Optimization Methods for Tracking and Mapping the Human Retina

Jay Shenoy
Ren Ng, Ed.
Austin Roorda, Ed.

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2022-159
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2022/EECS-2022-159.html

May 20, 2022

Acknowledgement

I would like to thank Professor Ren Ng, Professor Austin Roorda, James Fong, and the rest of my friends and family.

Optimization Methods for Tracking and Mapping the Human Retina

by

Jay Shenoy

A thesis submitted in partial satisfaction of the

requirements for the degree of

Master of Science

in

Electrical Engineering and Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Ren Ng, Chair
Professor Austin Roorda

Spring 2022

Optimization Methods for Tracking and Mapping the Human Retina

by Jay Shenoy

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

Committee:

Professor Ren Ng
Research Advisor

May 20, 2022

(Date)

* * * * * * *

Professor Austin Roorda
Second Reader

May 19, 2022

(Date)

Optimization Methods for Tracking and Mapping the Human Retina

Abstract

Optimization Methods for Tracking and Mapping the Human Retina

by

Jay Shenoy

Master of Science in Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Ren Ng, Chair

The human retina contains a mosaic of light-sensitive photoreceptor cells that capture visual stimuli. Determining the structure of the retina is important for ophthalmology and vision science, as well as for emerging display technologies that operate at the cellular level. Adaptive optics scanning laser ophthalmoscopy (AOSLO) and optical coherence tomography (AO-OCT) are two techniques for imaging the retina at high resolution in 2D and 3D, respectively. Both techniques scan the eye over a set period of time, and as a result produce images that contain distortions arising from the motion of the eye during acquisition. Prior methods for AOSLO and AO-OCT image processing have been reference-based, generating maps of the retina by registering the acquired image frames against a motion-corrected reference image and averaging these frames together. These methods rely on heuristics for estimating eye motion, resulting in inaccurate motion traces that engender mapping artifacts. In this work, we introduce an optimization-based framework for retina tracking and mapping that directly solves for the most likely trace of eye motion that occurred during the recording session. Our framework, R-SLAM, uses inter-frame feature correspondences and a novel convex optimization algorithm to compute the optimal motion solution, where optimality is defined in a probabilistic sense. By directly solving the inverse problem of calculating the likeliest map and motion trace that gave rise to the retina recording, we produce retina maps of higher quality than those found in prior work. This report includes research on AOSLO image processing that was part of a recent publication as well as subsequent work on applying R-SLAM to the AO-OCT regime that was conducted during the master's program. R-SLAM's success in both the AOSLO and AO-OCT domains indicates its utility as a theoretical and practical foundation that opens up new avenues for research on optimization-based retinal image processing.

To my family

Contents

List of Figures

List of Tables

Acknowledgments

I would first like to thank my advisor, Professor Ren Ng. I first met Ren as a student in his computer graphics course and was immediately captivated by his passion for the field of visual computing, both from a scientific and an aesthetic standpoint. Ren was gracious enough to take me on as a research assistant, and over the past several years, we have had so many fruitful discussions about science, entrepreneurship, and technical communication. More than anything, Ren instilled in me an enduring sense of the true value of any given aspect of my work, be it the minutiae of figure formatting or the broader narrative flow of a scientific argument. I am grateful to have had such a fine mentor during my time at Berkeley.

Much of the work presented here was done jointly with James Fong. As with Ren, I met James while enrolled in computer graphics, for which he served as a teaching assistant. The first time we met, James was kind enough to spend hours debugging one of my course projects, and after several years of working together, I can safely admit that this dynamic has not changed all that much. I have learned so much from James about the theory and practice of computer science through many discussions and hours of building software together, and I am fortunate to have him as a friend.

I would like to thank my friends outside the lab for their unending support and for providing a higher-level perspective on the purpose of research and work in general, which has been invaluable in guiding my own thinking as I finish this chapter at Berkeley. I am so lucky to be surrounded by such a talented, compassionate group of friends, and I wouldn't be where I am today without them.

This work was supported by a Hellman Fellowship, by the Air Force Office of Scientific Research under award numbers FA9550-20-1-0195 and FA9550-21-1-0230, and by National Institutes of Health (NIH) grant R01EY023591.

Lastly, I am grateful to my family – Rajesh, Sheetal, and Esha – for their unwavering support and love throughout my life. They have encouraged me wholeheartedly in each of my endeavors and taught me to never limit the scope of my dreams.

Chapter 1

Introduction

The human retina contains a mosaic of photoreceptor cells that capture light. The purpose of retinal imaging is to determine the visual structure of the retina at the cellular scale, which is important for ophthalmology and vision science. To that end, adaptive optics scanning laser ophthalmoscopy (AOSLO) is an imaging technique that has found success in capturing high-resolution, two-dimensional video recordings of the retina in vivo. In the three-dimensional setting, adaptive optics optical coherence tomography (AO-OCT) produces volumes that enable depth-wise inspection of the different layers of the retina. Both AOSLO and AO-OCT acquire images of the retina by scanning the eye over a set period of time. Figure 1.1 shows an AOSLO tabletop setup, and figures 1.2 and 1.3 show typical captures from AOSLO and AO-OCT systems.

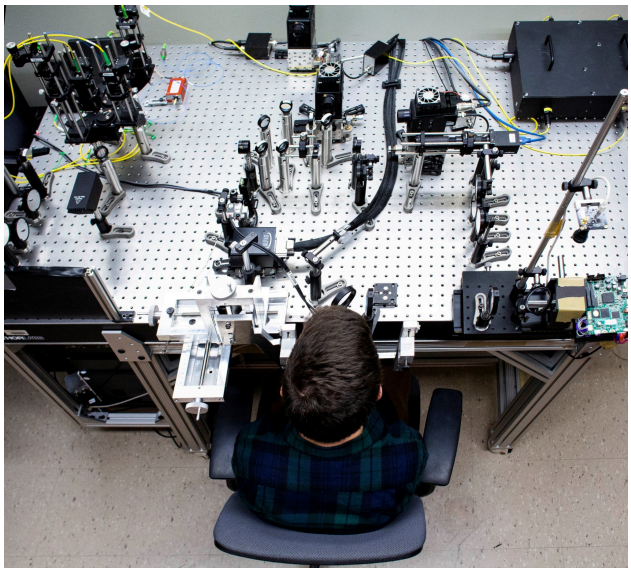


Figure 1.1: A subject conducting an imaging session with the AOSLO system. Photo credit: Elena Zhukova.

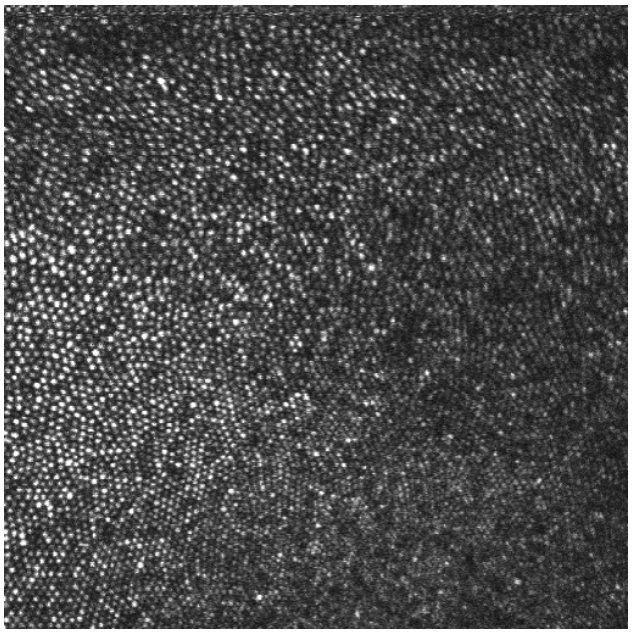


Figure 1.2: A typical frame captured by an AOSLO system, which possesses enough resolution to identify individual cones in the image. The dots (both light and dark) correspond to individual cells. The shearing and vertical elongation of the cone cells in the top part of the frame suggests that the eye was undergoing a microsaccade directed down and to the right when the top portion was scanned.

