Data-Driven Visual Tracking in Retinal Microsurgery

Raphael Sznitman[1], Karim Ali[1], Rogerio Richa[2], Russell H. Taylor[2], Gregory D. Hager[2], and Pascal Fua[1]

[1] École Polytechnique Fédérale de Lausanne, Switzerland
[2] The Johns Hopkins University, Baltimore, USA
{firstname.lastname}@epfl.ch, {richa,rht,hager}@jhu.edu

Abstract. In the context of retinal microsurgery, visual tracking of instruments is a key component of robotics assistance. The difficulty of the task and major reason why most existing strategies fail on *in-vivo* image sequences lies in the fact that complex and severe changes in instrument appearance are challenging to model. This paper introduces a novel approach, that is both data-driven and complementary to existing tracking techniques. In particular, we show how to learn and integrate an accurate detector with a simple gradient-based tracker within a robust pipeline which runs at framerate. In addition, we present a fully annotated dataset of retinal instruments in *in-vivo* surgeries, which we use to quantitatively validate our approach. We also demonstrate an application of our method in a laparascopy image sequence.

1 Introduction

Retinal microsurgery (RM) is one of the few available treatments options for many blinding eye conditions. During surgery, the operating surgeon uses a stereo microscope to visualize the retina and manipulates a set of surgical instruments (*i.e.* tipped forceps or picks) to perform the procedure, as depicted in Fig. 1.

Given its importance and the demanding nature of the surgery, a number of new technologies have focused on improving aspects of RM. Some of these technologies have included a steady-hand robot [1] or an instrument capable of visualizing anatomical structures below the surface of the retina via optical coherence tomography [2]. Yet, for these technologies to fully develop and ultimately be incorporated into clinical environments, one missing component is the ability to accurately and reliably estimate the location of an instrument when in the camera field of view. With this in mind, this paper focuses on real-time visual tracking of instruments in *in-vivo* RM monocular image sequences.

A major difficulty with this task is that instrument appearance is difficult to model well over time. Most existing methods have relied instead on knowing the instrument geometry beforehand to solve complex optimization problems [3, 4], or have constructed sophisticated and robust objective functions within more traditional gradient-based frameworks to deal with appearance change [5, 6]. Typically, these methods have extremely simple appearance models that combine

geometry with colour or edge-based features, and ultimately work well only in limited conditions such as in eye phantoms. For example, using the method of [6], tracking is often lost after only 5 frames in the *in-vivo* sequence of Fig. 1. Note that a similar observation can also be made regarding tool tracking techniques for laparoscopic surgery [7].

In short visual tracking of instruments in *in-vivo* RM is characterized by complex appearance changes that existing approaches fail to handle. In contrast, this paper introduces an alternative approach, one that is data-driven and complementary to the aforementioned methods. In particular, we show how to integrate the framework of [8], which constructs accurate classifiers, for the task of instrument detection. Coupled with simple gradient-based tracking, our pipeline is extremely robust and runs at video framerate. In addition, we present a fully annotated dataset of retinal instruments in human *in-vivo* surgeries and quantitatively validate our pipeline on this dataset. Finally, we also demonstrate how our approach performs on a laparoscopy image sequence.

The remainder of this paper is organized as follows: We begin by describing our pipeline and its components in Sec. 2. In Sec. 3 we validate our method experimentally and conclude with final remarks in Sec. 4.

2 Method

To motivate our approach and pipeline, we begin with the following observations:

1. To work reliably, gradient-based trackers [6, 9] need continuous template updating to maintain accurate position estimation when changes in the target appearance are severe.
2. Using reasonable amounts of training data (*e.g.* 500 positive examples), classifiers as in [8] provide excellent methods to detect the 2D location of a deformable target irrespective of its orientation.
3. Given that tracking is a sequential estimation problem, detection of targets can be restricted to promising locations provided by fast and moderately accurate methods.

Based on these observations, we propose a detection based scheme to track the 2D instrument tip position in *in-vivo* RM image sequences. Once initialized, our pipeline operates as follows: we first use a gradient based tracker to provide an approximate estimate of the target's new location. We then exhaustively evaluate a detector to predict the presence of an instrument in a reduced region of the image space, which is parametrized by tracker's estimate from the previous step. Finally, we use spatial and score weighting of the detector responses to provide accurate instrument position, and update the tracker template. This process is depicted in Fig. 1, and the following sections describe each component in detail. Note that, initialization of the instrument position, and reinitialization when the instrument is not found, is achieved by using the constructed detector and hence no user input is required in our pipeline.

