Author Manuscript Author Manuscript Author Manuscript Author Manuscript

HHS Public Access
Author manuscript
Int J Comput Assist Radiol Surg. Author manuscript; available in PMC 2020 May 01.

Published in final edited form as:
Int J Comput Assist Radiol Surg. 2019 May ; 14(5): 819–828. doi:10.1007/s11548-019-01925-1.

EyeSAM: Graph-based Localization and Mapping of Retinal Vasculature during Intraocular Microsurgery

Shohin Mukherjee[1], **Michael Kaess**[1], **Joseph N. Martel**[2], **Cameron N. Riviere**[1]

[1]The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213 USA camr@ri.cmu.edu; +1(412)268-3083

[2]Department of Ophthalmology, University of Pittsburgh, Pittsburgh, PA 15213 USA

Abstract

Purpose: Robot-assisted intraocular microsurgery can improve performance by aiding the surgeon in operating on delicate micron-scale anatomical structures of the eye. In order to account for the eyeball motion that is typical in intraocular surgery, there is a need for fast and accurate algorithms that map the retinal vasculature and localize the retina with respect to the microscope.

Methods: This work extends our previous work by a graph-based SLAM formulation using a sparse incremental smoothing and mapping (iSAM) algorithm.

Results: The resulting technique, "EyeSAM," performs SLAM for intraoperative vitreoretinal surgical use while avoiding spurious duplication of structures as with the previous simpler technique. The technique also yields reduction in average pixel error in the camera motion estimation.

Conclusions: This work provides techniques to improve intraoperative tracking of retinal vasculature by handling loop closures and achieving increased robustness to quick shaky motions and drift due to uncertainties in the motion estimation.

Keywords

surgical robotics; simultaneous localization and mapping; retinal surgery; factor graphs

Introduction

Intraocular microsurgery is a delicate procedure. The presence of physiological tremor in the surgeon's hand makes it even more challenging. Success of promising new techniques like cannulation, which has the potential to be useful in the treatment of diseases such as retinal vasculature occlusion (RVO), depends on precise micromanipulation to inject anticoagulants into veins less than 100μm in diameter, which is less than the amplitude of physiological hand tremor [1,2]. RVO is the second most common retinal vascular disease after diabetic retinopathy and affects an estimated 16.4 million adults worldwide [3].

The authors declare that they have no conflict of interest. This article does not contain any studies with human participants or animals performed by any of the authors.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

The accuracy needed to safely and effectively treat RVO is not generally attainable with traditional instrumentation, however. This has led to the development of several robot-assisted vitreoretinal surgery systems, most of which use a teleoperative approach [4–6], while others follow a shared-control approach, in which the surgeon and a robot arm hold the surgical instrument simultaneously, the robot arm selectively complying in order to suppress unwanted motion [7–9].

A handheld approach is possible, however, by actuating the tip of a handheld instrument. "Micron" is an actively stabilized handheld robotic microsurgical instrument that compensates for the surgeon's hand tremor [10]. Micron is tracked in real time by an optical tracking system (Apparatus to Sense Accuracy of Position (ASAP) [11] while a pair of stereo cameras attached to a microscope observe the retinal plane to enable vision-based control, as shown in Fig. 1. During retinal microsurgery the patient is typically sedated rather than anaesthetized, which leads to unwanted movement of the eyeball. Moreover, the surgeon moves the eyeball to bring the region of interest in the field of view being observed by the microscope. Therefore, there is a need to track this intended and unintended eyeball motion and compensate for it during control.

In [12], Micron was used for automated laser photocoagulation. As shown in Fig. 2, the targets were defined in the image space. A hybrid control scheme was proposed in which the 3-DOF motion of the tool tip was decoupled into 2-DOF planar motion parallel to the retinal plane and 1-DOF motion along the axis of the tool. The decoupled 2-DOF motion was then controlled via image-based visual servoing, to locate the laser aiming beam on a target. To compensate for the eyeball motion, the blood vessels were tracked. In [13], a method for retinal vein cannulation using Micron was proposed. The targets were defined in the image coordinates, and a homography matrix was used to map the image space target onto the ASAP space. Here again, vasculature tracking was used to account for eyeball motion.

Becker and Riviere [14] introduced EyeSLAM, a real-time simultaneous localization and mapping (SLAM) algorithm for retinal vasculature, which employed the iterative closest point (ICP) algorithm for registration between a skeletonized version of the occupancy map and the current vessel observations. EyeSLAM was improved further in [15] by introducing the following two components:

- A fast correlative scan-matching method proposed by Olson [16], in place of ICP;
- More robust vessel detection with better rejection of spurious detections.

However, the algorithm in [15] employs scan matching for frame-by-frame registration, which corresponds to visual odometry. This leads to significant drift since the uncertainty in the motion model builds up over time. To address this need, this brief paper builds on [15] by exploring more advanced SLAM algorithms that handle loop closures to remove drift. The paper concludes with an illustrative quantitative example.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

Materials and Methods

Problem Definition

The retinal localization and mapping problem can be defined as follows. Given a series of input video frames $I = [I_0, I_1, \cdots, I_T]$, over a discretized time period $t \in [0, 1, \cdots, T]$, we want to obtain a global map in the form of vasculature points and the corresponding camera viewpoint locations, or poses, $X = [x_0, x_1, \cdots, x_T]$ of the input video frames in the map. The retina is modeled as a plane because typical retinal surgeries have high magnification with the field of view being only a few square millimeters in an eye 25 mm in diameter. The 3-DOF planar motion model is parameterized as a 2D translation and a rotation: $[t_x, t_y, \theta]$.

Graph Optimization

We take a probabilistic approach and represent the SLAM problem as a factor-graph optimization. A factor graph is a bipartite graph $F = (U, V, E)$, where $f_i \in U$ are factor nodes and $x_j \in V$ are variable nodes. Edges $e_{ij} \in E$ always exist between factors and variables (17). A factor graph defines the factorization of a global function $f(X)$ over the camera viewpoint locations X as:

$$f(X) = \prod_i f_i(X_i) \quad (1)$$

where X_i is the subset of the poses relevant to factor f_i.

Each factor $f_i(X_i)$ in the graph is proportional to a corresponding factor in the posterior probability function $p(X|Z)$, i.e. the posterior density of the states X given the measurements Z. Each variable in the graph corresponds to the pose of the retina at a certain instant. Solving for the poses involves performing maximum *a posteriori* (MAP) inference on the variables, given the information obtained from the uncertain measurements

$$X^{MAP} = \underset{X}{argmax} f(X) \quad (2)$$

$$X^{MAP} = \underset{X}{argmax} \prod_i f_i(X_i) \quad (3)$$

In our factor-graph formulation, the factors are of two types: motion factors between successive frames, obtained as explained below under “Motion Estimation via Scan Matching,” and loop closure factors, obtained as explained below under “Loop Closure Detection.” Each of these factors is of the following form:

$$f_i(X_i) \propto exp\left(\frac{-1}{2}\left\|h_i\left(x_{A_i}, x_{B_i}\right) - z_i\right\|^2_{\Sigma_i}\right) \quad (4)$$

where $x_{A_i} \in X_i$ and $x_{B_i} \in X_i$ are the two poses between which the factor exists. They are consecutive poses in the case of incremental motion factors, but they could be non-consecutive in the case of loop closure factors. Here h is the measurement function that calculates the difference between the two poses and Σ is an $n \times n$ covariance matrix. This assumes measurements are corrupted by zero-mean Gaussian noise. Therefore, performing MAP inference involves solving the following nonlinear least-squares optimization:

$$X^{MAP} = \underset{X}{argmin}\left(\|x_1\|^2_{\Sigma} + \sum_i \|h_i\left(x_{A_i}, x_{B_i}\right) - z_i\|^2_{\Sigma_i}\right) \quad (5)$$