1.1 Retina Tracking and Mapping as Interdependent Problems

There are two main problems we wish to solve when performing offline processing of AOSLO or AO-OCT data: either (1) track the eye's motion over the course of the recording or (2) generate an accurate map of the retina. These two problems are interdependent in that solving one of them allows one to easily solve the other. That is, given an accurate motion trace of the retina during the recording, one can stabilize and average each frame to produce a retina map. Conversely, given a retina map, one can register portions of the recorded frames against the map to retrieve an estimate of the eye's motion. Unfortunately, the solution space is underconstrained as there are (infinitely) many different map and motion pairs that could give rise to the same retina recording, as illustrated by figure 1.4. Solving for the map therefore requires perfect knowledge of the motion and vice versa, which, combined with the infinite solution space, creates a chicken-and-egg problem with no definite solution.

We make several assumptions to resolve the underconstrained nature of the problem setup. First, we assume that the eye's motion cannot exceed a certain speed, which, in our empirical studies, always holds because we analyze retina recordings acquired while the subject is asked

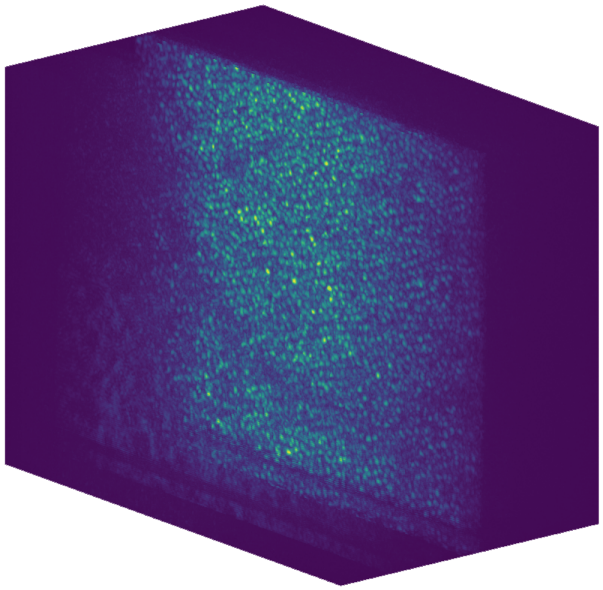


Figure 1.3: A single volume captured via AO-OCT, visualized with volume rendering to show three-dimensional structure. The blue-green plane corresponds to the photoreceptor layer, and the purple region is the enclosing volume.

to fixate steadily on a target. Second, we assume that while the retina exhibits a semi-regular arrangement of photoreceptor cells, different regions of cones remain visually distinct from one another under some metric. Together, these assumptions preclude the solution ambiguity found in figure 1.4. The second assumption, in particular, permits the use of feature tracking algorithms to derive information about the motion of cone regions over time. This sort of feature tracking is standard across prior methods for retina tracking and mapping as well as the techniques we introduce in this work.

1.2 Prior Work

Prior methods for retina tracking and mapping mostly operate under the same principle, which is to designate a single frame as the reference and register all the other frames in the recording against the reference. The registered frames are averaged to produce a retina map with a higher signal-to-noise ratio than any individual frame, and sub-frame registration offsets provide estimates of eye motion. The problem with this approach is that the eye is constantly moving during the acquisition of each frame because of the rolling shutter nature of the imaging systems, so the approach fails to correct for distortion in the reference itself. Alternative methods [2, 4] have been proposed to perform intra-frame distortion correction, but they are not always accurate because they assume that the eye's motion is of zero-mean over the course of the entire recording, which fails to hold for longer recordings. Even a few pixels of inaccuracy in the motion-corrected reference can

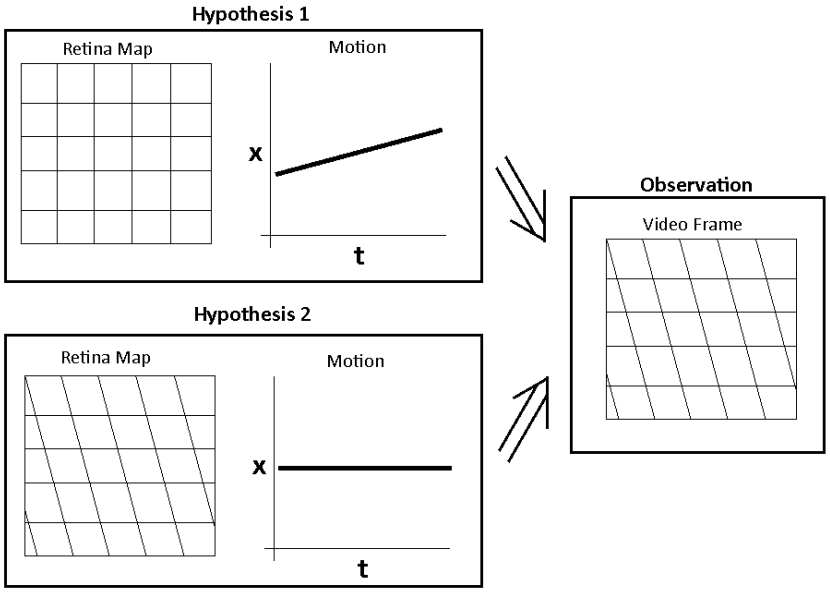


Figure 1.4: An example of two distinct motion and map pairs giving rise to the same observation. In the top pair (hypothesis 1), the retina is a rectilinear grid undergoing constant-velocity rightward motion during the image acquisition, resulting in a slanted observation. In hypothesis 2, the retina is a slanted grid undergoing zero motion, resulting in the same observed frame.

cause noticeable mapping artifacts when registering and stitching several other frames with the reference. A more complete discussion on prior methods and their limitations can be found in subsequent chapters.

1.3 An Optimization-Based Approach to Retina Tracking and Mapping

The main contribution of this work is to frame the recovery of retinal appearance and motion as an optimization problem. The novelty of this approach is that it simultaneously estimates the map and motion that produced a given retina recording, which is commonly referred to as an inverse problem in the field of computational imaging. Here, the optimization is done over the space of all map and motion pairs. At a high level, the optimization objective aims to determine the most likely map and motion pair giving rise to the recording, where likelihood is defined in terms of a probabilistic prior on eye motion as well as a feature-based penalty. This optimization framework, which we name retina-based simultaneous localization and mapping (R-SLAM), is general enough to be applied to both AOSLO and AO-OCT imagery.

Two versions of R-SLAM, one for AOSLO and another for AO-OCT, are described in the rest of this report, which is organized as follows:

- Chapter 2 describes R-SLAM for 2D AOSLO Imaging, a method for retrieving the retina's appearance and motion from AOSLO video. [1] Practically, we perform retina tracking and mapping in two steps: (1) tracking features in a dense fashion across the acquired frames followed by (2) solving a convex optimization objective that produces a motion trace fitting the tracked features as closely as possible while also obeying a chosen prior on eye motion. The output motion trace is used to generate a retina map by stabilizing and averaging each AOSLO frame. Finally, we perform a gradient-based refinement step for artifact correction. The advantage of this optimization setup is its generality in allowing the user to implement the feature tracking techniques and motion priors of their choice. We present robust feature tracking methods that are tailored to retina imagery and choose a motion prior that renders the optimization objective convex, which offers theoretical guarantees regarding the existence of a solution as well as practical performance benefits.

- Chapter 3 describes R-SLAM for 3D AO-OCT Imaging, a method for performing retina tracking and mapping from AO-OCT scans. The method initially projects the 3D volumes to 2D frames containing cellular structure similar to that captured by AOSLO. Next, we use the fast feature tracking method from chapter 2 to obtain 2D feature information, followed by 1D cross-correlation to retrieve feature information for the projected dimension. Splitting feature tracking into 2D and 1D subproblems improves the efficiency of this step, which is a bottleneck in prior methods. Finally, the 3D features are input to a 3D variant of the convex optimization procedure from chapter 2, which outputs the motion trace and therefore the retina map. We skip the gradient refinement step here as it is too computationally expensive in the 3D setting.

[1] Based on work originally published in ICCV 2021 [22].

Chapter 2

R-SLAM for 2D AOSLO Imaging

The adaptive optics scanning laser ophthalmoscope (AOSLO) is a device that images the retina at high resolution, capturing a 30 FPS video stream that can resolve individual photoreceptor cells. The AOSLO has mainly been applied in ophthalmology settings for recording videos of the retina, and existing approaches demonstrate real-time eye tracking speeds of 1 kHz [26]. These methods register strips of incoming retinal video against a pre-computed retina map, which in turn is generated by stabilizing a previously-recorded AOSLO video using offline eye tracking algorithms. Unfortunately, these offline techniques often produce distorted maps because they fail to completely correct for the entanglement of eye motion with the AOSLO's rolling shutter capture process.