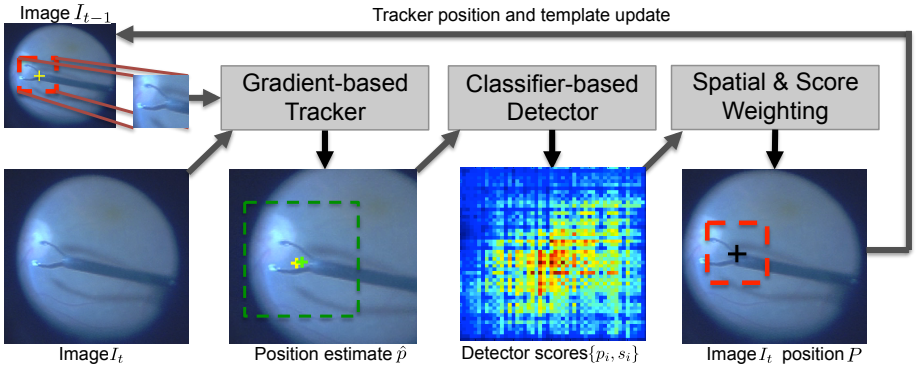

Fig. 1. Pipeline Diagram: First, an updated template and the previous frame instrument position (yellow cross) are used to initialize a gradient based tracker. The new position estimate, $\hat{p}$ (green cross) serves as the center of the region of interest (green box), that the detector evaluates at every location. At each location, positions and scores $\{p_i, s_i\}$ are computed and weighted to provide the final instrument position, P (black cross).

2.1 Tracking

In order to provide an approximate location for the instrument position, we first compute the displacement of a window centered at the previous tool tip location using a gradient-based tracking method. The method is based on Efficient Second-Order Minimization [9]. Assuming that no large illumination variations occur between sequential images, SSD was adopted as similarity measure. The reference template used in this step is updated at every new image using the tool tip position estimated from the previous image. In our experiments, we maintain a fixed template size: 50×50 pixels. This process results in a tool tip estimate, which we denote as $\hat{p}$. Fig. 1 shows an example of $\hat{p}$ (green cross) on a given frame. Note that alternative similarity measures could be substituted instead.

2.2 Detector

The strong appearance changes of tools during RM severely complicate the detection task. Standard learning based detection methods can only cope with deformations and rotations via a detailed labeling of training data and an exhaustive exploration of these parameters when evaluating the classifier. This latter point makes detection of targets particularly slow, which helps explain their lack of use so far. Recently however, a framework was presented in [8] which overcomes these difficulties. The authors design a set of so-called pose-estimator features which modulate feature extraction according to various image cues. The result is a deformable detector which can learn the deformations and rotations present in the training data. The method, based on AdaBoost, does not require

an exploration of the pose parameters at test time and is thus well-suited for the task at hand.

We therefore use this framework along with the proposed set of deformable features, which compute sums of oriented edges in various image areas. To train this detector, positive and negative examples must be provided (*i.e.* instrument and non-instrument images, respectively). Here, we use square bounding boxes of the instruments indicating the location and spatial extent of the instrument for positive samples. Negative samples are randomly selected from the remainder of the images. One additional difficulty, not considered in [8], is how to efficiently train such a deformable detector of fixed size $r \times r$ from an image sequence exhibiting multi-scale data. To this end, we compute a Gaussian Pyramid for each image by successive smoothing and downsampling. For each positive example, its bounding box is replaced by an appropriately located box of size $r \times r$ at the Gaussian Pyramid level l which results in the best $r \times r$ approximation of the original sample. Detection proceeds in a similar fashion with each image decomposed into a Gaussian Pyramid and our $r \times r$ detector exhaustively visiting every location in the Pyramid.

Given the approximate instrument position provided by the tracker, we only evaluate the classifier at each location in a 50×50 region of the image, centered on the position estimate provided by the tracker. This results in a set of pixel positions and associated unsigned classifier score, $\{p_i, s_i\}$.

2.3 Estimating Instrument Position

Given the position estimate of the tracker, $\hat{p}$ and the set of detection scores $\{p_i, s_i\}$, we now describe how to combine these estimates to provide the final instrument location.