The first term in the minimization arises from a unary factor on the first pose, which corresponds to our setting of the initial reference position to [0,0,0]. Each node in our graph contains a local map of the vascular tree that was observed in the pose that corresponds to that node. When a new image frame is obtained from the camera, first its vascular tree is extracted and the incremental motion relative to the previous frame is estimated using scan matching. A new node corresponding to the current frame is then added to the factor graph with a motion factor, shown in black in Fig. 3. Then loop closure is checked for. If an overlap with a previously mapped area is detected, a loop closure factor is added, as shown in red. The graph is incrementally optimized using the iSAM2 algorithm [18].

Feature Extraction

In visual odometry there are two categories of methods to construct an optical flow field. In the first category the frames are directly matched in a dense or semi-dense fashion [19,20]. This obviates feature extraction and exploits all the information present in the images. The second category of methods relies on feature detectors/descriptors such as SIFT [21], SURF [22] or other custom-designed detectors. Such detectors fail to find distinctive points on the textureless retina.

We use the network of blood vessels on the retina as features. They are extracted using the fast vessel-detection algorithm proposed by Can et al [23]. It is a highly efficient algorithm, suitable for real-time high-definition video. Its efficiency comes from direct processing on gray-level data without any preprocessing, and from processing only a minimally necessary fraction of pixels in an exploratory manner, avoiding low-level image-wide operations such as thresholding, edge detection, and morphological processing. To remove spurious detections of vessel-like structures such as the instrument tip or the light-pipe that is used for illumination, each potential vessel point undergoes a color test and a bloom proximity test [15]. The color test rejects pixels that are too dark or insufficiently red, while the bloom proximity test rejects vessel points that are too close to large white specular blooms in the

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

image. As in [15], EyeSAM is equipped with a filter that applies a mask on the dark area circling the visible retina area to mask out fringing effects caused by the microscope. Fig. 4 shows the vessel extraction in a real eye. The vessels undetected because of surgical instrument shadows in one frame will become visible eventually when the instrument moves. The observation will then be used to update the probability of a vessel being present at that location in the global map, as explained in subsequent sections.

Mapping via Occupancy Grids

Like in [15], a global occupancy map holds the current best estimate of all the observed vasculature. Each pixel in the map stores a score that is proportional to the probability of it being a vessel pixel. At each time instant, the current observations z_t are transformed to the map with the localization estimate x_t. For each vein point z_t^i, a fixed value is added to the map at the corresponding cell location, thereby increasing the probability of a vessel existing at that location. The occupancy map has a maximum allowed value in order to keep the scores bounded. In [15], a decay function was used to the decrease the probability of all grid cells, allowing vessels which have not been detected for a while to vanish. What this essentially means is that the part of the built map that is no longer consistent with the current observations is discarded. Though this approach generates a smoother map by artificially removing the drift, it does not solve the underlying cause of the drift, i.e., the accumulated gross error in the localization caused by the inaccurate motion estimate. Hence, even though the map appears smooth, the localization is still under gross error. The utility of EyeSAM is to accurately track the eyeball motion so that it can be compensated for during control. A smooth map is not of primary concern.

Motion Estimation via Scan Matching

To estimate the eyeball motion, a 3-DOF planar motion model is used. In the first formulation of the algorithm, the iterative closest point (ICP) algorithm was used for registration between a skeletonized version of the occupancy map and the current vessel observations [14]. Any rapid motion caused it to fail.

In the second formulation [15], ICP was replaced with the fast correlative scan-matching approach. The scan-matching was performed incrementally between the features observed in the current frame and a global map of features accumulated in the form of an occupancy map. It used a hierarchical approach in which a first scan on a low-resolution version of the map quickly finds an approximate solution and avoids local minima. This approximate solution is then used to initialize a search on the high-resolution map. To ensure speed of execution, at most 500 random vessel points are used for scan matching. In [15], a constant-velocity Kalman filter was used to smooth the localization estimation.

In this work we use a graph-based formulation to allow correction of the map upon loop closure, requiring a different strategy for scan matching. Matching against the global map results in corruption of the map caused by drift: when revisiting a previously observed part of the map, drift in the state estimate results in a duplication of structure in the map. In a graph-based solution, instead of matching against the previous map, one matches to the previous frame only, generating pairwise constraints between pose estimates. Upon

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

revisiting previously observed parts of the scene, one additionally matches against older, nearby frames to generate again pairwise constraints. But this time they serve as loop closures between the current frame and an older part of the trajectory. The graph optimization then corrects the trajectory and localizes the camera viewpoints, and one can re-render a corrected global map—though that is only needed for visualization, not for the SLAM algorithm itself.

However, in practice, this approach needs to be refined because noise in the vessel detection causes failure in scan matching. Features observed in one frame may not be observed in the one immediately following. Therefore we propose a hybrid approach which estimates motion factors by registering the current frame to a locally accumulated map, without significantly invalidating the independence assumption that a factor between two nodes should express the relationship between just the two nodes. We do this by using a decay function that reduces the probabilities that were added to the occupancy map every Nth frame by a scale y, where $y \in (0,1)$. A value of $y = 0$is the same as matching consecutive frames. A value of $\gamma = 1$ is the same as matching the current frame with a globally accumulated map as was done in [15]. The lower the value of y and the higher the value of N, the more accurate the model, but the less robust the motion estimation. This is because the probabilities that were added to the occupancy map due to previously observed frames decay by a larger fraction and more frequently. In our implementation, y was chosen to be 0.9 and N was chosen as 10, based on preliminary experimentation.

Loop Closure Detection

If the pose estimate from incremental scan-matching is in gross error, while the camera is observing an already mapped region of the retina, the likelihood of the measurements being explained by the pose and map estimate is vanishingly small. This leads to previously visited areas getting re-mapped in the wrong global location and the error accumulates without bound [24]. Therefore loop closure factors are added to the graph when the camera re-observes a previously observed scene. In order to estimate loop closure, the local map of the current node needs to be compared to the local maps of every node that has been previously added to the group. Therefore the number of times we need to run the expensive scan-matching increases linearly with the graph size. This is too costly given the real-time requirements of the application. In order to save computation time, a better approach is to compare the current node with the nodes that are inside its uncertainty ellipse. However, this may still be very computationally expensive, especially if all frames are processed, in which case there will be multiple nodes within the uncertainty ellipse. Therefore we always match with a fixed number of previously visited nodes, depending upon how many threads can run in parallel in the given machine. In order to ensure that nodes recently visited are not matched with the current node, and loop constraints are added to the graph, we enforce a minimum time difference between the current frame and the frame it is matched with for a loop constraint.