In this chapter, we introduce a principled approach to disentangling eye motion from the rolling shutter video. We formulate and solve a holistic optimization problem that simultaneously computes retina motion and a map of retina appearance that faithfully explain the recorded AOSLO video. Jointly solving for this motion and retina map has not been attempted before because it is an under-determined problem; much like in visual SLAM (Simultaneous Localization and Mapping [24]), there is an inherent ambiguity between the moving location of the retina/eye and the underlying map of the retina's appearance. Our method, R-SLAM (for retina-based SLAM) consists of two stages (see Figure 2.1): first, we use convex optimization to compute an initial estimate of the motion. Formulating this initial step in a convex fashion offers guarantees about the existence and uniqueness of the optimal motion solution, as well as efficient algorithms to find this solution. Second, we perform joint refinement of the retina map and initial motion estimate, aiming to reconstruct the input video using gradient descent. Our contributions include:

- Formulation of eye-tracking from rolling-shutter retina video as an optimization problem.
- Convex initialization and gradient-based refinement of retina motion and retina map, in an offline algorithm that results in 3x less tracking error than prior work.
- Real-time eye tracking with 2x less error than prior methods, using the high-accuracy retina maps produced in the offline process and applying robust statistics to fast tracking based on normalized cross-correlation.

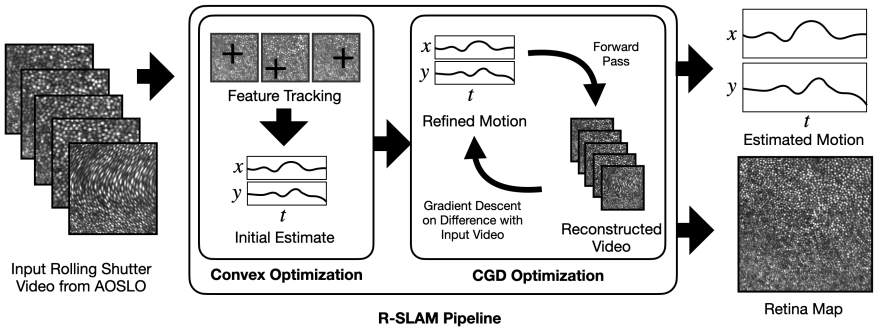


Figure 2.1: Our offline tracking algorithm's pipeline. R-SLAM receives as input distorted video of the retina, then proceeds to compute an initial motion estimate using convex optimization. We then use constrained gradient descent to jointly optimize the retina's motion and map, the latter of which can be used for real-time tracking.

2.1 Related Work

2.1.1 AOSLO-Based Eye Tracking

The AOSLO offers promising hardware for high-frequency eye tracking with subarcminute accuracy, but current software solutions for processing AOSLO video fail to completely disentangle the effect of rolling shutter from motion of the eye, leaving artifacts in motion estimates and residual distortions in estimated retina maps. Full details of AOSLO are provided by Roorda *et al*. [20], but we provide a brief overview here. The AOSLO records an image by measuring the scattered light from a focused spot on the retina as it sweeps in a raster scan. AOSLOs are capable of recording a live video of the human retina at a cellular resolution and high sampling density (typically 9.5 pixels per arcminute). Since the laser scans line-by-line from the top to bottom of each frame, the bottom portions of video frames are recorded later in time than the top portions. This vertical sweep combined with the eye's motion introduces rolling shutter distortion in the video frames [23] but also provides the opportunity for high-speed tracking [21].

There are several techniques that attempt to dewarp rolling shutter AOSLO video. Stevenson *et al*. [23] perform offline tracking by constructing a reference frame from a registered set of seed frames from a video sequence and subsequently registering all video frames to that reference, strip-wise, to form a larger retina map. Azimipour *et al*. [2] solve for motion within a single frame by registering the strips in the frame against the other frames and computing a dewarping bias. Bedggood *et al*. [4] use a similar method to [2], except Bedggood *et al*. solve for the eye's motion in the whole video by registering all the other frames against the single dewarped frame in a strip-based fashion.

These methods are moderately effective. Stevenson *et al*.'s approach reduces, but does not eliminate, artifacts from distortions in the reference frame. The outcome is that the apparent motion that gives rise to the distortion in the reference frame appears in the motion trace from each frame in the video. Empirically, these periodic artifacts manifest as spikes in the power spectrum

at the frame rate and higher harmonics (30 Hz, 60 Hz, 90 Hz, and so on) [5]. We make use of this phenomenon in analyzing tracking error in the absence of ground truth motion data in Section 2.3.2. Azimipour *et al*. and Bedggood *et al*. effectively minimize these artifacts, but their algorithms are not suited to stabilize the movie over the entire extent of the field of view, generate high fidelity images over the largest possible extent, or generate the most accurate and continuous eye motion traces. Unlike R-SLAM, these methods rely on registering motion against one or more seed frames that may contain rolling shutter artifacts themselves, and attempts to dewarp the seed frames fail to utilize dense interframe correspondence information throughout the entire video.

2.1.2 Rolling Shutter Correction for Frame Dewarping

A variety of algorithms exist to correct for rolling shutter, but most of them assume 3D world geometry [12, 27, 28]. Baker *et al*. [3] use the same 2D translational motion assumption as us, and their method is similar to our convex optimization step. Their algorithm consists of feature tracking via optical flow followed by linear programming to solve for a camera motion trace that is consistent with the tracked features. Our method is different from [3] in that we track features across the whole video instead of just neighboring frames in order to enable loop closure, which is important for the mapping aspect of our algorithm. Secondly, we use an l_2 loss to impose a Brownian random walk prior on the eye's motion, whereas [3] uses an l_1 loss to remove outliers from the set of tracked features (which we handle using RANSAC [11]). Thirdly, we refine the initial output of the convex step using gradient-based optimization over a different objective function.

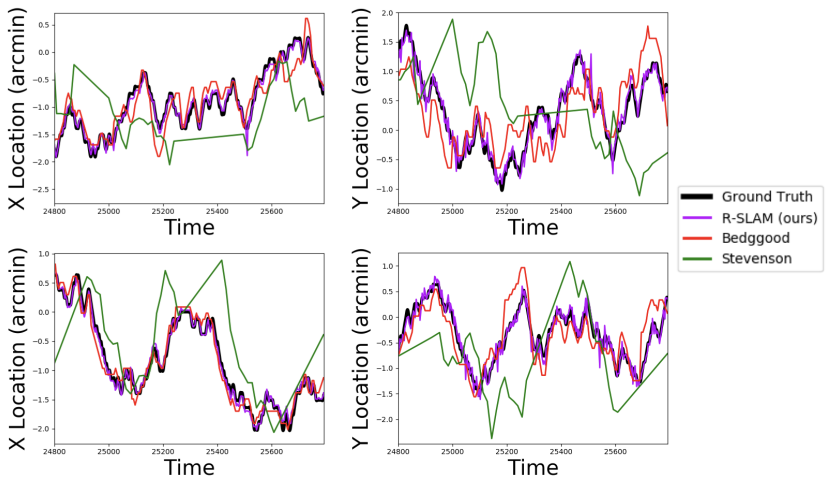


Figure 2.2: (Left) Comparison of offline tracking techniques on two different simulated AOSLO videos. Note that the simulated motion was set to a high level to stress-test all methods. R-SLAM, Azimipour *et al*., and Bedggood *et al*. are able to track the motion and offer a fair comparison, but the Stevenson *et al*. algorithm was not suited to track this magnitude of motion. Stevenson was able to track the real AOSLO videos (see Table 2.1) albeit with evident reference frame artifacts (Figure 2.4). R-SLAM achieves the most faithful reconstruction of the ground truth motion, particularly in the vertical (y) direction.

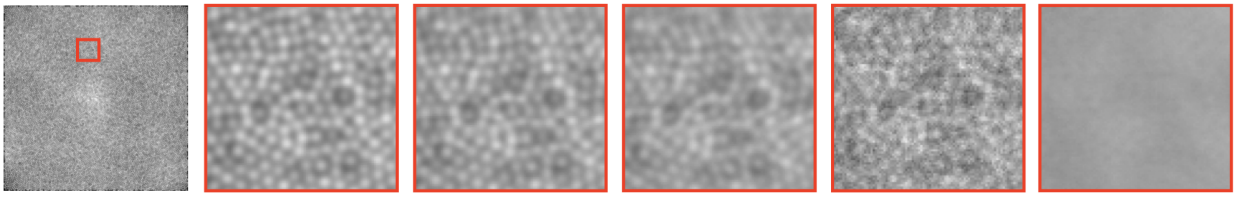


Figure 2.3: Comparison of different techniques for offline estimation of the retina map from simulated retina video, which contains extreme eye motion for stress testing. Stevenson fails to stabilize the input video, Azimipour (using only one frame) contains significant noise because it only stabilizes a single frame, and Bedggood *et al*. (equivalent to Azimipour with averaging of multiple frames) suffers from blurry cones in the top portion of the inset. Only R-SLAM properly resolves all cone cells in the image.