We first perform a weighting of the classifier scores with regard to their spatial placement. In particular, we favour locations that are near the position estimate provided by the tracker. That is, we first compute spatially adjusted scores, $\tilde{s}$, $\tilde{s_i} = s_i e^{-\frac{1}{2\sigma^2}(p_i - \hat{p})^2}$, where σ is half the radius of the search window ($\sigma = \frac{50}{2} = 25$ in our experiments). Then, instead of doing non-maximum suppression as in [8], we estimate the final position of the instrument, P, by averaging the weighted scores, $\tilde{s_i}$, $P = \frac{\sum_i^N \tilde{s_i} p_i}{\sum_i^N \tilde{s_i}}$. This effectively reduces the effect of extreme scores and outlier influence by weighted voting. We consider a detection valid if the score associated with the location P is above a threshold (in practice it is set to provide a 80% true positive rate). When no instrument is found in a frame, then detector is then evaluated at all locations of subsequent images, until a new instrument location is found.

3 Experiments and Results

The presented pipeline is implemented in C++, and all experiments were performed on a MacBook Pro, 2.5 GHz Quad core computer with 4GB RAM. Our pipeline runs at 15fps and should run even faster implemented on a GPU.

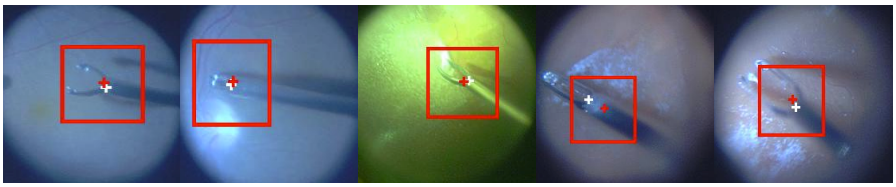

Fig. 2. Example of instrument detections (red) in our *in-vivo* dataset with annotated ground truth (white cross). The pixel distance error for each example are: 8, 7, 11, 25 and 18. See Video 1 for full detection and tracking sequence.

3.1 Retina Microsurgery Dataset:

We begin by introducing a fully annoted dataset for RM instrument detection and tracking. This image set consists of 4 sequences of *in-vivo* vitreoretinal surgery, containing a total of 1500 images (640 $\times$ 480 pixels). Fig. 2 shows representative images from the dataset, illustrating variations in illumination type and quantity, light source position and the presence of blur and shadows. Different types of cameras were used to acquire the images but in each case the video was collected directly from the surgical microscope. Calibration data was not available since the surgeon frequently varies the focal length during the procedure. Each image contains at most one instance of a tool, with some images being tool free. The tool tip of each instrument has been annotated by hand. This dataset is publicly available online via the corresponding authors website, at https://sites.google.com/site/sznitr/.

Full Dataset Evaluation In our first experiment, we trained our classifier by using the first half of each sequence in the above dataset and evaluated our method on the remaining sequence halves. The result of our pipeline can be seen in Video 1, with some snapshots shown in Fig. 2 (see above website for associated videos). In general, consistent tracking is achieved even in cases of strong appearance changes.

To provide some quantitative validation of our method we plotted the proportion of frames where the instrument tip was determined correctly, as function of sensitivity of the detection criteria. More specifically, we defined a correct detection to be any pixel estimation that is within δ pixels of the groundtruth annotation. Fig. 3(left) show this plot when varying δ between 15 and 40. 15 pixel may appear as a large starting threshold, but consider that the average tool shaft diameter in the dataset is of 20 pixels, and due to blurring and illumination changes throughout the sequences, the annotations themselves are noisy (see Fig. 2). Hence, smaller threshold results are not particularly meaningful here.

We compare our approach to three existing gradient based trackers on the same set of images: the Mutual Information of [6], the SCV of [10] and the SSD tracker used in this pipeline. To allow a fair comparison, when any of these

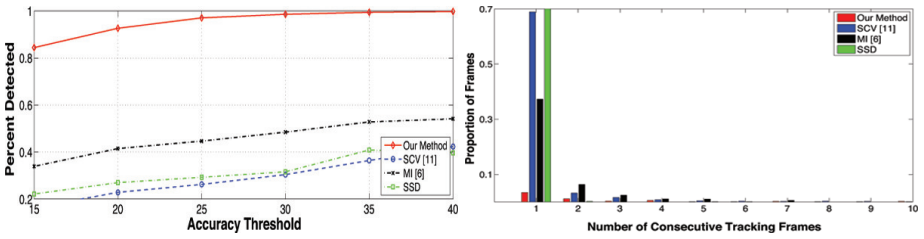

Fig. 3. Tracking Accuracy. *(left)* we show the percent of correctly detected instruments as function of the accuracy threshold. In red, our approach clearly outperforms state-of-the-art gradient-based trackers. *(right)* Proportion of frames for each number of consecutive correct detections.