Evaluation

A. Qualitative Evaluation—We evaluated EyeSAM on several videos of real human eye surgeries. These videos are of different surgical procedures including retinal membrane

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

peeling and laser photocoagulation. They were shot under varying conditions of lighting, resolution, and tool occlusion. There is no ground truth available for real surgery videos; hence a qualitative evaluation was made. In these experiments we processed all frames offline so that we can compare the performance with the old EyeSLAM. Hence the code was not optimized to run in real time. Real-time performance can easily be achieved by running the algorithm on a faster processor with more cores.

We also carried out qualitative evaluation of EyeSAM in an eye phantom. The eye phantom consisted of a network of blood vessels printed on paper at a scale that matches a real retina, similar to that used in [15]. We call this video sequence A (duration 75 s). All videos for the experiments in this paper were taken from the same monocular camera mounted on a high-magnification microscope. Our approach is expected to transfer to other devices, as long as the image scale is estimated. The camera plane is assumed to be parallel to the retinal plane.

B. Accuracy on Phantom—In order to better test our hypothesis that the factor-graph-based EyeSAM performs better, we carried out quantitative evaluation on another video sequence of the eye phantom. This was not possible to do on real surgery videos because no ground truth is available or can be generated for them. We call this video sequence B (duration 135 s). Ground truth was generated by tracking colored fiducials printed on the retina phantom.

Results

Qualitative Evaluation

The maps developed on the real eye surgery videos are shown in Fig. 5. Fig. 6(a) and 6(b) show the old EyeSLAM and the current EyeSAM working on video sequence A. The colored image on the left is the map superimposed on the image frame; the image on the right is the global occupancy map. The brighter the pixel, the more likely a vessel exists at that point. The drift due to the inaccurate motion estimation is clearly visible in Fig. 6(a). The new EyeSAM takes into account loop closure and therefore eliminates the drift, as expected, and as seen in Fig. 6(b). Fig. 8 shows the trajectory of the center of the camera frame with respect to the global map for video sequence A. The camera executes a loop and returns close to the starting position. The loop is detected and the drift in the motion is corrected as seen in Fig. 6(b). Fig. 7(a) and 7(b) compare the older EyeSLAM and the new EyeSAM working on a real surgery video. The map in Fig. 7(b) is sharper because of less drift as compared to the map in Fig. 7(a).

Accuracy on Phantom

For video sequence B, Fig. 9 compares the camera trajectory for the old EyeSLAM and the current EyeSAM with the ground truth trajectory. Fig. 10 compares the SLAM maps, where drift is apparent for the old EyeSLAM, while our implementation of factor-graph-based EyeSAM estimates camera motion much more accurately. The average error in translation was 18.8 pixels for the old EyeSLAM and 6.8 pixels for EyeSAM. Taking camera scale into account, which was calibrated separately, this corresponds to an error of 161.4 μm for the old EyeSLAM and 58.0 μm for EyeSAM. Therefore our method will allow us to safely

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

operate on vessels of 100 μm. The max error in translation was 59.5 pixels (511.0 μm) for the old EyeSLAM and 32.8 pixels (281.4 μm) for EyeSAM. From Fig. 9, drift is apparent in the old EyeSLAM, while EyeSAM estimates camera motion more accurately. EyeSLAM manages to correctly localize eventually, because we are matching against a globally accumulated map. However, it does not correct for the previously observed poses and the duplicated structures in the map. Fig. 11 compares the translation (in x and y) and rotation for the two cases with the ground truth.

Discussion

In this work we introduced EyeSAM, a factor-graph-based formulation of retinal SLAM that can incorporate loop closures and therefore eliminate the drift that is introduced by noise in the frame-by-frame scan matching. In this work we ignored intra-operative retinal deformation. Heart pulse and breathing effects are also ignored and a planar motion-model for the retina is assumed like in [15]. We evaluated our new algorithm on several real eye surgery videos under different lighting conditions and in the presence of tool occlusion. As can be seen from Fig. 5, our approach handles a wide range of pathologies and procedures. In case of retinal hemorrhage, the vessel extraction is likely to detect intraoperative retinal hemorrhages as vessels. Vitreous hemorrhage and other vitreous media opacities are cleared prior to retinal treatment or surgical maneuvers, as is standard practice for vitreoretinal surgery. The color test and the bloom proximity test, as explained under Feature Extraction, reject many false positives in the vessel detection. We also evaluated EyeSAM on eye-phantom videos and compared the generated map with that generated from the old EyeSLAM. EyeSAM generated a more consistent map. We also showed reduction in average pixel error in the camera motion estimation.

Future improvements include optimization to allow the algorithm to run in real time on high-definition videos. In addition, implementing 3D SLAM using stereo vision while handling distortion could be beneficial.

Acknowledgments

This study was partially funded by the U.S. National Institutes of Health (grant no. R01EB000526).

References

1. Weiss JN, Bynoe LA (2001) Injection of tissue plasminogen activator into a branch retinal vein in eyes with central retinal vein occlusion. Ophthalmology 108(12):2249–57 [PubMed: 11733266]
2. Bynoe LA, Hutchins RK, Lazarus HS, Friedberg MA (2005) Retinal endovascular surgery for central retinal vein occlusion: initial experience of four surgeons. Retina 25:625–32 [PubMed: 16077361]
3. Rogers S, McIntosh RL, Cheung N, Lim L, Wang JJ, Mitchell P, Kowalski JW, Nguyen H, Wong TY (2010) The prevalence of retinal vein occlusion: pooled data from population studies from the United States, Europe, Asia, and Australia Ophthalmology 117(2):313–319.e1 [PubMed: 20022117]
4. Nambi M, Bernstein PS, Abbott JJ (2016) A compact telemanipulated retinal-surgery system that uses commercially available instruments with a quick-change adapter. J Med Robot Res 1(2): 1630001