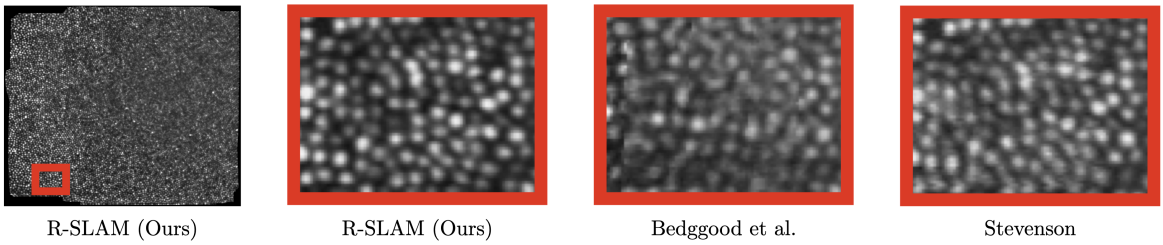


Figure 2.4: Comparison of different techniques for offline estimation of the retina map from real AOSLO video. Bedggood *et al*. suffers from blurry/distorted cone cells, and Stevenson contains sharpness issues and duplicated cone cells towards the bottom of the image. Only R-SLAM properly resolves all cone cells in the image.

2.2 Mathematical Background and System Overview

Our system directly models the AOSLO's video capture process to simultaneously optimize the retina's motion and appearance from an input recording. This allows arcminute accurate offline tracking and distortion-free map generation that subsequently enables high-quality real-time tracking.

We define the problem mathematically as follows:

We define the retina map as a 2D rigid image, with scalar intensity given by $R(x, y)$, where (x, y) are spatial coordinates on the retina. All spatial units are defined such that the AOSLO's output has unit width and height.

We define the retina's motion as a function of time, with the retina's 2D position given by $M(t) = [X(t), Y(t)]$. This $[X(t), Y(t)]$ is a position within the map of the retina. We ignore any torsional effects (rotations about the eye's optical axis) and model retina motion completely as translations. While torsion indeed occurs during fixational eye movement [15], we find that the translational eye motion assumption suffices for our AOSLO data.

We define the AOSLO's video as $V(u, v, i)$, which is the scalar intensity at position (u, v) within frame i. Positive u is rightward, and positive v is downward, with $u, v \in [0, 1]$. One unit of time is the inverse of the AOSLO's FPS.

We define the AOSLO forward model as $F(M, R)$, a function that attempts to reconstruct V given the retina's appearance R and its motion M. That is, $F(M, R)(u, v, i) = R(X(i + v) + u, Y(i + v) + v)$. Notice that we sample M at time $i + v$ rather than i to model rolling shutter capture.

We define the true motion and retina map to be M^*, R^*. Our goal is to produce $\hat{M}, \hat{R}$ from V which are as close to M^*, R^* as possible. To do this, we minimize the squared error between our reconstruction $F(M, R)$ and the input V:

$$\hat{M}, \hat{R} = \arg\min_{M,R} ||F(M, R) - V||^2 \tag{2.1}$$

2.2.1 Conceptual Overview

Equation 2.1 represents a video reconstruction objective. If it were of the form $\arg\min_x ||Ax-b||^2$, then we could potentially apply inverse-problem, optimization-based reconstruction techniques often used in computational imaging. However, in our case the map and motion are entangled in F, so we instead use constrained gradient descent (CGD) to optimize the objective as described in Section 2.2.4.

We initialize this gradient descent search with an input $\hat{M}_0, \hat{R}_0$ which is sufficiently close to M^*, R^*. Empirically, gradient descent search on Equation 2.1 performs poorly unless it is initialized well. We efficiently compute this initialization with a convex optimization to find a sufficiently accurate $\hat{M}_0$ followed by simple image rasterization to find the accompanying $\hat{R}_0$.

Our convex optimization step efficiently finds $\hat{M}_0$ as a globally optimal minimizer to a separate objective function defined in Section 2.2.2. Surprisingly, this convex optimization not only finds a good $\hat{M}_0$ to initialize CGD with, but it can do so independent of the retina map R. That is, rather than needing to jointly estimate both M^* and R^* simultaneously, this convex optimization step can estimate M^* directly without ever computing an $\hat{R}_0$. This is done by substituting V with a set of dense 2D features that are globally motion-tracked in V, as described in Section 2.2.2.1.

Given an estimate $\hat{M}$ and the original video V, we can use simple image rasterization techniques to produce an accompanying $\hat{R}$. This $\hat{R}$ is chosen to minimize Equation 2.1 for a fixed $\hat{M}$. This rasterization is expressed as $S(M, V)$, which yields a 2D image analogous to R. This is how we get $\hat{R}_0$ as $S(\hat{M}_0, V)$. S is described in more detail in Section 2.2.3.

2.2.2 Initial Eye Motion Estimation via Convex Optimization

We use convex optimization to efficiently compute an initial estimation of the eye's motion $\hat{M}_0$. Our construction is novel in the way it formulates global eye motion recovery as a convex problem using motion-tracked 2D points.

We define $\mathcal{G}$ to be a list of globally motion-tracked 2D image features found via the method described in Section 2.2.2.1. Each $G \in \mathcal{G}$ is a single 2D image patch which we represent as a list of the times and locations it is found in V. That is, $(u_j, v_j, t_j) \in G$ means that the jth time that G was found in V, it was found at time t_j at position u_j, v_j within the frame. G is sorted in increasing t.

Each $G \in \mathcal{G}$ is a noisy estimate of M^*, as visualized in Figure 2.5. We define the following loss for our estimate of M^* given a single $G \in \mathcal{G}$:

$$L(M, G) = \sum_{j=1}^{|G|-1} ||(M(t_j) - M(t_{j+1})) - (p_{j+1} - p_j)||^2 \tag{2.2}$$

where $p_j = (u_j, v_j)$.

We also impose a Brownian prior on M^* to help regularize our estimation. We model M^* as a Brownian random walk sampled at discrete steps. The sample times are a list T, sorted in increasing order, and are the collection of all t_j for all G in $\mathcal{G}$. Each step of the Brownian random walk is a zero-mean 2D Gaussian with variance equal to the duration of the step. Taking the negative log likelihood:

$$L(M) = \sum_{i=1}^{|T|-1} \frac{||M(t_{i+1}) - M(t_i)||^2}{t_{i+1} - t_i} \tag{2.3}$$

By combining the inter-frame, tracking-based objective of Equation 2.2 with the intra-frame objective of Equation 2.3, we arrive at our overall convex optimization formulation:

$$\hat{M}_0 := \arg\min_M \lambda_B L(M) + \lambda_T \sum_{G \in \mathcal{G}} L(M, G) \tag{2.4}$$

Here, $\lambda_B, \lambda_T \geq 0$ are hyperparameter weights. Equation 2.4 is a convex quadratic program (QP) with no constraints, and so we can readily use existing convex optimization software packages [1, 7] to compute an optimal solution for $\hat{M}_0$. A proof of convexity is provided in appendix A. $\hat{M}_0$ is a discrete motion trace. We form a continuous representation of $\hat{M}_0$ via linear interpolation.

2.2.2.1 Feature Tracking

We track patch features across the entire duration of the video to ensure loop closure, doing so using a map-aware tracker that runs in linear time with respect to the size of the map times the duration of the input video.

