Autonomously Navigating a Surgical Tool Inside the Eye by Learning from Demonstration

Ji Woong Kim[1], **Changyan He**[1], **Muller Urias**[3], **Peter Gehlbach**[3], **Gregory D. Hager**[2], **Iulian Iordachita**[1], **Marin Kobilarov**[1]

[1]Laboratory for Computing + Sensing (LCSR) dept. at the Johns Hopkins University, Baltimore, MD 21218 USA

[2]Computer Science dept. at the Johns Hopkins University, Baltimore, MD 21218 USA

[3]Wilmer Eye Institute at the Johns Hopkins Hospital, Baltimore, MD 21287 USA

Abstract

A fundamental challenge in retinal surgery is safely navigating a surgical tool to a desired goal position on the retinal surface while avoiding damage to surrounding tissues, a procedure that typically requires tens-of-microns accuracy. In practice, the surgeon relies on depth-estimation skills to localize the tool-tip with respect to the retina in order to perform the tool-navigation task, which can be prone to human error. To alleviate such uncertainty, prior work has introduced ways to assist the surgeon by estimating the tool-tip distance to the retina and providing haptic or auditory feedback. However, automating the tool-navigation task itself remains unsolved and largely unexplored. Such a capability, if reliably automated, could serve as a building block to streamline complex procedures and reduce the chance for tissue damage. Towards this end, we propose to automate the tool-navigation task by learning to mimic expert demonstrations of the task. Specifically, a deep network is trained to imitate expert trajectories toward various locations on the retina based on recorded visual servoing to a given goal specified by the user. The proposed autonomous navigation system is evaluated in simulation and in physical experiments using a silicone eye phantom. We show that the network can reliably navigate a needle surgical tool to various desired locations within 137 μm accuracy in physical experiments and 94 μm in simulation on average, and generalizes well to unseen situations such as in the presence of auxiliary surgical tools, variable eye backgrounds, and brightness conditions.

I. Introduction

Retinal surgery is among the most challenging microsurgical endeavors due to its micron scale precision requirements, constrained work-space, and the delicate non-regenerative tissue of the retina. During the surgery, one of the most challenging tasks is the spatial estimation of the surgical tool location with respect to the retina in order to precisely move its tip to a desired location on the retina. For example, when performing retinal-peeling or vein cannulation, the surgeon must rely on intuitive depth-estimation skills to navigate toward a targeted location on the retina, while ensuring that the tool-tip contacts the retina

murias1,pgelbach@jhmi.edu .

precisely at the desired location. Such maneuvers introduce high risk because the surgical tools are sharp and the slightest misjudgement can damage the surrounding tissues, which could lead to serious complications.

To alleviate the difficulty of the tool-navigation task, we propose to automate it by learning the closed-loop visual servoing process employed by surgeons, i.e. mapping from visual input (video) to euclidean position control commands to actuate the robot. Specifically, we train a deep network to imitate expert trajectories toward various locations on the retina based on many demonstrations of the tool-navigation task. The input to the network are the monocular top-down view of the surgery through a microscope and user-input defining the 2D goal location to be reached. The advantage of this method is that the user only specifies the goal in 2D, e.g. as simple as clicking the desired location using a mouse (Fig. 2), and the network outputs an incremental 3D waypoint toward the target location on the retina. Since estimating depth is the challenging task for humans, the network takes the burden of extrapolating how to navigate along the depth dimension based on its training experience. Learning such simple tool-navigation maneuver is fundamental in automating surgery, since it is the primitive action performed in any surgical procedures.

We note that our approach is grounded in the hypothesis that the tool-navigation task may be automated primarily using vision. In fact, surgeons rely on their vision to localize objects and estimate their spatial relationship to navigate the surgical tool. Furthermore, the surgical scene captures a distinct tool-shadow dynamics which can be useful for recognizing proximity between the tool-tip and the retina. Specifically, the tool and its shadow converge upon approaching the retina (Fig. 1), which can be as cues to train the network. In addition, while a complete setup can include stereo vision, in this work we rely on a single camera alone for simplicity. We also utilize a force-sensing modality to detect contact with retina, such that the surgical tool can be stopped upon contact.

The system performance is validated experimentally using both an artificial eye-phantom as well as in simulation employing the Unity3D (Unity Technologies) environment [1]. The main objective is to assess the quality of surgical tool navigation to desired locations on the retina. To achieve this, we employ a batch of benchmark tasks where various positions on the retina are targeted in a grid-like fashion (Fig. 6). For simplicity, we keep the eye position and tool-orientation fixed during the experiments. While this is not a realistic assumption in practice, since the eye could involuntarily move during procedures, our approach can easily extend to the more general setting of different eye rotations and tool orientations through additional training. To test the robustness of our network, we also perform the benchmark task in the presence of unseen distractions in the visual input, such as a light-pipe (used for illuminating the surgery scene) and forceps (used in retinal-peeling) which are commonly used surgical tools. On average, we report that the network achieves 137 μm accuracy in various unseen scenarios in physical experiments using a silicone phantom, and 94 μm accuracy in simulation on average. Lastly, we propose a change to the baseline network resulting in marked improvement in its performance, specifically by training the network using future images along with waypoints as labeled outputs, which turns out to be a richer representation useful for control. We demonstrate that learning such auxiliary task improves the performance on the tool-navigation task.