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

5. Nasseri MA, Eder M, Eberts D, Nair S, Maier M, Zapp D, Lohmann CP, Knoll A (2013) Kinematics and dynamics analysis of a hybrid parallel-serial micromanipulator designed for biomedical applications. In: Proc. IEEE/ASME Int Conf Adv Intell Mechatronics pp. 293–9
6. Wilson JT, Gerber MJ, Prince SW, Chen CW, Schwartz SD, Hubschman JP, Tsao TC 2018 Intraocular robotic interventional surgical system (IRISS): Mechanical design, evaluation, and master-slave manipulation. Int J Med Robot Comput Assist Surg 14(1):e1842
7. Taylor R, Jensen P, Whitcomb L, Barnes A, Kumar R, Stoianovici D, Gupta P, Wang z, Dejuan E, Kavoussi L (1999) A steady-hand robotic system for microsurgical augmentation. Int J Rob Res 18(12):1201–10
8. Üneri A, Balicki MA, Handa J, Gehlbach P, Taylor RH, Iordachita I (2010) New steady-hand eye robot with micro-force sensing for vitreoretinal surgery. In: Proc 3rd IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics pp. 814–9
9. Willekens K, Gijbels A, Schoevaerdts L, Esteveny L, Janssens T, Jonckx B, Feyen JH, Meers C, Reynaerts D, Vander Poorten E, Stalmans P (2017) Robot-assisted retinal vein cannulation in an in vivo porcine retinal vein occlusion model. Acta Ophthalmol 95(3):270–5 [PubMed: 28084059]
10. Yang S, MacLachlan RA, Riviere CN (2015) Manipulator design and operation for a six-degree-of-freedom handheld tremor-canceling microsurgical instrument. IEEE/ASME Trans Mechatron 20(2):761–72
11. MacLachlan RA, Riviere CN (2009) High-speed microscale optical tracking using digital frequency-domain multiplexing. IEEE Trans Instrum Meas 58(6):1991–2001 [PubMed: 20428484]
12. Yang S, Martel JN, Lobes LA Jr, Riviere CN (2018) Techniques for robot-aided intraocular surgery using monocular vision. Int J Rob Res 37(8):931–952 [PubMed: 30739976]
13. Mukherjee S, Yang S, Maclachlan RA, Lobes LA, Martel JN, Riviere CN (2017) Toward monocular camera-guided retinal vein cannulation with an actively stabilized handheld robot. In: Proc IEEE Int Conf Robot Autom p. 2951–6
14. Becker BC, Riviere CN (2013) Real-time retinal vessel mapping and localization for intraocular surgery. In: Proc IEEE Int Conf Robot Autom p. 5346–51
15. Braun D, Yang S, Martel JN, Riviere CN, Becker BC (2018) EyeSLAM: real-time localization and mapping of retinal vessels during intraocular microsurgery. Int J Med Robot Comput Assist Surg 14:e1848
16. Olson EB (2009) Real-time correlative scan matching. In: Proc IEEE Int Conf Robot Autom pp. 4387–93
17. Dellaert F, Kaess M (2017) Factor graphs for robot perception. Foundations and Trends in Robotics 6(1– 2):1–139
18. Kaess M, Johannsson H, Roberts R, Ila V, Leonard JJ, Dellaert F (2012) iSAM2: Incremental smoothing and mapping using the Bayes tree. Int J Rob Res 31(2):216–35
19. Engel J, Sturm J, Cremers D (2013) Semi-dense visual odometry for a monocular camera. In: Proc IEEE Int Conf Comput Vision p. 1449–56
20. Engel J, Schöps T, Cremers D (2014) LSD-SLAM: Large-scale direct monocular SLAM. In: Proc Eur Conf Comput Vision p. 834–49
21. Lowe DG (1999) Object recognition from local scale-invariant features. In: Proc Int Conf Comput Vision 2:1150–7
22. Bay H, Ess A, Tuytelaars T, van Gool L (2008) Speeded-Up Robust Features (SURF). Comp Vis Image Understanding 110(3):346–59
23. Can A, Shen H, Turner JN, Tanenbaum HL, Roysam B (1999) Rapid automated tracing and feature extraction from live high-resolution retinal fundus images using direct exploratory algorithms. IEEE Trans Inf Technol Biomed 3(2):125–38 [PubMed: 10719494]
24. Newman P, Ho K (2005) SLAM-loop closing with visually salient features. In: Proc IEEE Int Conf Robot Autom p. 635–42

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

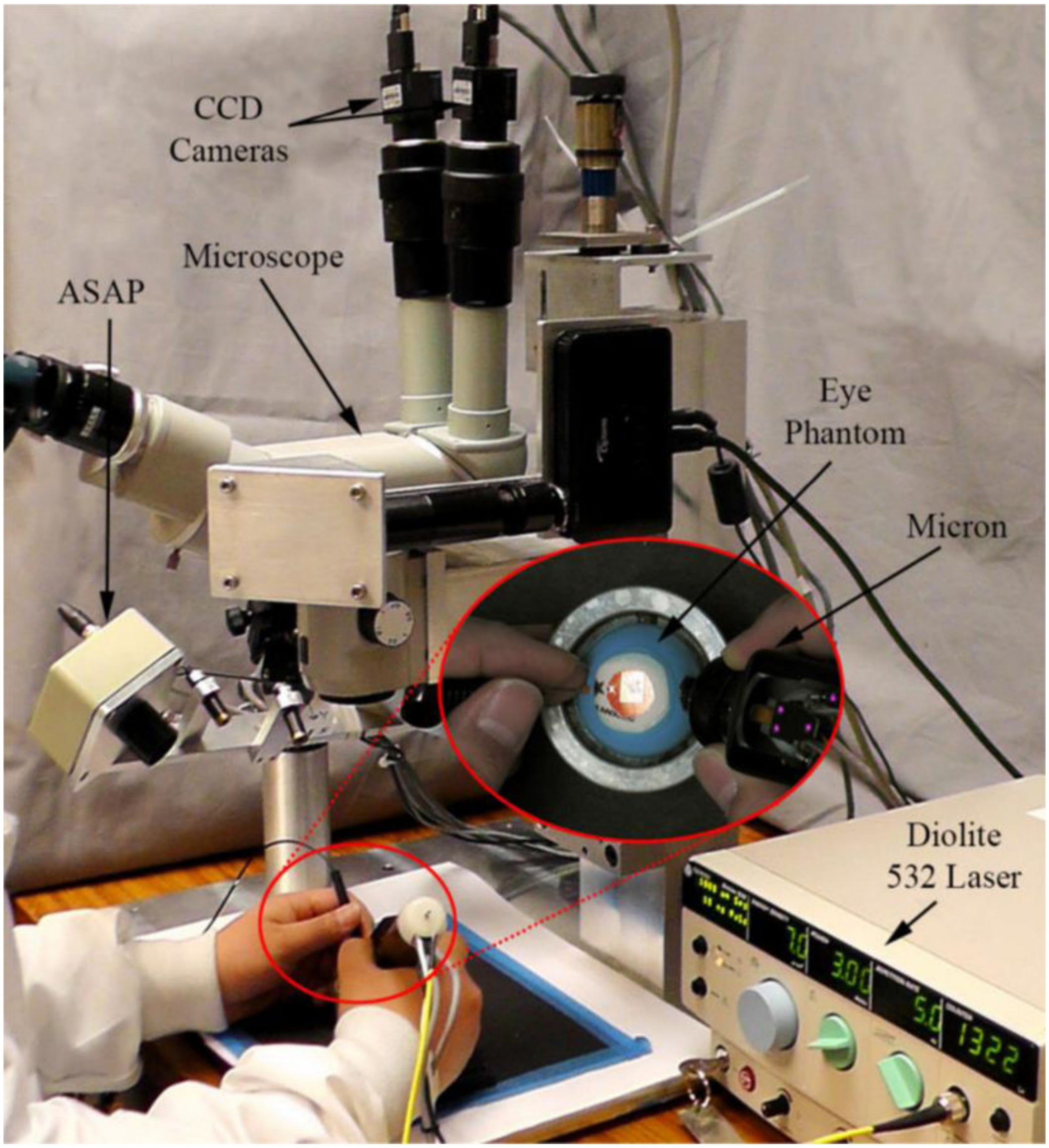


Figure 1.
Micron setup. The main components have been labeled: the optical tracking system (Apparatus to Sense Accuracy of Position, or “ASAP”), CCD cameras, microscope, and laser.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