Each incoming video frame (384 pixels wide by 496 pixels tall) of the input V is divided into a grid of 64 by 16 patches. The patch height of 16 pixels in the vertical scan direction corresponds to 1 ms of capture time, which is short enough to prevent the eye from moving significantly and causing distortion within the patch (in the absence of saccades). Each patch is a single feature that is tracked forward through time by registering it against all future incoming frames via fast

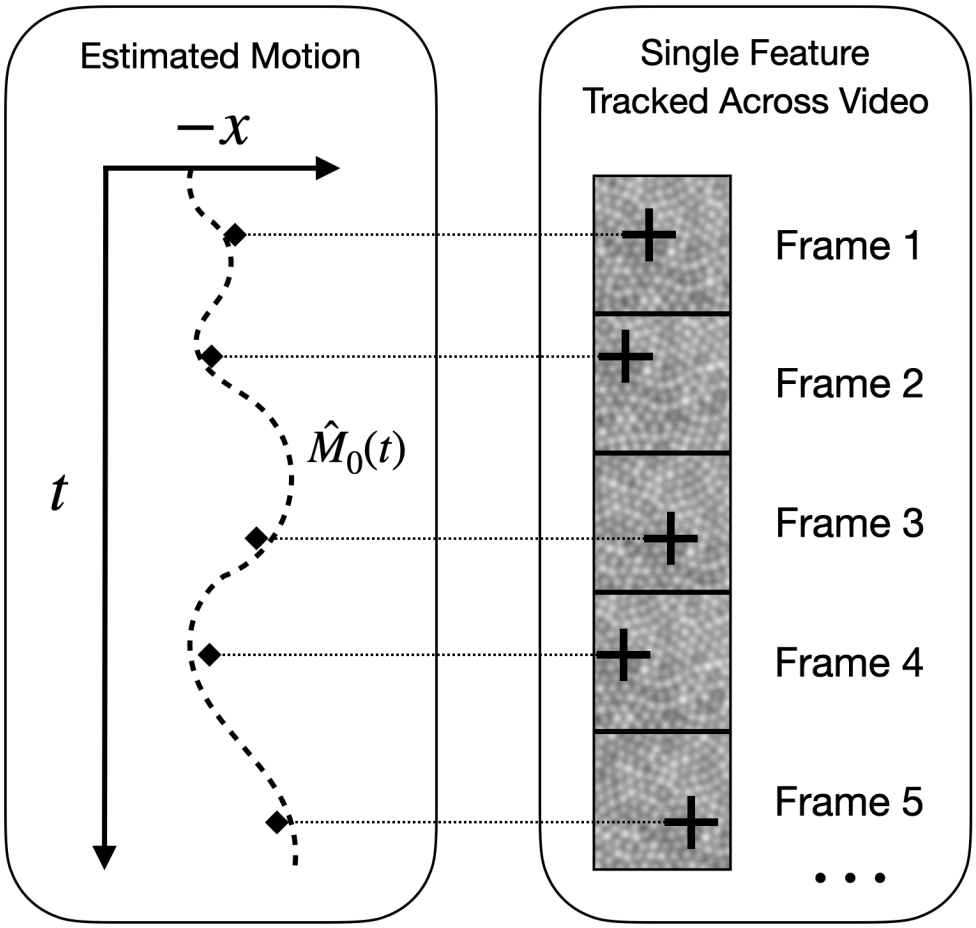


Figure 2.5: The relationship between tracked video features and eye motion trace, used as a loss term in our convex optimization formulation (Equation 2.4). Left: horizontal position of the retina as a function of time. The noisy motion samples from a single feature are shown as diamonds. The dashed curve is the motion estimate $\hat{M}_0(t)$ resulting from our convex optimization. Right: a sequence of video frames with a single feature highlighted. If a feature appears at column u_1 at time t_1, and appears at column u_2 at time t_2, then the eye must have moved horizontally by approximately $u_1 - u_2$ between t_1 and t_2. The difference between the total estimated motion $\hat{M}_0$ and the motion implied by the feature tracking is minimized in Equation 2.2.

normalized cross-correlation [13] implemented on the GPU. To make this feature tracking robust to outliers, we group together features that lie in the same row into strips and perform RANSAC on these strips, aiming to calculate each strip's displacement in every subsequent frame based on the maximum number of constituent features that agree on that displacement. Features are said to agree if their displacements are less than 2 pixels apart.

Tracking features across the entire video is important because it ensures loop closure, allowing

the algorithm to recognize when a frame in the video revisits a part of the map that was explored much earlier. However, tracking every feature against every other frame would be of complexity $O(mn^2)$, where n is the number of frames and m is the number of features per frame. This brute force approach is computationally infeasible, so we instead only choose to track features that correspond to distinct areas of the underlying retina map. Every time our map-aware tracker encounters a new frame, if a particular candidate feature in that frame has been matched with a previously-seen feature, then we discard the candidate feature and don't add it to the set of tracked features. More specifically, if some fraction λ_f of the candidate feature's area intersects a previous feature, the candidate is discarded. This ensures that the number of tracked features remains proportional to the size of the map, making the algorithm run in $O(rn)$ time, where r is the number of tracked features.

Furthermore, the feature tracker maintains a concept of good and bad features within the tracked set - if a feature has not been matched to at least λ_n frames in total or one of the past λ_m frames, it is immediately discarded. This rule removes features that offer little tracking data.

The λ hyperparameters can be tuned to give various trade-offs between speed and the density of features, but in practice $\lambda_f = 0.9$, $\lambda_n = 4$, and $\lambda_m = 6$ give considerable speed-up for no noticeable loss in performance.

2.2.3 Drawing a Retina Map Given Estimate of Eye Motion

If we are given M, then we can directly solve for a map $S(M, V)$ which minimizes $\min_R ||F(M, R) - V||^2$ for the fixed M. Given the motion of the retina, we know where in retina map each pixel of V is sampling from. Therefore, we construct the retina map S where the value at each location is the average of the samples taken at that location. This average value minimizes the squared error against the noisy samples, thus minimizing $||F(M, R) - V||^2$. Conceptually, we build $S(M, V)$ by first using M to cancel out the motion in each frame of V, producing a stabilized video. The frames of this stabilized video are averaged together to produce S.

2.2.4 Simultaneous Refinement of Eye Motion and Retina Map via Constrained Gradient Descent (CGD)

R-SLAM jointly estimates the retina map and motion using constrained gradient descent (CGD), with the initialization $\hat{M}_0, \hat{V}_0$ from the earlier steps. CGD converges much faster than naively performing gradient descent on Equation 2.1 because it enforces consistency between the current map estimate R and the input video V.

Using $\hat{M}_0, \hat{V}_0$ as the starting point for gradient descent is not enough to ensure quick convergence. One issue is that the optimization problem in Equation 2.1 is not sufficiently constrained. To remedy this, we expect the following to hold true for M^*, V^*:

$$||S(M^*, V) - R^*||^2 \leq \epsilon. \tag{2.5}$$

In Equation 2.5, ϵ serves as a measure of the noise in the process used record V. This constraint can be added to Equation 2.1 to produce the new optimization problem:

$$\begin{aligned} &\arg\min_{M,R} ||F(M,R) - V||^2 \\ &\text{s.t. } ||S(M,V) - R||^2 \leq \epsilon. \end{aligned} \tag{2.6}$$

Recall that M^*, R^* are the desired optimal solutions. To make the constraint in Equation 2.6 amenable to gradient descent, we observe that in the presence of white noise, $S(M^*, V) = R^*$ in expectation, which holds in deterministic terms as the number of video frames goes to infinity by the central limit theorem. That is, averaging noisy frame measurements should yield the true retina map R^* as the number of frames goes to infinity. Thus, given sufficient frames, the constant ϵ that bounds the difference between $S(M^*, V)$ and R^* is negligible. We then make the approximation that $\epsilon = 0$, which implies:

$$\begin{aligned} &||S(M,V) - R||^2 \leq \epsilon = 0 \\ \implies &||S(M,V) - R||^2 = 0 \\ \implies &S(M,V) = R. \end{aligned} \tag{2.7}$$

We approximate Equation 2.6 with the new objective:

$$\arg\min_{M} ||F(M, S(M,V)) - V||^2. \tag{2.8}$$

The retina map R is no longer a variable being optimized directly—it is captured completely in the stabilization function S. Nevertheless, the retina map is still being jointly estimated with the motion M, it is simply stored as a function of the input video V. Equation 2.8 ensures consistency between $\hat{M}$, $\hat{R}$, and V, enabling faster convergence. We iteratively optimize Equation 2.8 via Algorithm 1.

Algorithm 1: Motion Refinement

Input: $V, \hat{M}_0, \alpha, n$
for $i \leftarrow 1$ **to** n **do**
  $\hat{R}_{i-1} \leftarrow S(\hat{M}_{i-1}, V)$;
  $\hat{V} \leftarrow F(\hat{M}_{i-1}, \hat{R}_{i-1})$;
  $L \leftarrow ||V - \hat{V}||^2$;
  $\hat{M}_i \leftarrow \hat{M}_{i-1} - \alpha \nabla_{\hat{M}_{i-1}} L$;
end

In this algorithm, α and n are tunable hyperparameters corresponding to the step size and number of descent iterations, respectively. The functions S and F are implemented as differentiable rasterization operations in PyTorch [19], and the reconstruction loss L is naturally differentiable as it is simply the Euclidean norm of the difference of two three-dimensional tensors (video representations).