II. Related Work

A. Retinal Surgery

Past works in computer-assisted retinal surgery have focused on state estimation or detection systems to assist surgeons with more information about the surgery. For example, image segmentation can be applied to estimate tool-tip and shadow-tip to model proximity when the tips approach close by a predefined threshold pixel-distance [2]. In addition, stereo vision can be employed to estimate the depth of the tool and the retina respectively to create a proximity detection system [3]. More recently, optical coherence tomography (OCT) was utilized to sense depth between the tool-tip and the retina [4], [5]. While these methods do not address automation, they are relevant in the sense that if one could estimate distance from tool-tip to goal, then autonomous navigation may be achieved by interpolating between the two points. Although stereo-vision method may be one alternative, it is not expected to work reliably in a clinical setting due the unknown distortion caused by the patient's lens and out-of-focus images, which may cause depth reconstruction error. OCT is also a promising alternative, however, OCT measures distance locally and is challenging to attach on tool-tips with noncylindrical geometries such as forceps, which are commonly used surgical tools. On the other hand, the learning based approach proposed in this paper can be trained to be robust to visual distortions and navigate complex-shaped surgical tools, given that appropriate training data is available. This is possible since deep networks succeed in the task by imitating the expert, who always succeeds in the task.

B. Learning

Various works have shown the effectiveness of deep learning in sensorimotor control such as playing computer games [6], [7] or navigating in complex environments [8], [9], [10], [11], [12], [13]. In particular, the approach employed in our work borrows from the architecture proposed in [12], where a network is trained to drive a vehicle based on user's high level commands such as "go left" or "go right" at an intersection. Similarly, [13], [11] employ topographical maps to communicate the desired route to a destination selected by a user to drive a vehicle. In our work, we also communicate the goal position as a topographical representation to navigate the surgical tool (Fig. 2). Furthermore, several prior works employ the idea of learning auxiliary tasks to improve accuracy, such as predicting high-dimensional future image conditioned on input (e.g. goal or action) [14], [15], [16], [17], [18], and learning auxiliary tasks for improved sensorimotor control [19], [13].

III. Problem Formulation

We consider the task of autonomously navigating a surgical tool to a desired location using a monocular surgical image and topographical 2D goal-position specified by the user as inputs (Fig. 2). We formulate the problem as a goal-conditioned imitation learning scenario, where the network is trained to map observations and associated goals to actions performed by the expert. The goal-conditioned formulation is necessary to enable user-control of the network at test time (i.e. navigate the surgical tool to a desired location). Given a dataset of expert demonstrations, $D = \{(o_i, g_i), a_i\}_{i=1}^{N}$, where o_i, g_i and a_i denote observation, goal, and action, respectively, the objective is to construct a function approximator $a = F(o; \theta)$

with parameters θ, that maps observation-goal pairs to actions performed by the expert. The objective function can then be expressed as the following:

$$\underset{\theta}{\text{minimize}} \sum_{i=1}^{N} L(a_i, F(o_i, g_i; \theta)), \qquad (1)$$

where L is a given loss function.

In our case, we choose the observation to be an monocular image $o \in \mathcal{I}$ of the surgical scene viewed from top-down, the action $a \in \mathcal{W} \subset \mathbb{R}^3$ to be the 3D euclidean coordinates of a point in the surgical workspace $\mathcal{W}$ or a waypoint, and the goal input to be $g_i = (x_i, y_i) \in \mathbb{R}^2$ which specifies the final desired projected 2D position on the retinal surface. Further details on how the expert dataset is collected and network is trained are given next.

IV. Method

A. Eye Phantom Experimental Setup

Our experimental setup consists of the robot, a surgical needle, and a microscope that records top-down view of the surgery as shown in Fig. 3. For our robot platform, we used the Steady Hand Eye Robot (SHER), which is a surgical robot built specifically for eye surgery applications [20]. The surgical needle is attached at the end-effector with thickness $500 \mu m$ in diameter and its small tip measuring $300 \ \mu m$ in diameter (Fig. 6). The artificial eye phantom (i.e. a rubber eye model) is 25.4mm in diameter, slightly larger than a human eye which ranges $20 - 22.4$mm [21]. To collect data in our experiments, we control the robot using motors attached on the robot joints. We record the images from the microscope and the tool-tip position based on the robot motor encoders.