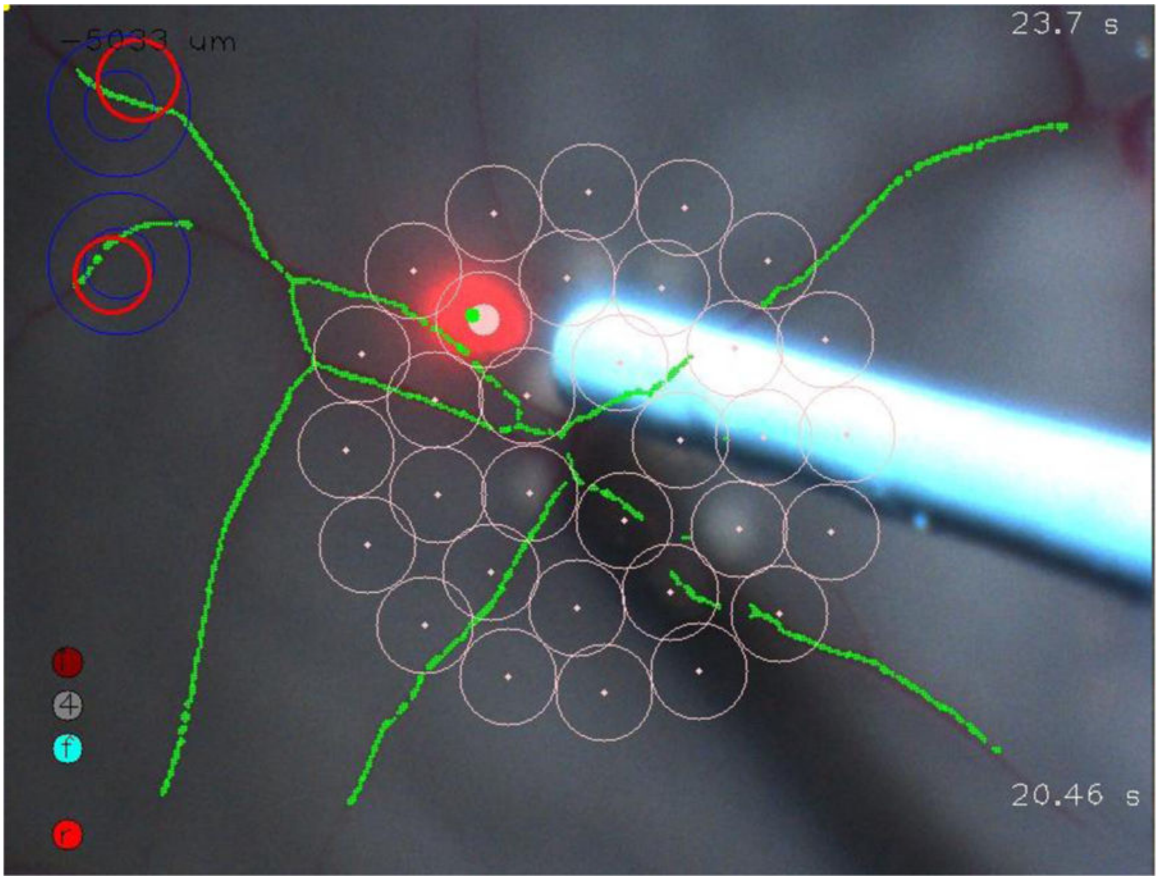


Figure 2.
Micron-aided photocoagulation in porcine retina *ex vivo*.

Author Manuscript

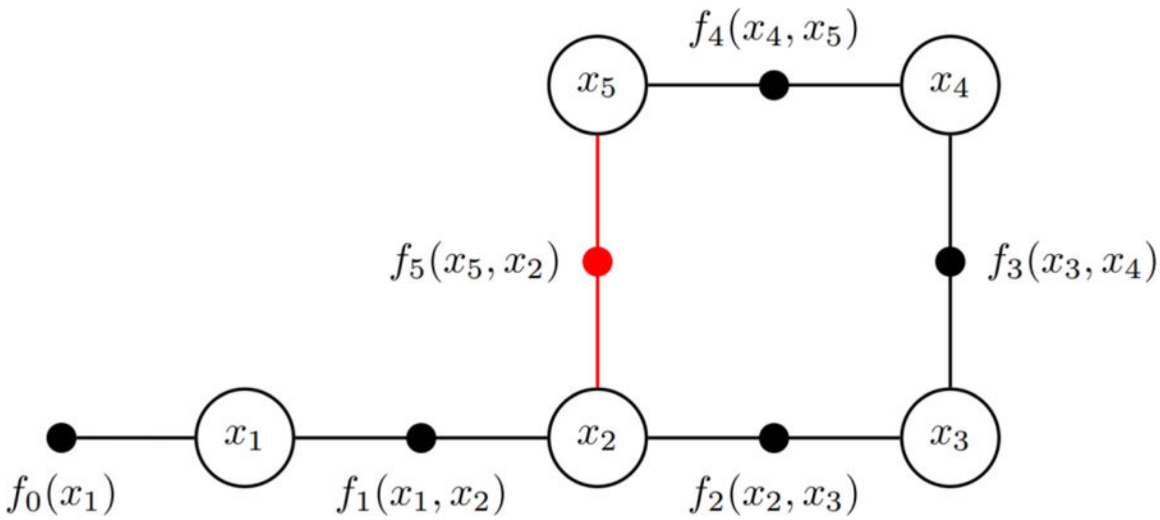


Figure 3.

Factor graph representation [17] of EyeSAM. The unfilled circles correspond to the variables over which we intend to infer. The incremental motion constraints are shown in black. The loop closure constraints are shown in red. $f_0(x_1)$ is a prior factor to fix a coordinate frame.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