2.2.5 Real-Time Eye Motion Tracking

Like the prior art reviewed above, our real-time tracking method uses normalized cross-correlation to calculate the position of the latest strip of video against a retina map. We make two important improvements. First, we use a more accurate retina map, optimized in the offline process described above. Second, we increase robustness of calculating the location of the latest strip by applying RANSAC. Every incoming video frame from the AOSLO is split into horizontal strips 384 pixels wide and 16 pixels tall. Each strip is split into n sub-strips each of size $\lfloor 384/n \rfloor$ x 16. Each sub-strip is then independently registered to acquire n estimates $P = \{p_1, \ldots, p_n\}$ for the retina's 2D position relative to the AOSLO. We use RANSAC [11] to filter out outliers, which is more robust than determining strip registration quality directly with the peak values output by normalized cross-correlation.

2.3 Evaluation

R-SLAM is evaluated on both simulated and real AOSLO video. Simulated tests allow us to compute exact accuracies at the arcminute scale, while tests on real video highlight R-SLAM's ability to remove distortions that manifest as spikes in the power spectrum. We only compare R-SLAM to prior motion estimation techniques intended for retinal imagery. More general SLAM algorithms are excluded from comparison because they typically employ feature trackers that are tailored for macroscopic objects and are therefore ill-suited for tracking the self-similar cone cells of the retina.

2.3.1 Simulation

We first evaluate the accuracy of tracking algorithms on simulated AOSLO video, where we have ground truth eye motion. First, we generate 15 synthetic cone mosaics using the particle system described in [2]. Then, we compute a pair of three-second videos per mosaic using artificial eye motion traces derived from the random walk model in [9], which integrates fixational eye movements and microsaccades. The simulated motion was set to a high level as a stress-test for all methods. Altogether, the simulated dataset contains 30 synthetic videos. The results of the evaluations described below on this dataset are given in Table 2.1.

The offline tracking algorithm is tested on individual simulated videos, whereby the trace output by our method is sampled at the frequency of the ground truth motion trace and then compared to this ground truth. We compute the average magnitude of the 2D vector difference between the output and ground truth traces. Since these traces can be arbitarily offset, we use the offset that minimizes the error magnitude.

The real-time algorithm is tested on individual videos, using retina maps generated by other videos of the same cone mosaic.

To test the effect of CGD on RMSE, we evaluate an ablation of our system with CGD held out.

	Method	Stevenson [23]	Bedggood *et al*. [4]	Azimipour *et al*. [2]	R-SLAM without CGD	R-SLAM
1	Simulated Video: Offline Mean Error Magnitude (pixels / arcmin) ↓	9.97 / 1.05	2.67 / 0.280	N/A	1.15 / 0.121	**0.821 / 0.086**
2	Simulated Video: Real-time Mean Error Magnitude (pixels / arcmin) ↓	4.82 / 0.506	2.67 / 0.280	2.68 / 0.281	**1.31 / 0.138**	**1.31 / 0.138**
3	Real Video: Offline Spectral Error (X / Y Direction) ↓	4.84 / 6.54	3.25 / 5.59	N/A	N/A	**1.81 / 1.67**
4	Real Video: Real-time Average Difference Magnitude w.r.t. Offline R-SLAM (pixels / arcmin) ↓	26.95 / 2.83	34.60 / 3.63	38.17 / 4.01	N/A	**23.98 / 2.52**

Table 2.1: R-SLAM evaluated on simulated video (rows 1/2) and real AOSLO video (rows 3/4). Row 1: we compute the magnitude of the error (displacement) of each output motion trace against the ground truth, taking the mean over all tracking points in the trace. R-SLAM incurs 3x less error than prior work. Row 2: each method outputs a map that is used for real-time tracking, and the real-time traces are compared against the ground truth. We compute the mean magnitude of the error for each real-time motion trace. R-SLAM incurs 2x less error than prior work. Row 3: in the absence of ground truth, we compute the spectral error of each output motion trace, which penalizes spike artifacts occurring at the harmonics of 30 Hz in the power spectrum (defined in Section 2.3.2). Row 4: each method outputs a map that is used for real-time tracking, and the real-time traces are compared to the trace output by offline R-SLAM, which is the best offline tracking method available in the absence of ground truth. Azimipour *et al*. [2] is only used to test real-time tracking because we only use it to compute a retina map. R-SLAM without CGD is only included as an ablation for comparison on simulated video.

2.3.2 Real-World AOSLO Video

We validate R-SLAM on 34 real AOSLO videos previously recorded from two of the human subjects that were reported in a paper published by Wang *et al*. [25]. Since real-world AOSLO videos do not have ground truth motion traces, we use alternative metrics to evaluate the performance of our algorithms. The results of the evaluations described below on this dataset are given in Table 2.1.

One method consists of a spectral analysis, in which we inspect the amplitude vs. frequency of the estimated motion trace. As described in Section 2.1.1, periodic artifacts arising from distortions in the retina map manifest as spikes at the video frame rate and higher harmonics (30 Hz, 60 Hz, 90 Hz, and so on) that deviate from the expected inverse-frequency dependence of eye movements (-1 slope on a log-log plot) [10]. Similar-appearing spectral artifacts are reported in prior work [5, 23]. To quantify the magnitude of these spikes in the power spectrum, we fit a line to the spectrum on a log-log plot and define the spectral error as the sum of any positive deviations from the linear fit evaluated at 30 Hz and higher harmonics.

We also run our real-time method using various offline retina mapping techniques. The 34 videos consist of 17 pairs, where each pair comes from a single retinal location in one of the two subjects. The real-time method is evaluated on each video by using the other video in the pair to create a retina map, and the real-time motion trace is compared to the output of offline R-SLAM on the same video.

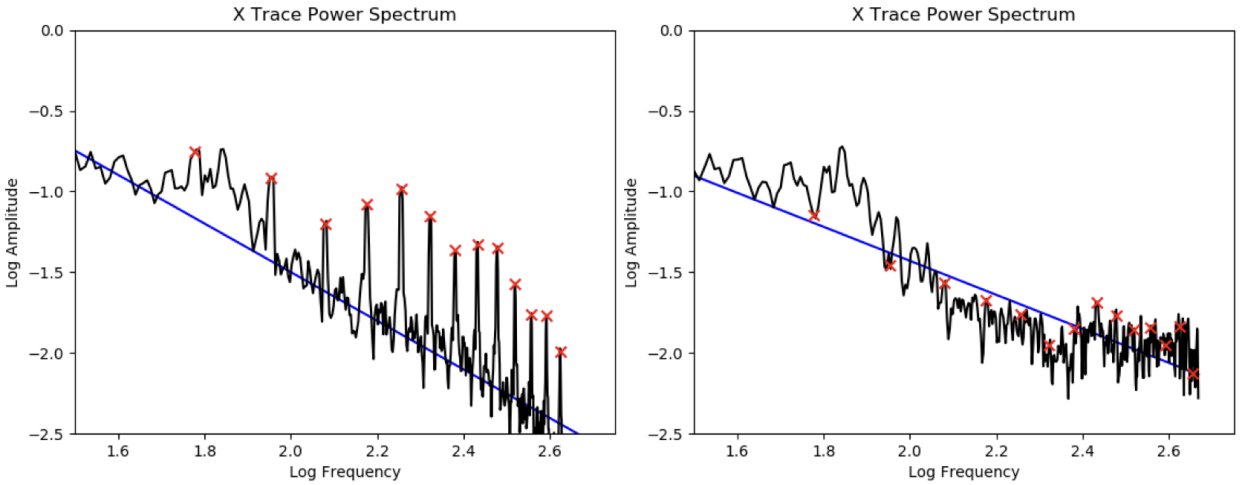


Figure 2.6: Comparison of power spectra of motion traces output by Stevenson [23] (left) and R-SLAM (right). In black is the power spectrum of the motion trace for a given video, in blue is the best linear regression fit, and in red are markers denoting the harmonics of 30 Hz (30 Hz, 60 Hz, 90 Hz, and so on). Stevenson exhibits large spikes at these harmonics, indicating that their motion traces contain periodic artifacts. The R-SLAM estimated motion does not exhibit these artifacts.