B. Simulation Setup

In simulation, we used Unity3D software to replicate similar experiment scenarios as the physical experiment as shown in Fig. 3. For sense of scale, the thick part of the tool shaft measures $500 \mu m$ and the tool-tip measures $300 \mu m$ (Fig. 6). We perform domain randomization to change the eye background texture and the lighting condition. Specifically, we created 15 different eyes, each varying in dimension at 20.4mm, 21.2mm, and 22.4mm. These measurements reflect the minimum, medium, and maximum dimension of human eye sizes [21], [22], and 5 eyes were created for each dimension. The texture of the eyes were obtained from [23]. Domain randomization was used to help the network generalize-well to changing brightness conditions, size of the eye, and unseen eye background textures. three eyes from each size were used in training, and the remaining two eyes from each size were used for testing.

C. Data Collection

For real-life experiments, we collected 2000 trajectories in low and high brightness settings. In simulation, we collected 2500 trajectories under a wide range of brightness conditions, while various eyes with different size and backgrounds were randomly replaced. The procedure for data collection were as follows: we initialized the tool at a random position,

then navigated towards a randomly selected position below the eye in a straight-line trajectory. We use straight-line trajectories as a way to generate predictable and simple ground-truth expert data. When collision was detected between the tool and the eye phantom using a force sensor at the end-effector (Fig. 3), we logged the images and the tool-tip positions of the trajectory. For simplicity, we kept the orientation of the tool fixed and only moved the XYZ-position of the tool. The position of the eye remained fixed as well.

To synthesize the goal for each trajectory, we used the last XY tool-tip position of the executed trajectory and plotted it as a white square with dark background as illustrated in Fig. 2. Effectively, we changed the 2-dimensional coordinate representation of the goal to an image representation. This design choice can be useful in an application where a surgeon may simply use a mouse to click the goal location in the visual-feed of the surgery, and the network will navigate the surgical tool to the exact location of the white square corresponding to the surgical scene (Fig. 2).

We note that our method for synthesizing goal images do not guarantee precise spatial correspondence between the tool-tip position and the goal image. This is due to perspective projection, where objects further away from the camera are subject to shift towards the vanishing point, but the plotted goal position, which is obtained from robot kinematics, is not subject to perspective projection. One way to guarantee spatial correspondence is to manually annotate the tool-tip positions to create the goal images; however, we leave this for future work. Our objective is to assess how consistently the network can generate desired trajectories given a particular goal image. Thus, similar consistency in performance is expected given manually-annotated goal images with precise spatial correspondences.

D. Network Details and Training

The input to the network are the current image of the surgery ($224 \times 224 \times 3$) and the goal image ($224 \times 224 \times 1$) stacked along the channel dimension, yielding a combined dimension of ($224 \times 224 \times 4$). The output of the network is a XYZ-waypoint in the surgical workspace (three-dimensional vector) which the network must travel in order to reach closer to the target location on the retina. Specifically, for a single trajectory consisting of n frames $I_1, \dots I_n \in \mathcal{I}$, n tool-tip positions $p_1, \dots, p_n \in \mathbb{R}^3$, and the goal image coordinate $g \in \mathbb{R}^2$ specific to this trajectory, a single sample is expressed as (input, output) = $((I_t, g), p_{t+d})$, for $t = 1, \dots, n - d$, where d is a parameter denoting the *look-ahead* of the commanded action, which is used as a feed-forward reference signal to the robot. We chose $d = 8$, which is equivalent to approximately $70 \mu m$ apart between learned waypoints to ensure that the network moved the surgical tool by a noticeable distance every control cycle. The complete data set D is constructed using multiple such trajectories and their corresponding samples. Internally, the network maps the goal g into an image $I_g \in \mathcal{I}$ which is concatenated with the actual camera image I_i to form the complete network input.

We experimented with two architectures, a baseline network that predicts an XYZ waypoint and another network that predicts an XYZ waypoint plus the future image as shown in Fig. 4, which we refer to as extended network. The extended network was enforced to learn an auxiliary task which helped with the main objective of predicting accurate waypoints. For

the extended network, a single training sample is (input, output) = $((I_t, g), (p_{t+d}, I_{t+d}))$. In the following, we discuss each network in greater detail.

1) Baseline Network—We use Resnet-18 [24], which takes high dimensional image $(224 \times 224 \times 4)$ as input to 512-dim feature vectors. To learn the waypoints or the action output, we discretize the continuous X, Y, and Z coordinate representation into 100 steps. Specifically, we add a fully-connected layer outputting 300 neurons on top of the 512-feature vectors, where each 100 neurons is a discretized representation of the continuous X, Y, and Z coordinates of the euclidean surgical workspace respectively. The network was trained using cross-entropy loss with Adam optimizer [25] with a learning rate of 0.0003 and batch size of 170. The loss function is defined as

$$L(b, \hat{p}) = \sum_{j \in \{x, y, z\}} \sum_{c=1}^{M_j} - b_{j,c} \log(\hat{p}_{j,c}), \tag{2}$$

where $b_{j,c}$ are binary indicators for the true class label c, and $\hat{p}_{j,c}$ are the predicted probability that the coordinate j is of class c. The cost combines the errors for all three dimensions $j \in \{x, y, z\}$. As specified above, we employed $M_x = M_y = M_z = 100$ discrete bins.