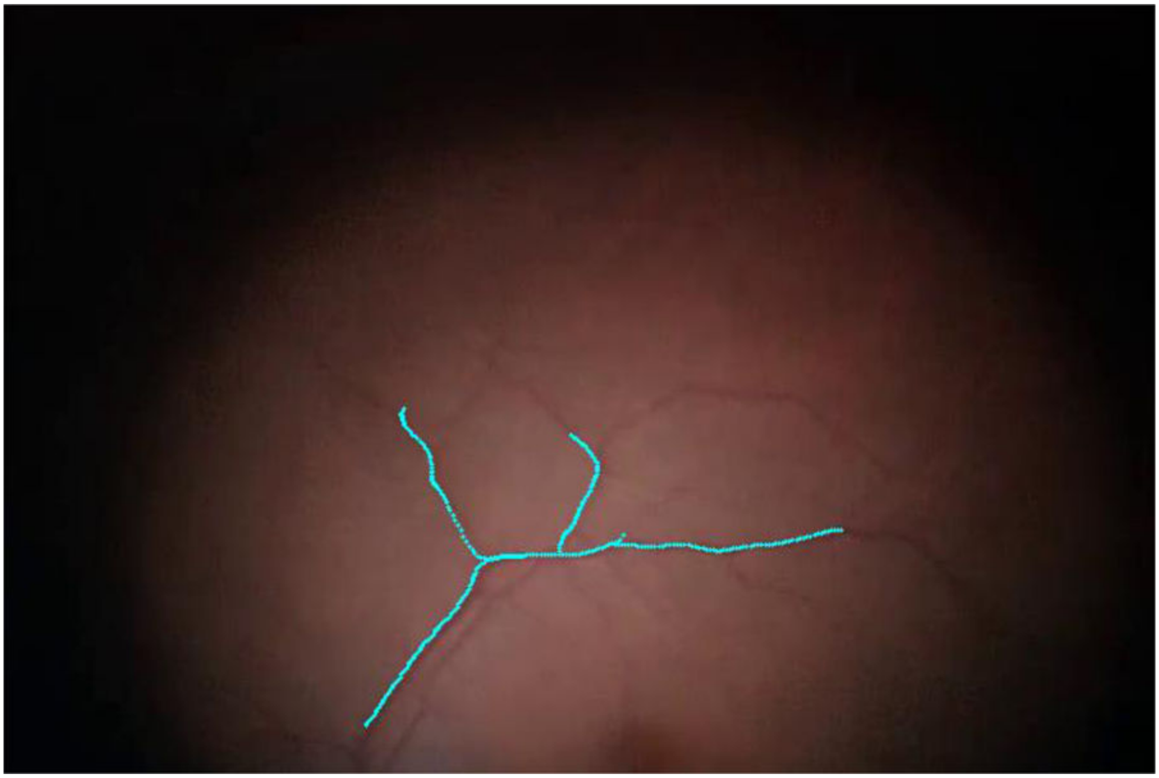


Figure 4.
Vessel extraction in a real eye using the algorithm proposed in [23]. The algorithm detects the major vessels. The thinner vessels are harder to detect, but once they are detected in subsequent frames, they will be added to the global map.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

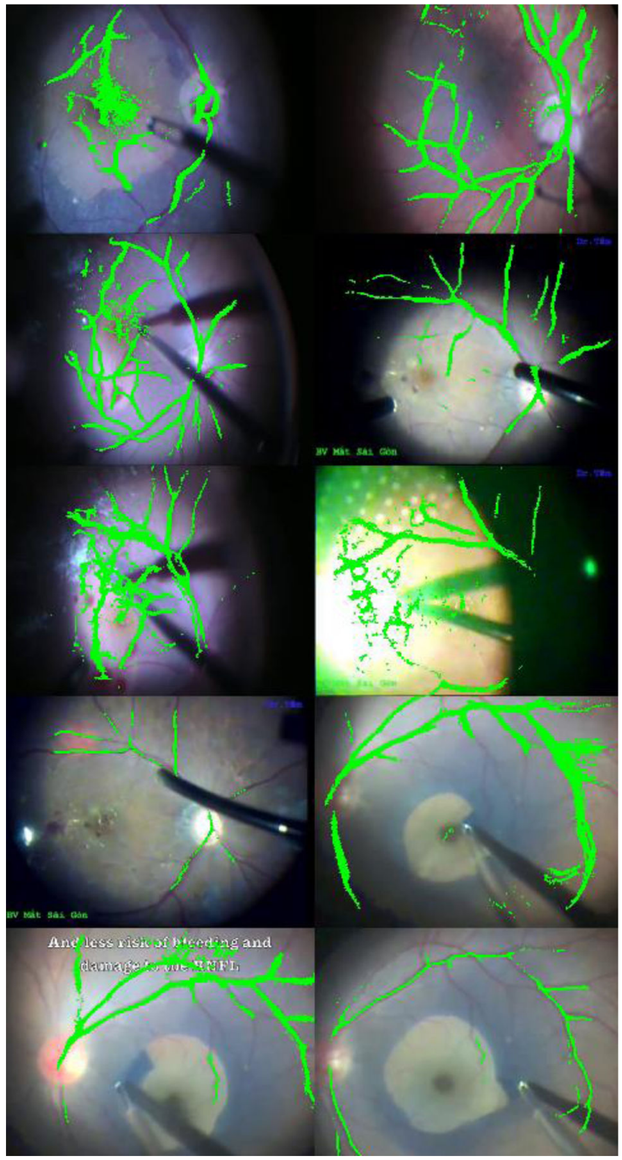
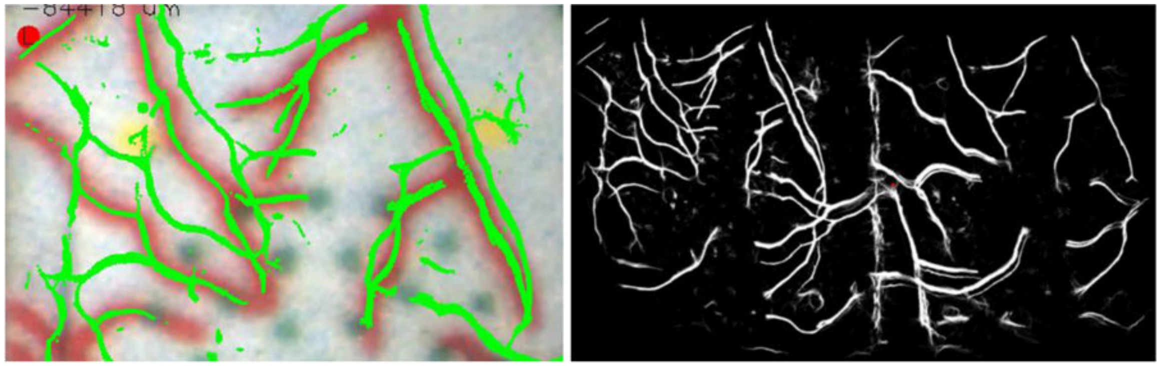
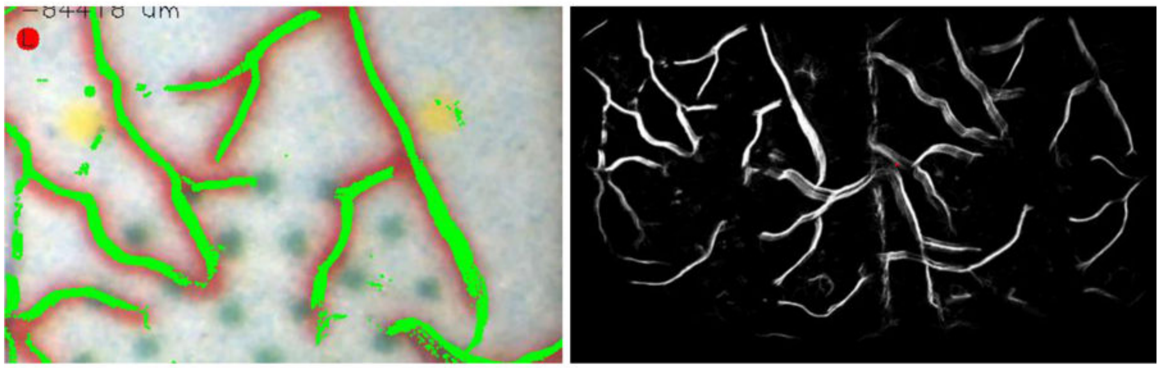


Figure 5.

EyeSAM was run on several video sequences of eye surgery. The videos correspond to different surgical procedures, under different lighting conditions. It can be seen that EyeSAM manages to map most of the vasculature. It also handles tool occlusions well.