2.4 Discussion

In this section, we examine in greater detail R-SLAM's differences with prior methods. We also provide further directions for future work.

2.4.1 Analysis of Results

On the simulated dataset, R-SLAM achieves the lowest error on both real-time and offline eye motion tracking compared to prior work (Table 2.1). R-SLAM achieves 0.8 pixels of mean displacement error against the ground truth (improving from 1.15 pixels of error with only convex optimization and no CGD). This represents a 3x error reduction compared to prior work. When using each method's estimated retina maps for real-time tracking, we find that R-SLAM achieves 2x lower error compared to prior work. There is no significant difference between using the retina maps obtained before and after CGD. This is unsurprising since the cross-correlation step in real-time tracking is only accurate to a pixel, and CGD only yields sub-pixel improvements on this dataset.

On the real-world dataset, R-SLAM also achieves lower errors on both real-time and offline eye motion tracking compared to prior work (Table 2.1). The real-world data lacks known ground truth motion. In place of ground truth we use R-SLAM's offline output since in simulation it achieves sub-pixel error. For this reason, it is impossible to evaluate the offline output using the same metric as in Row 1 of Table 2.1. Using the spectral error metric defined in Section 2.3.2, R-SLAM achieves the lowest error, which corresponds to the artifact-free power spectra shown in Figure 2.6.

2.4.2 Future Work

We hope that our optimization-based framework can bring new AOSLO applications within reach. One place for improvement is to incorporate 2D rotation (torsion) into our model. Baker *et al.* [3] show some success in approximating rotation with a general affine transform, and similar modifications may be applicable to our model. Another future direction is to generate maps that encompass larger areas of the retina, which would require us to address its curvature. We currently model the retina as a planar surface, which proves sufficient for our motion and map estimation experiments. However, a natural extension would be to adopt a spherical model, which would enable the creation of larger maps where the retina's curvature becomes a significant factor.

Empirically, we have used R-SLAM to produce larger, 2x2 degree retina maps from grids of retina recordings. One limitation we have noticed is that running R-SLAM multiple times on the same set of input videos produces slight perturbations between consecutive tracking points in the output motion traces. As a result, retina maps output from separate runs of R-SLAM do not align along the vertical dimension. The issue is that due to rolling shutter, the system loses information about the absolute positioning of consecutive lines of video such that multiple motion traces with subpixel offsets between consecutive lines may appear equally optimal to R-SLAM, while in fact the accumulation of these offsets over entire frames can produce retina maps that are visually

different from one another. One potential area for future work is to incorporate alternate scanning patterns into the solution of the motion objective in order to resolve the ambiguity that arises from rolling shutter.

Chapter 3

R-SLAM for 3D AO-OCT Imaging

Adaptive optics optical coherence tomography (AO-OCT) is a technique for imaging the 3D structure of the retina in vivo. Figure 3.1a shows a retina volume scanned via AO-OCT, with a canonical set of axes overlaid for the purpose of visualizing the scanning mechanism. The volumes studied in this chapter are acquired via line scan AO-OCT, which operates by imaging a two-dimensional fast B-scan aligned with the xz plane that consists of multiple one-dimensional A-lines captured simultaneously along the z axis. A scanner then sweeps along the y axis, stacking consecutive fast B-scans to produce a 3D volume.

There are three types of axis-aligned 2D images that can be sliced from an AO-OCT volume: (1) a fast B-scan, which is parallel to the xz plane and directly measured by the system via simultaneously-captured A-lines, (2) a slow B-scan, which is parallel to the yz plane, and (3) a C-scan, which is parallel to the xy plane. Figure 3.1a labels three examples of these images. In addition, one can average all the C-scan images to produce an en face projection image that captures the appearance of the photoreceptors. Figures 3.1b and 3.1c compare a C-scan image with an en face projection, demonstrating that the latter can reveal structures that are not visible in a single C-scan.

The en face projection resembles a 2D AOSLO image in that it makes cellular structures visible. Because all the samples along each A-line are captured simultaneously, the information recorded in each pixel of the en face projection corresponds to a unique instant in time. Moreover, the en face projections are less noisy than the individual C-scans that produce them, suggesting that en face projection images could be amenable to the 2D feature tracking algorithm introduced in chapter 2.

The main contribution of this chapter is to cast the problem of estimating the retina's map and motion from AO-OCT in terms of the R-SLAM optimization framework. The core idea of simultaneously estimating the retina map and motion remains the same as in the AOSLO setting, and the convex optimization setup is identical, with the addition of a z variable for the depth dimension. The main difference from chapter 2 is that we split 3D feature tracking into two steps: first, we perform 2D feature tracking on the en face projections of the volumes to estimate dense, inter-volume (x, y) offsets, and second, we perform 1D correlation on the tracked features to retrieve z offsets. The retina map is generated by stabilizing and averaging all the input volumes.

3.1 Related Work

Previous approaches to 3D retina tracking and mapping fall into two categories: one based on 3D correlation and a second based on segmentation. 3D correlation-based techniques register entire volumes against a reference in 3D, while segmentation-based techniques first perform 2D registration on en face projections followed by z-dimensional registration to correct depth-wise motion. Both approaches rely on heuristics to remove intra-volume distortion, and are described in more detail in the following sections. For clarity, these sections refer to the volume being registered against the reference as the target volume.

3.1.1 3D Correlation-Based Registration

3D correlation-based techniques operate by registering each fast B-scan of the target volume against the reference. However, 3D correlation is an expensive operation as it involves computing the Fourier transforms of volumes containing tens of millions of voxels. As such, registering every single fast B-scan in the target volume against the entire reference volume is too costly. To remedy this issue, Do [8] proposes a coarse-to-fine scheme that first registers every n-th fast B-scan in the target volume against the reference and subsequently registers the remaining fast B-scans against subsets of the reference. The coarse-to-fine technique effectively reduces the search space for most of the fast B-scans, speeding up registration as a result.

Do's algorithm does not address distortions in the reference volume itself. Li *et al*. [14] attempt to account for these distortions by extending the correction technique of Bedggood and Metha [4] to 3D, employing a coarse-to-fine strategy that is similar to that of Do.

3.1.2 Segmentation-Based Registration

Segmentation-based methods perform registration separately in the xy and z dimensions. Azimipour *et al*. [2] achieve this by first segmenting, or cropping, each volume to just the photoreceptor layer, followed by conducting xy registration of the target volume's en face projection against that of the reference. The z dimension is resolved by registering strips from the slow B-scan projection of the target against that of the reference. Finally, Azimipour *et al*. correct reference distortions via a technique similar to that of Bedggood and Metha [4]. The advantage of segmentation-based methods is that they rely on 2D registration, which is computationally cheaper than 3D registration. Our method, outlined in the following section, similarly splits 3D feature tracking into a 2D registration step on the en face projections followed by a 1D correlation step along the z dimension, which has the added benefit of allowing us to use the robust 2D feature tracking method developed in chapter 2. We use a segmentation-based method described by Li *et al*. [14] as a baseline with which to compare our method.

3.2 Method

R-SLAM for AO-OCT operates by a similar principle as in the AOSLO case, which is to estimate the retina map and motion that most likely produced the set of acquired volumes. This optimization problem is described by equation 2.1 from the previous chapter, restated below for clarity.

$$\hat{M}, \hat{R} = \arg\min_{M,R} ||F(M, R) - V||^2 \qquad \text{(2.1 revisited)}$$

In this equation, M is the retina motion, R is the map, and V is the set of AO-OCT input volumes. F represents the forward operator, which computes the volumes that would be recorded if the retina map R were to undergo motion M. Practically, solving the optimization problem in equation 2.1 involves the same two steps as for the AOSLO case: feature tracking followed by convex optimization to retrieve the motion trace.

3.2.1 3D Feature Tracking

3D features correspond to one or more consecutive fast B-scans stacked together into a volume – in other words, subsets of the input volumes. Given a feature f_1 located in volume v_1, the aim of feature registration is to determine the location of f_1 within a second volume v_2. Formally, we compute the the spatial displacement (x', y', z') of f_1 between the two volumes. Figure 3.2 illustrates this definition of 3D registration.

Our method first determines the (x', y') components of the displacement by performing 2D feature registration between the en face projections of v_1 and v_2. Afterwards, we resolve the z' displacement via 1D correlation. Figure 3.3 illustrates how we split feature registration along the xy and z dimensions. To perform feature tracking, we compute dense feature correspondences between all the input volumes, skipping previously tracked features in the same manner as in chapter 2.

