weiss, M. Sommersperger, A. Nasseri, A. Eslami, U. Eck, and N. Navab, "Processing-aware real-time rendering for optimized tissue visualization in intraoperative 4d oct," in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part V 23*. Springer, 2020, pp. 267–276.
[43] R. Horaud and F. Dornaika, "Hand-eye calibration," *The international journal of robotics research*, vol. 14, no. 3, pp. 195–210, 1995.
[44] S. Matinfar, M. Salehi, D. Suter, M. Seibold, S. Dehghani, N. Navab, F. Wanivenhaus, P. Fürnstahl, M. Farshad, and N. Navab, "Sonification as a reliable alternative to conventional visual surgical navigation," *Scientific Reports*, vol. 13, no. 1, p. 5930, 2023.
[45] S. Matinfar, M. A. Nasseri, U. Eck, M. Kowalsky, H. Roodaki, N. Navab, C. P. Lohmann, M. Maier, and N. Navab, "Surgical soundtracks: Automatic acoustic augmentation of surgical procedures," *International journal of computer assisted radiology and surgery*, vol. 13, pp. 1345–1355, 2018.
[46] H. Roodaki, N. Navab, A. Eslami, C. Stapleton, and N. Navab, "Sonifeye: Sonification of visual information using physical modeling sound synthesis," *IEEE transactions on Visualization and Computer Graphics*, vol. 23, no. 11, pp. 2366–2371, 2017.