2) Baseline + Predicting Future Image (Extended Network)—The extended architecture aims to achieve the baseline task and additionally predict future images. The architecture is shown in Fig. 4. On top of the Resnet-18 architecture, a decoder network with skip-connections is added. The waypoints were trained using cross-entropy loss similar to the baseline network and the future-image prediction was trained using RMSE function. The network is trained using Adam optimizer with a learning rate of 0.0003 and batch size of 120. The combined loss function is given as

$$L((b, I), (\hat{p}, \hat{I})) = \sum_{j \in \{x, y, z\}} \sum_{c=1}^{M_j} - b_{j,c} \log(\hat{p}_{j,c}) + (I - \hat{I})^2, \tag{3}$$

where $\hat{I}$ denotes the future-image prediction by the network and I denotes the label for the future image. To balance the loss functions, drop-out approach was used where we performed back-propagation 70% of the time for the future-image loss term.

E. Data Augmentation

For robust learning, we utilized data augmentation such as random drop-out of pixels, Gaussian noise, and random brightness, contrast, and saturation. We also expanded the initialization space so that the network could reach the same target location from various initial positions as shown in Fig. 5. This effectively enabled the network to recover from mistakes when it deviated from its hero path [26]. These techniques were crucial to generalizing well to unseen environments.

V. Results and Discussion

A. Physical Experiment Results

To assess the accuracy of our networks, we performed benchmark experiments where the baseline and the extended network visited 50 predefined locations in the training region in grid-like fashion (5×10), starting from three different initial locations as shown in Fig. 6. The objective of such experiment was to test how accurately the network could navigate to various targeted locations, given various goal inputs. We tested each network in the familiar and unseen environments as listed in Table I and Fig. 6. For experiments with tool-distractions, we only tested from the right-most initial position out of the three, since a human had to hold the tools throughout the long experiments. The light-pipe and forceps were dynamically maneuvered to follow the tool-tip. Both tools occasionally occluded the surgical tool and its shadow.

Our experimental results are summarized in Table I and the executed trajectories are shown in Fig. 9. The table contains numeric XY-error values in reaching the goal position under various test conditions. Since the eye position is fixed during training and experimental validation, the error can be calculated by comparing the input goal-image coordinate (x, y) against the final landing position of the surgical tool (x', y') after the trajectory execution is complete (e.g. when force is detected using the force sensor). Thus, the error reported in Table I is calculated using the formula $\sqrt{(x - x')^2 + (y - y')^2}$. The accuracy reported in Table II are the classification accuracies achieved on the validation dataset offline, not the online benchmark experiments. In Table II, we are able to report errors in the z-axes (depth) because we have ground-truth xyz-values of the full trajectory from the previously collected dataset.

Our results show that the both baseline and extended network generalizes well to unseen scenarios, achieving $166 \mu m$ and $137 \mu m$ in error, even in the presence of unseen brightness conditions and unseen surgical tools significantly occluding the scene. The extended network also performed marginally better than the baseline network. This result is expected given the higher accuracy achieved by the extended network in the validation dataset, achieving 2.5% higher accuracy than the baseline (Table II). Also, as shown in Fig. 7, the extended network trains faster and is more data-efficient than baseline network, achieving best classification accuracy on the 18th epoch versus 26th epoch by the baseline network. In addition to improving the baseline network performance, the extended network is able to predict clear future images. The extended network can imagine different futures depending on various goal inputs (e.g. move the tool forward, left, right), recognize the surgical tool as a dynamic object apart from the static background, and also reliably reconstruct surgical objects it has never seen during training (Fig. 8).

B. Simulation Experiment Results

Similar to real-life experiments, we performed benchmark tests where each baseline and extended network visited 100 predefined locations in the training region in grid-like fashion (10×10), starting from three different initial locations as shown in Fig. 6. We tested each network in the following conditions as listed in Table III and shown in Fig. 6. For

experiment with tool-distractions, we only tested from the middle initial position out of the three. Both the light-pipe and forceps were moved randomly to imitate hand-tremor, and both tools occasionally occluded the surgical tool and its shadow.

The simulation results are summarized in Table III and the executed trajectories are shown in Fig. 9. The errors shown in Table III are calculated using the same formula mentioned in the real-life experiment results, specifically using the formula $\sqrt{(x - x')^2 + (y - y')^2}$ where (x, y) denotes input goal-image coordinate and (x', y') denotes the final landing position of the surgical tool after trajectory execution. Similarly, Table IV shows network results on the validation dataset. Our results show that both baseline and extended networks achieve good performance and can generalize robustly to unseen scenarios, even in the presence of unseen eye backgrounds and unseen surgical tools occluding the scene. Similar to real-life experiments, the extended network also performed marginally better than the baseline network. This result is expected since the extended network achieved 4.9% higher accuracy than the baseline network in the validation dataset (Table IV). Similar to real-life experiments, the extended network is also more data-efficient than the baseline network, achieving maximum accuracy at 13th epoch versus 9th epoch by the baseline network (Fig. 7) and achieving significantly higher maximum validation accuracy.