(a)



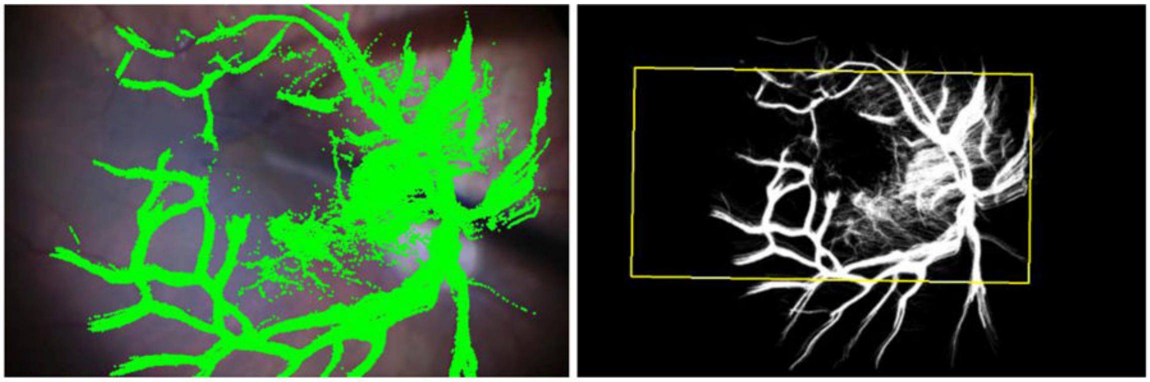
(b)

Figure 6.

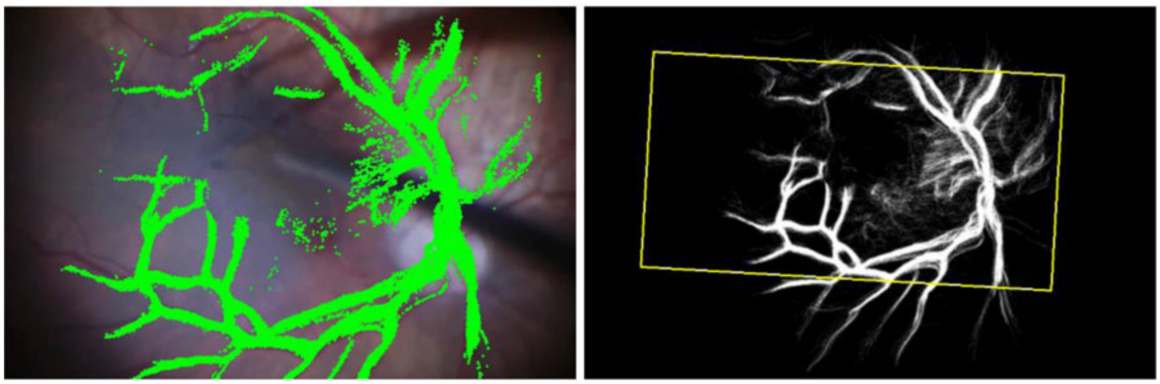
Comparison of SLAM maps generated from (a) old EyeSLAM and (b) factor-graph-based EyeSAM on video sequence A. The images on the right show the global map. The images on the left show the map superimposed on the current frame. The drift is apparent in old EyeSLAM which leads to duplicated structures; this has been corrected in EyeSAM.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

Author Manuscript Author Manuscript Author Manuscript Author Manuscript



(a)



(b)

Figure 7.
Comparison of SLAM maps generated from (a) old EyeSLAM and (b) factor-graph-based EyeSAM using video from clinical vitreoretinal surgery.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

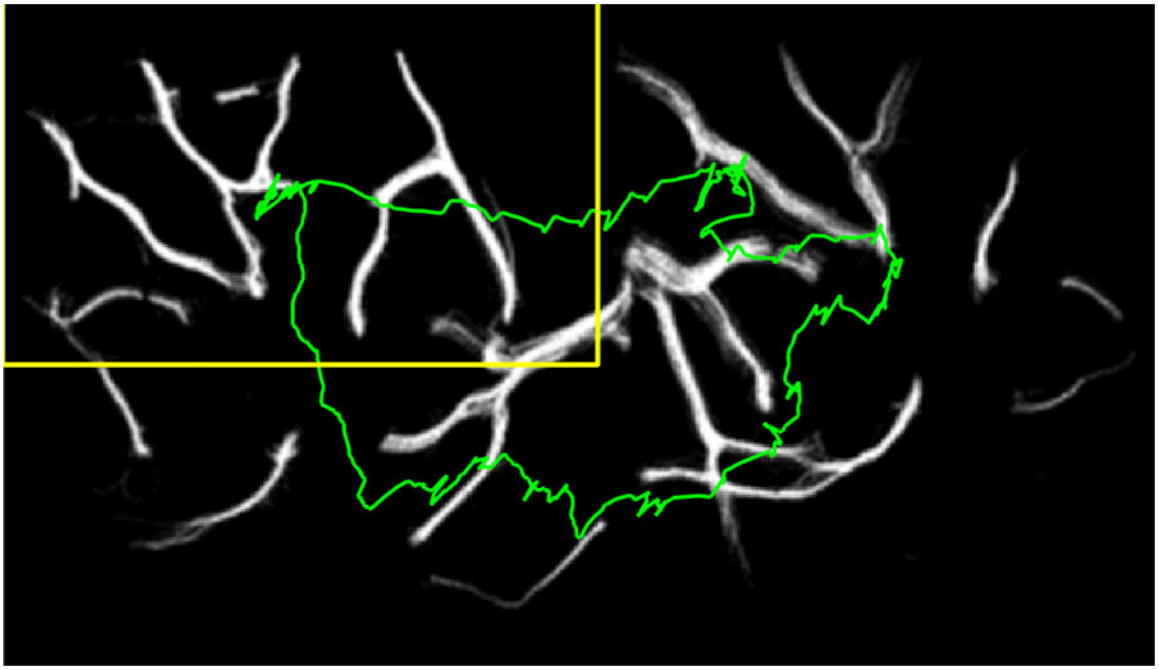


Figure 8.
Camera trajectory (green) superimposed on the globally corrected map for video sequence A. The yellow frame shows the footprint of the last image, which completed a loop in the camera trajectory.