trackers provided false detections they were re-initialized with the ground truth, and we report the proportion of frames where re-initialization is not required. From the figures, we clearly see that our approach outperforms that all three trackers. For example, for $\delta = 20$ our approach detects over 70% more than [6] and over 40% more than [10]. This corresponds to 449 and 309 more correct instrument detections, respectively. We also show in Fig. 3(right) the proportion of time where a certain number of consecutive correct detections ($\delta = 20$) took place. In particular, we see that the SCV and MI can only track for 1 frame over 35% and 65% of the time, while this only occurs 11% of the time for our approach. On average our method tracks for 25 consecutive frames while the SCV and MI achieve 2 and 5 frames, respectively. Also, in the cases where our method did lose tracking, correct reinitialization occurred on average after 1.5 frames.

Generalization: Detection-based methods as this one are often criticized for needing large amounts of training data and only working well on images similar to those found in the training set. To demonstrate, that this can be avoided, we show that even when training our classifier on three sequences, and testing on an unseen fourth, reliable tracking is achieved. As in typical cross-validation protocols, we trained our classifier on 3 sequences, and tested on the remaining set. We did this for 3 different sets (the 4th set was not usable in this case, since it contains no instruments in it). Fig. 4 shows training and testing image examples for each experiment (*i.e.* Exp.2a through c). Videos 2a though 2c show the tested sequences for these experiments. As in the previous case, we plotted detection accuracy against the detection criteria, showing that our method significantly outperforms [6] and [10] on all three sequences.

3.2 Laparoscopy Sequence

Finally, we briefly show how our approach can work for laparoscopic instrument tracking. Here, we downloaded a video sequence from Youtube[3], extracted images and hand labeled the locations of instruments in 1000 images. This provided

[3] URL: http://www.youtube.com/watch?v=IVp1sgjQ5To

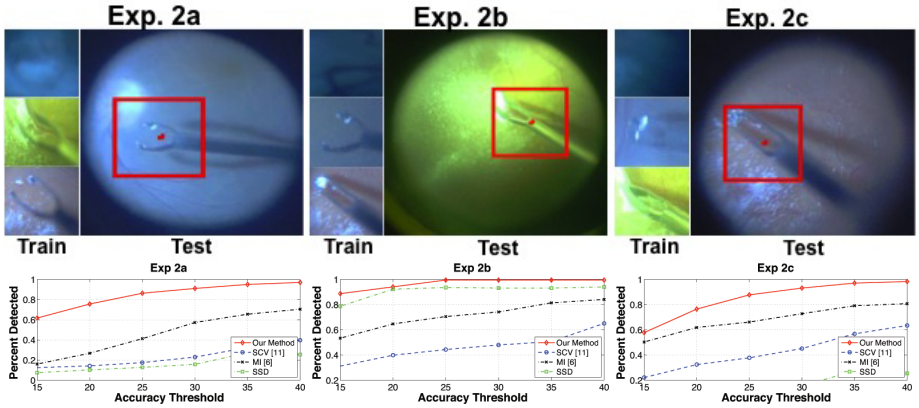

Fig. 4. Generalization experiments by training on 3 image sequences and tested on an unseen fourth sequence. Accuracy plots are also shown for each experiment.

roughly 2000 instrument locations (two instruments per image). From this, we trained our classifier on the first 500 images and evaluated our pipeline on the remaining images. Given, that two instruments are present in frames, we proceeded as in the RM case, found an instrument, suppressed it from the image, and repeated this process for the second tool. Otherwise, the pipeline is identical to that of the previous experiments.

Video 5 shows the result of our pipeline, of which a few frames are shown in Fig. 5. In summary, tracking is maintained for a substantial number of frames. However, two main failing points can be seen: 1) Extreme changes in instrument structure, that were not observed in the training sequences, are poorly handled by our system (as shown in Fig. 5(right)), 2) occluded instruments are not found given that there is no geometrical model to help with such situations. A possible alternative to overcome this may be to integrate our approach with more elaborate prior instrument knowledge (as in [3, 4, 7]).