VI. Conclusions

In this work, we show that deep networks can reliably navigate a surgical tool to various desired locations within 137 μm accuracy in physical experiments and 94 μm in simulation on average. In future work, we hope to consider more realistic scenario including a physical scelera constraint, eye movement, and also using porcine eyes.

Acknowledgements

References

[1]. Juliani A, Berges V-P, Vckay E, Gao Y, Henry H, Mattar M, and Lange D, "Unity: A general platform for intelligent agents," 2018.

[2]. Tayama T, Kurose Y, Marinho MM, Koyama Y, Harada K, Omata S, Arai F, Sugimoto K, Araki F, Totsuka K, Takao M, Aihara M, and Mitsuishi M, "Autonomous positioning of eye surgical robot using the tool shadow and kalman filtering," in 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), July 2018, pp. 1723–1726.

[3]. Richa R, Balicki M, Sznitman R, Meisner E, Taylor R, and Hager G, "Vision-based proximity detection in retinal surgery," IEEE Transactions on Biomedical Engineering, vol. 59, no. 8, pp. 2291–2301, 8 2012. [PubMed: 22692869]

[4]. Ourak M, Smits J, Esteveny L, Borghesan G, Gijbels A, Schoevaerdts L, Douven Y, Scholtes J, Lankenau E, Eixmann T, Schulz-Hildebrandt H, Hüttmann G, Kozlovszky M, Kronreif G, Willekens K, Stalmans P, Faridpooya K, Cereda M, Giani A, Staurenghi G, Reynaerts D, and Vander Poorten EB, "Combined oct distance and fbg force sensing cannulation needle for retinal vein cannulation: in vivo animal validation," International Journal of Computer Assisted Radiology and Surgery, vol. 14, no. 2, pp. 301–309, 2 2019. [Online]. Available: 10.1007/s11548-018-1829-0 [PubMed: 30056592]

[5]. Smits J, Ourak M, Gijbels A, Esteveny L, Borghesan G, Schoevaerdts L, Willekens K, Stalmans P, Lankenau E, Schulz-Hildebrandt H, Httmann G, Reynaerts D, and Vander Poorten EB,

"Development and experimental validation of a combined fbg force and oct distance sensing needle for robot-assisted retinal vein cannulation," in 2018 IEEE International Conference on Robotics and Automation (ICRA), May 2018, pp. 129–134.

[6]. Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G, et al. , "Human-level control through deep reinforcement learning," Nature, vol. 518, no. 7540, p. 529, 2015. [PubMed: 25719670]

[7]. Dosovitskiy A and Koltun V, "Learning to act by predicting the future," 2016.

[8]. Wang X, Huang Q, elikyilmaz A, Gao J, Shen D, fang Wang Y, Wang WY, and Zhang L, "Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation," ArXiv, vol. abs/1811.10092, 2018.

[9]. Bojarski M, Testa DD, Dworakowski D, Firner B, Flepp B, Goyal P, Jackel LD, Monfort M, Muller U, Zhang J, Zhang X, Zhao J, and Zieba K, "End to end learning for self-driving cars," 2016.

[10]. Xu H, Gao Y, Yu F, and Darrell T, "End-to-end learning of driving models from large-scale video datasets," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017, pp. 3530–3538.

[11]. Amini A, Schwarting W, Rosman G, Araki B, Karaman S, and Rus D, "Variational autoencoder for end-to-end control of autonomous driving with novelty detection and training de-biasing," in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Oct 2018, pp. 568–575.

[12]. Codevilla F, Miiller M, López A, Koltun V, and Dosovitskiy A, "End-to-end driving via conditional imitation learning," 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 1–9, 2018.

[13]. Bansal M, Krizhevsky A, and Ogale A, "Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst," 2018.

[14]. Oh J, Guo X, Lee H, Lewis RL, and Singh S, "Action-conditional video prediction using deep networks in atari games," in Advances in Neural Information Processing Systems 28, Cortes C, Lawrence ND, Lee DD, Sugiyama M, and Garnett R, Eds. Curran Associates, Inc., 2015, pp. 2863–2871. [Online]. Available: http://papers.nips.cc/paper/5859-action-conditional-video-prediction-using-deep-networks-in-atari-games.pdf

[15]. Gafni O, Wolf L, and Taigman Y, "Vid2game: Controllable characters extracted from real-world videos," CoRR, vol. abs/1904.08379, 2019. [Online]. Available: http://arxiv.org/abs/1904.08379

[16]. Paxton C, Barnoy Y, Katyal K, Arora R, and Hager GD, "Visual robot task planning," in 2019 International Conference on Robotics and Automation (ICRA), May 2019, pp. 8832–8838.