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

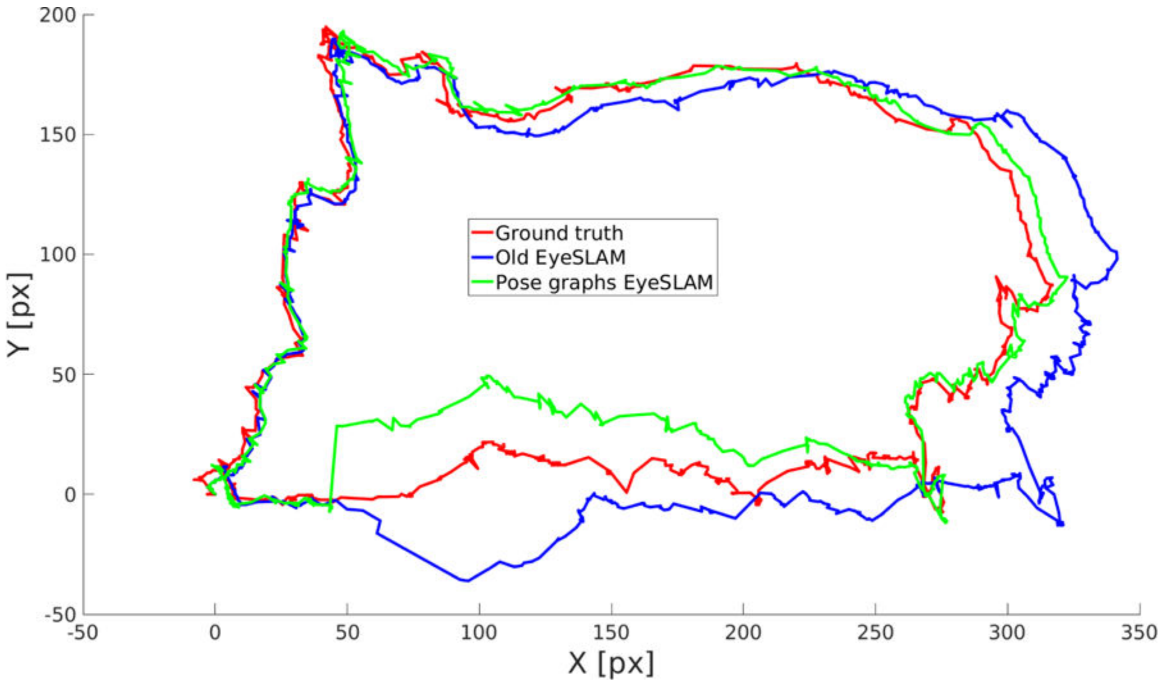
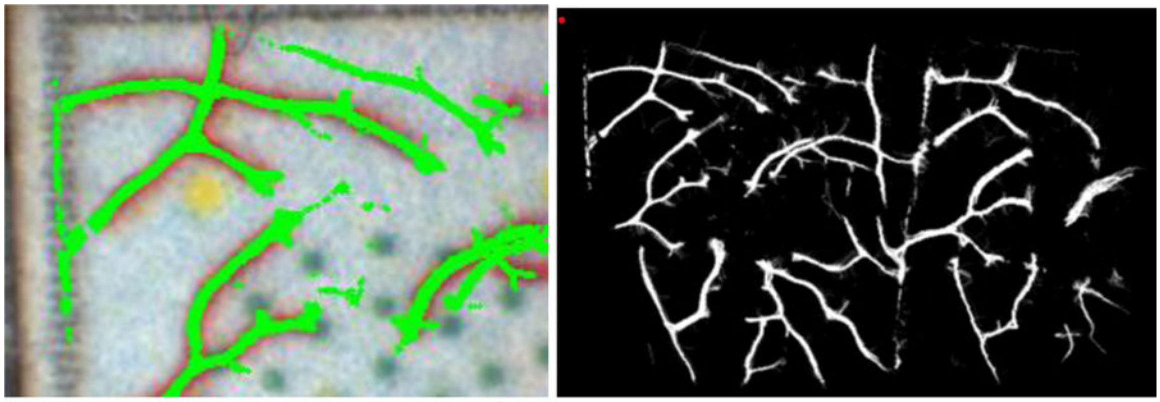


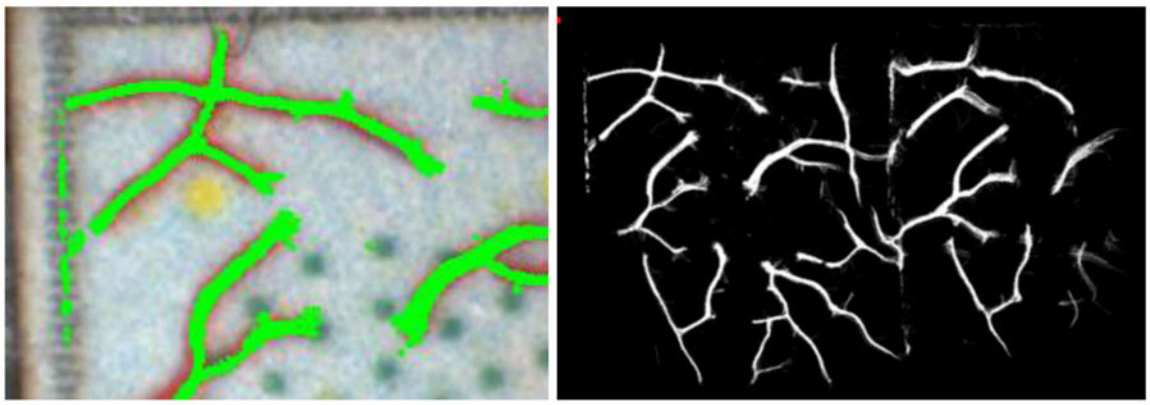
Figure 9.

Camera trajectory for ground truth (red), old EyeSLAM (blue), and factorgraph-based EyeSAM (green) on video sequence B.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript



(a)



(b)

Figure 10.
Comparison of SLAM maps generated from (a) old EyeSLAM and (b) new EyeSAM on video sequence B.

Author Manuscript
Author Manuscript
Author Manuscript
Author Manuscript

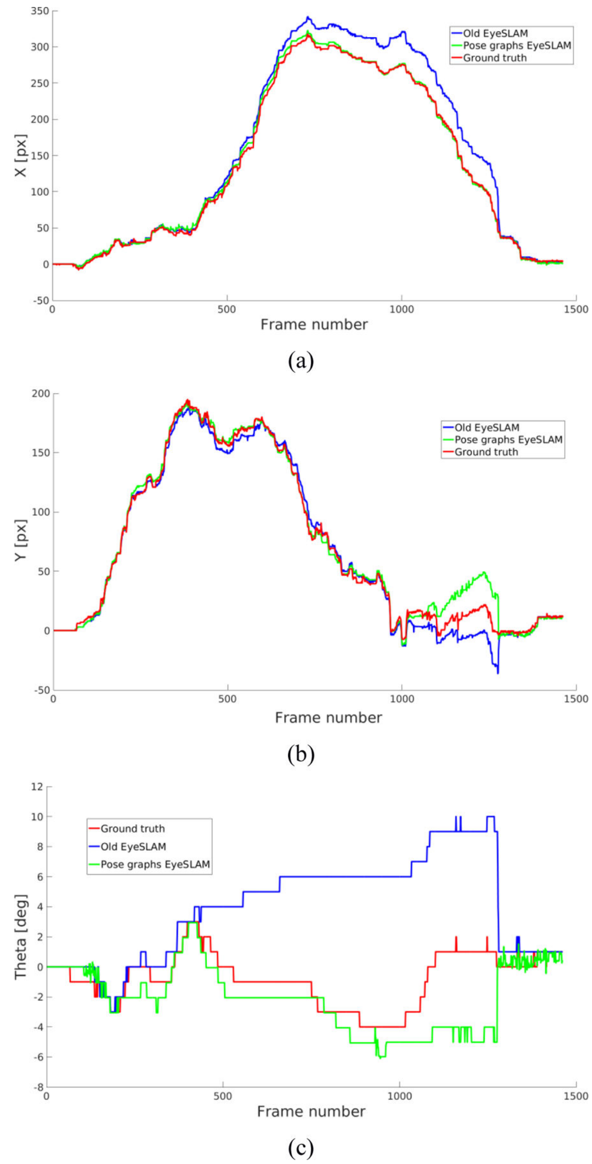


Figure 11.
Comparing translation and orientation for ground truth (red), old EyeSLAM (blue), and factor-graph-based EyeSAM (green). (a) Translation in *x*. (b) Translation in *y*. (c) Rotation.