4 Conclusion

We presented an alternative approach for visual detecting and tracking retinal instruments during *in-vivo* retinal microsurgery. Our technique involves training a highly accurate instrument detector, coupled with a simple gradient based tracker to produce reliable tracking. Soft weighting of both classifier scores and locations are fused to produce accurate position estimates even in challenging cases. We extensively validated our method on a fully annotated *in-vivo* dataset, where we showed consistent tracking. We also demonstrated the applicability of our approach on a laparoscopy image sequence.

Acknowledgements: Funding for this research was provided in part by NIH Grant R01 Eb 007969-0 and internal JHU funds.

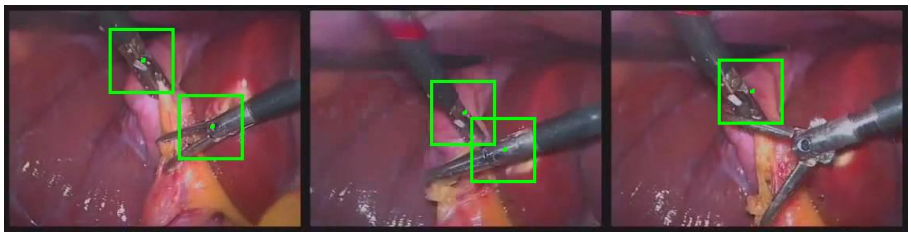

Fig. 5. Example of our approach tracking two instruments during Laparoscopic surgery.

References

1. Uneri, A., Balicki, M., Handa, J., Gehlbach, P., Taylor, R., Iordachita, I.: New steady-hand eye robot with micro-force sensing for vitreoretinal surgery. In: 3rd IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob), 2010. (sept. 2010) 814 –819
2. Balicki, M., Han, J.H., Iordachita, I., Gehlbach, P., Handa, J., Taylor, R., Kang, J.: Single fiber optical coherence tomography microsurgical instruments for computer and robot-assisted retinal surgery. In Yang, G.Z., Hawkes, D., Rueckert, D., Noble, A., Taylor, C., eds.: MICCAI 2009. Volume 5761 of LNCS. Springe, Heidelberg (2009) 108–115
3. Pezzementi, Z., Voros, S., Hager, G.D.: Articulated object tracking by rendering consistent appearance parts. In: IEEE International Conference on Robotics and Automation, 2009. (may 2009) 3940 –3947
4. Sznitman, R., Basu, A., Richa, R., Handa, J., Gehlbach, P., Taylor, R., Jedynak, B., Hager, G.: Unified detection and tracking in retinal microsurgery. In Fichtinger, G., Martel, A., Peters, T., eds.: MICCAI 2011. Volume 6891 of LNCS. Springer, Heidelberg (2011) 1–8
5. Burschka, D., Corso, J.J., Dewan, M., Lau, W., Li, M., Lin, H., Marayong, P., Ramey, N., Hager, G.D., Hoffman, B., Larkin, D., Hasser, C.: Navigating inner space: 3-d assistance for minimally invasive surgery. Robotics and Autonomous Systems **52** (2005) 5–26
6. Richa, R., Balicki, M., Meisner, E., Sznitman, R., Taylor, R., Hager, G.: Visual tracking of surgical tools for proximity detection in retinal surgery. In Taylor, R., Yang, G.Z., eds.: IPCAI 2011. Volume 6689 of LNCS. Springer, Heidelberg (2011) 55–66
7. Voros, S., Long, J.A., Cinquin, P.: Automatic detection of instruments in laparoscopic images: A first step towards high-level command of robotic endoscopic holders. International Journal of Robotic Research **26**(11-12) (2007) 1173–1190
8. Ali, K., Fleuret, F., Hasler, D., Fua, P.: A real-time deformable detector. Transactions on Pattern Analysis and Machine Intelligence **34**(2) (2011) 225–239
9. Benhimane, S., Malis, E.: Homography-based 2D Visual Tracking and Servoing. International Journal of Robotics Research **26**(7) (2007) 661–676
10. Pickering, M., Muhit, A., Scarvell, J., Smith, P.: A new multi-modal similarity measure for fast gradient-based 2d-3d image registration. In: Annual International Conference of the IEEE on Engineering in Medicine and Biology Society, 2009. (sept. 2009) 5821 –5824