[17]. Finn C and Levine S, "Deep visual foresight for planning robot motion," 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 2786–2793, 2016.

[18]. Isola P, Zhu J, Zhou T, and Efros AA, "Image-to-image translation with conditional adversarial networks," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017, pp. 5967–5976.

[19]. Mirowski PW, Pascanu R, Viola F, Soyer H, Ballard A, Banino A, Denil M, Goroshin R, Sifre L, Kavukcuoglu K, Kumaran D, and Hadsell R, "Learning to navigate in complex environments," ArXiv, vol. abs/1611.03673, 2016.

[20]. Üneri A, Balicki MA, Handa J, Gehlbach P, Taylor RH, and Iordachita I, "New steady-hand eye robot with micro-force sensing for vitreoretinal surgery," in Biomedical Robotics and Biomechatronics (BioRob), 2010 3rd IEEE RAS and EMBS International Conference on. IEEE, 2010, pp. 814–819.

[21]. Bekerman Gottlieb, Paul, and Michael, "Variations in eyeball diameters of the healthy adults," 11 2014. [Online]. Available: https://www.hindawi.com/journals/joph/2014/503645/

[22]. Vurgese S, Panda-Jonas S, and Jonas JB, "Scleral thickness in human eyes," PLOS ONE, vol. 7, no. 1, pp. 1–9, 01 2012. [Online]. Available: 10.1371/journal.pone.0029692

[23]. "Diabetic retinopathy detection." [Online]. Available: https://www.kaggle.com/c/diabetic-retinopathy-detection/overview/description

[24]. He K, Zhang X, Ren S, and Sun J, "Deep residual learning for image recognition," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016, pp. 770–778.

[25]. Kingma DP and Ba J, "Adam: A method for stochastic optimization," CoRR, vol. abs/1412.6980, 2014.

[26]. Ross S, Gordon G, and Bagnell D, "A reduction of imitation learning and structured prediction to no-regret online learning," in Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, ser. Proceedings of Machine Learning Research, Gordon G, Dunson D, and Dudk M, Eds., vol. 15. Fort Lauderdale, FL, USA: PMLR, 11–13 Apr 2011, pp. 627–635. [Online]. Available: http://proceedings.mlr.press/v15/ross11a.html

Illustration of Robot-Assisted Surgery
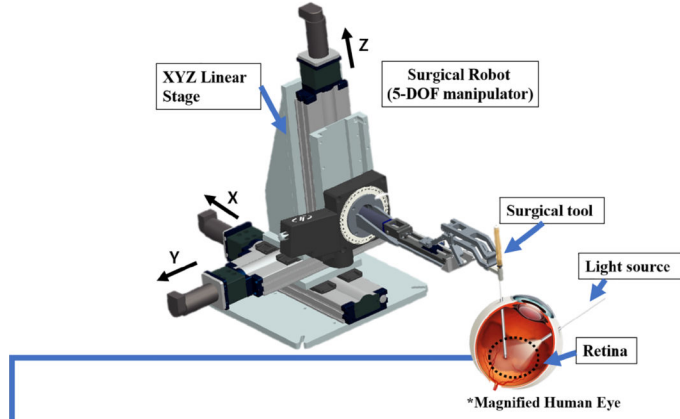
Illustration of Tool-Shadow Dynamics
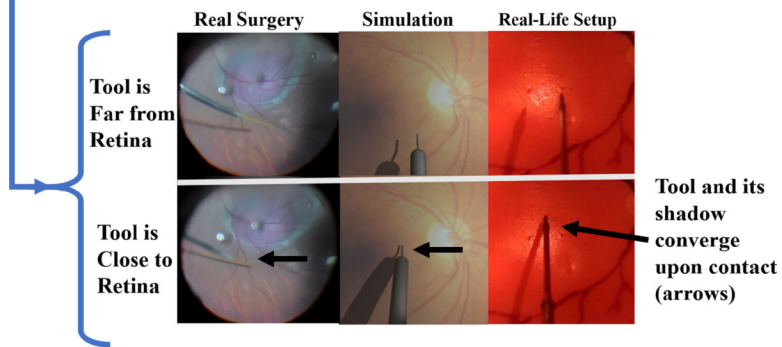
Fig. 1:
(Top) During robot-assisted retinal surgery, a light source projects a shadow on the retina which can be used as cues to estimate proximity between the tool-tip and the retina. (Bottom) Demonstration of tool-shadow dynamics; as the surgical tool approaches close to the retina, the tool and its shadow converge (compare top row to bottom row), which can be used as cues to train a network how to navigate inside the eye.
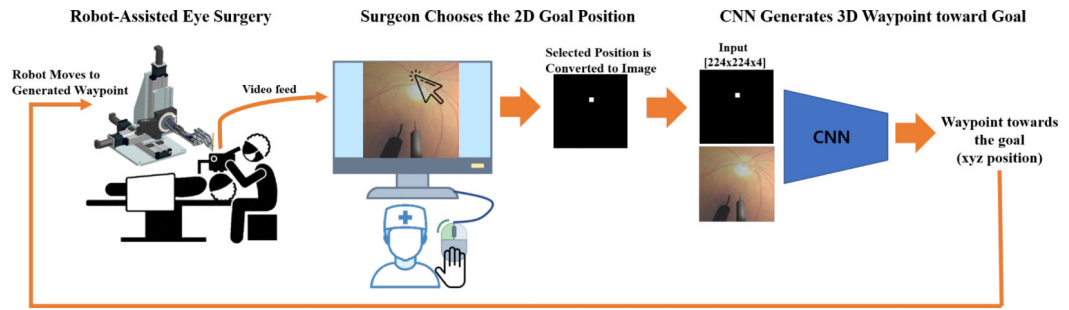
Fig. 2:

System setup: a surgeon chooses the goal location in 2D and the network generates a 3D waypoint that navigates the surgical tool toward the selected location.
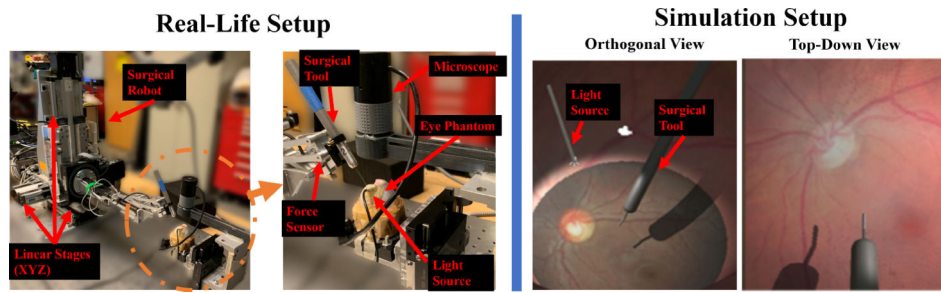
Fig. 3:

(Left) Real-life experimental setup using an eye phantom. (Right) Experimental setup in simulation
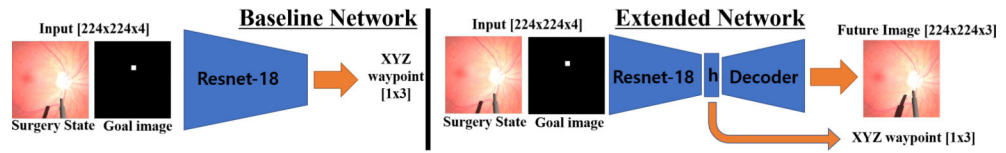
Fig. 4:
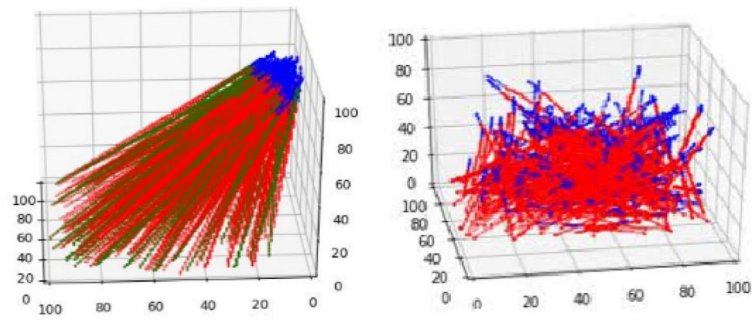(Left) Baseline network (Right) Extended network

Fig. 5:
(Left) training trajectories with small initialization region; blue tips indicate starting position of the trajectory (Right) Training trajectories with large initialization space as data augmentation.
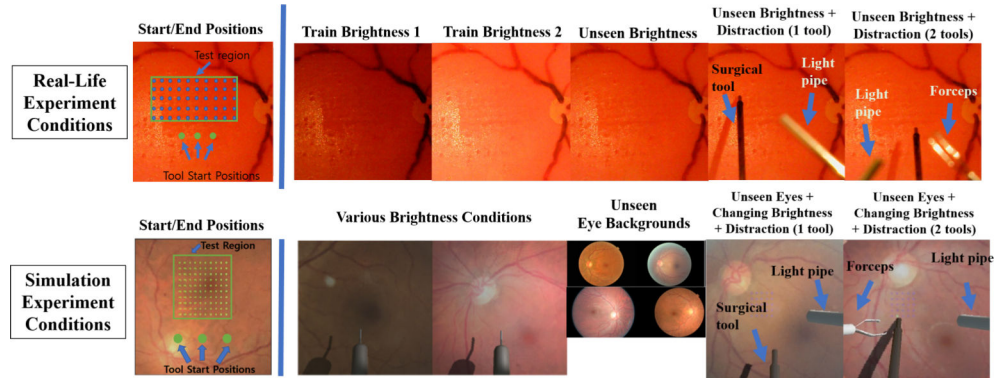
Fig. 6:
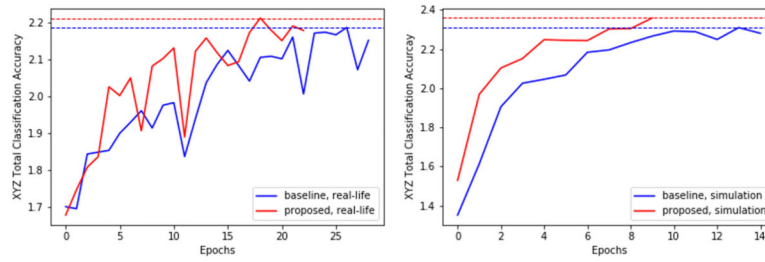(Top) Example test conditions in real-life physical experiment and (Bottom) in simulation.

Fig. 7:
(Left) Waypoint prediction accuracies on the "real-life" validation dataset and in (Right) simulation. Y-axis is the sum of the classification accuracies for xyz axes (i.e. maximum possible is 3.0). Dashed lines mark the maximum accuracy achieved.
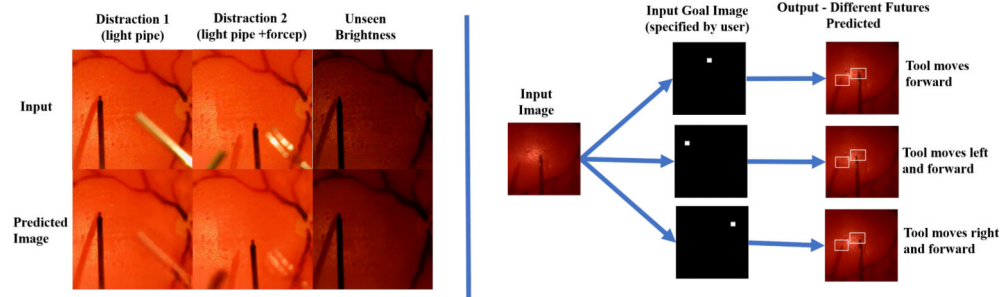
Fig. 8:

(Left) Future prediction by the extended network under various unseen conditions (Right) Different futures predicted by changing the goal input (rectangle frames are fixed, added to clarify shifted positions of the tool)
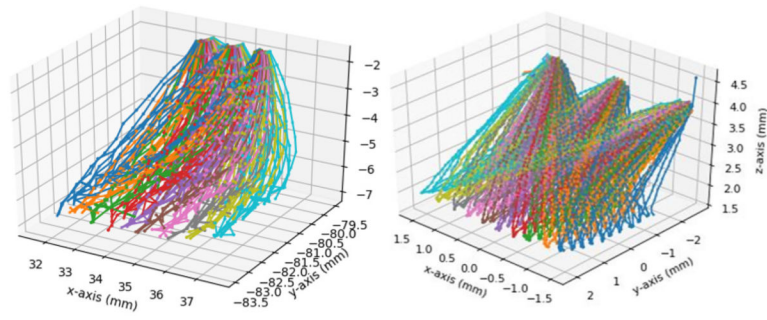
Fig. 9:

(Left) Trajectories executed in real-life in unseen brightness condition (Right) trajectories executed in real-life in changing brightness condition + unseen eyes

TABLE I:

Eye Phantom Experiment Results

Test Condition	Baseline Network Error (mm)	Extended Network Error (mm)
Train Low Brt.1	0.134	0.139
Train High Brt. 2	0.092	0.108
Unseen Brt.	0.177	0.127
Unseen Brt.+Distr. (1 tool)	0.165	0.146
Unseen Brt.+Distr. (2 tools)	0.155	0.137
"Unseen" Avg. (above 3 rows)	**0.166**	**0.137**

TABLE II:

Eye Phantom Training Results

Axes	Baseline Val. Acc. (%)	Extended Network Val. Acc. (%)
X	82.0	82.8
Y	76.0	76.7
Z (Depth)	60.8	61.9
XYZ Total Sum	**218.8**	**221.3**

TABLE III:

Simulation Experiment Results

Testing Condition	Baseline Network Error (mm)	Extended Network Error (mm)
Train	0.107	0.098
Unseen Eyes	0.102	0.096
Unseen Brt. + Distr. (1 tool)	0.140	0.100
Unseen Brt. + Distr. (2 tools)	0.169	0.087
"Unseen" Avg. (above 3 rows)	**0.137**	**0.094**

TABLE IV:

Simulation Training Results

Axes	Baseline Val. Acc. (%)	Extended Network Val. Acc. (%)
X	78.9	81.4
Y	84.8	84.0
Z (Depth)	67.3	70.6
XYZ Total Sum	**231.0**	**235.9**