3.2.2 Convex Optimization

The convex optimization step is identical to that of the AOSLO case, with the addition of a z variable. At a high level, its objective is to calculate a motion trace that agrees with the feature correspondences as closely as possible while also obeying a Brownian prior. This objective is captured in equation 2.4 from the previous chapter, which is reproduced below.

$$\hat{M}_0 := \arg\min_{M} \lambda_B L(M) + \lambda_T \sum_{G \in \mathcal{G}} L(M, G) \qquad \text{(2.4 revisited)}$$

As stated before, this objective function now solves for a 3D motion trace instead of a 2D trace as done for AOSLO. The problem still remains convex because even in the AOSLO setting, the solution of the x and y dimensions can be considered independent, convex quadratic programs, and the z dimension behaves symmetrically, maintaining the convexity property of the overall problem.

As such, we use the same solver from the previous chapter, extended with 3D coordinates, to retrieve the motion trace for the AO-OCT volumes. This motion trace is used to stabilize the input volumes, which are then averaged to obtain a 3D retina map.

3.3 Evaluation and Discussion

We evaluate our method on a dataset of volumes acquired during a single imaging session using adaptive optics line scan spectral domain OCT, courtesy of the Sabesan Lab at the University of Washington [16, 17, 18]. Qualitatively, we observe that R-SLAM stabilizes input volumes in a smooth, realistic fashion, avoiding some of the discontinuous artifacts exhibited by the segmentation-based method of Li *et al*. [14].

Figure 3.4 compares the retina maps generated by R-SLAM and the segmentation-based method of Li *et al*. [14]. Visually, the two methods perform similarly, with R-SLAM producing a map of slightly higher quality that avoids some of the vertical artifacts produced by Li *et al*. These maps are created by stabilizing and averaging all the input volumes.

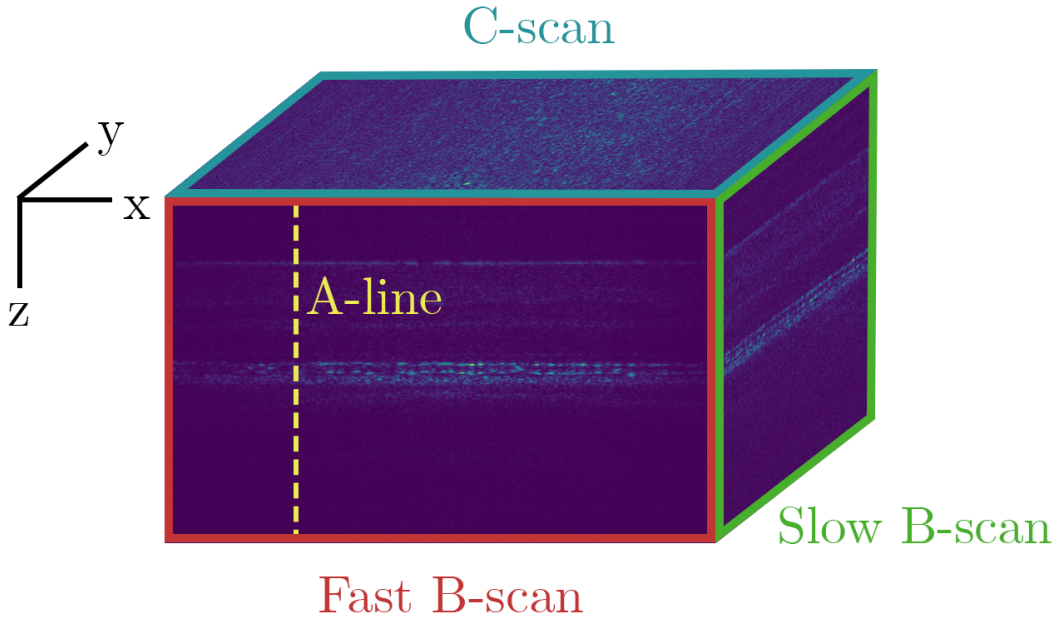
Figures 3.5 and 3.6 compare the efficacy of R-SLAM and Li *et al*. at correcting motion artifacts in two distinct input volumes. The portion of the retina being imaged in these figures is approximately planar, but intra-volume motion causes significant z-dimensional distortion in the input volumes shown. R-SLAM is able to correct this distortion in order to properly resolve the true appearance of the retina, as evidenced by the fact that its slow B-scan projections contain a smooth photoreceptor layer exhibiting slight curvature. By contrast, Li *et al*.'s corrected volumes contain sawtooth artifacts that fail to represent the continuous nature of the photoreceptor layer. These sawtooth artifacts occur because the segmentation-based method of Li *et al*. attempts to register discrete vertical strips from the slow B-scan projections of the input volumes to that of the reference volume, resulting in discontinuities between consecutive strips when the input volumes are stabilized. By contrast, R-SLAM computes a continuous motion trace for all the volumes, smoothly correcting intra-volume motion distortion for each fast B-scan.

R-SLAM succeeds at producing volumes with realistic appearance and curvature, indicating that it can accurately estimate eye motion even in the presence of heavy distortion. In some settings, it may be desirable to flatten the photoreceptors into a horizontal plane rather than reconstructing the true, curved appearance of the retina as done by R-SLAM. This type of flattening could be performed as a post-processing step following R-SLAM by: (1) identifying a set of points lying on the photoreceptor layer, (2) triangulating these points into a mesh, and (3) warping the volume such that the 2D mesh becomes a horizontal flat plane.

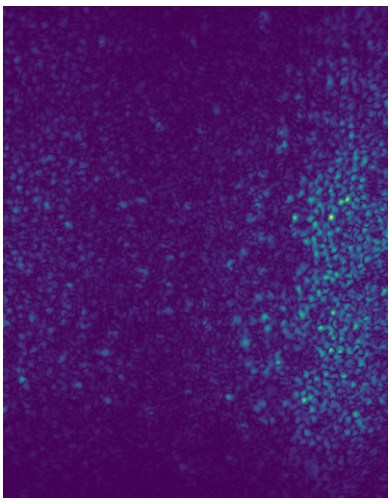
The quality of R-SLAM's mapping and motion tracking points to the effectiveness of an inverse optimization approach in the AO-OCT regime. Our 3D feature tracking and convex optimization techniques perform well on the provided dataset, and we hope they find immediate application within the lab. R-SLAM has fast performance, processing the provided dataset of 39 volumes at a rate of about 1 minute and 4 seconds per volume end-to-end. While the input volumes are of cubic complexity, our method's memory requirements (excluding the input and output volumes) are of

sub-cubic complexity because we operate on 2D and 1D projections of the volumetric data. Our codebase will be made available to researchers upon request.

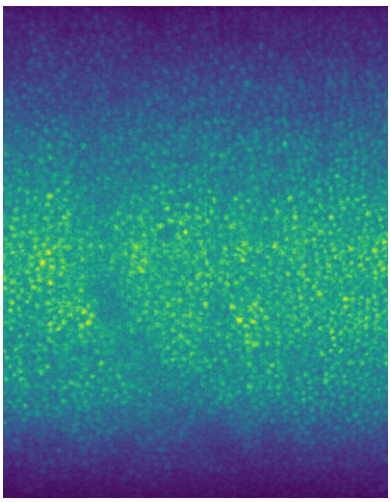
It is important to note that while our particular implementation choices are effective, R-SLAM as presented is a general framework that offers an optimization-based theoretical foundation for retina tracking and mapping. Future work could be done to improve, for instance, the 3D feature tracking method (the splitting technique is a heuristic we use for speed purposes) or even the choice of motion prior in the convex optimization objective. The framework is agnostic to these particular choices. Indeed, one could forgo the feature tracking and convex optimization entirely and directly solve the inverse problem by using a gradient-based approach similar to that in chapter 3, although we avoid that here due to the inefficiency of 3D gradient-based optimization.



(a) AO-OCT volume with labeled axes. A series of A-lines are captured simultaneously to form a single fast B-scan, and a scanner repeats this process along the y dimension to produce the volume.



(b) C-scan taken from the photoreceptor layer of the volume above, which reveals some but not all of the cellular structure.



(c) En face projection computed by averaging the C-scans from the volume above, revealing the entire photoreceptor layer.

Figure 3.1: Visualization of an AO-OCT volume with an example of a C-scan and en face projection.

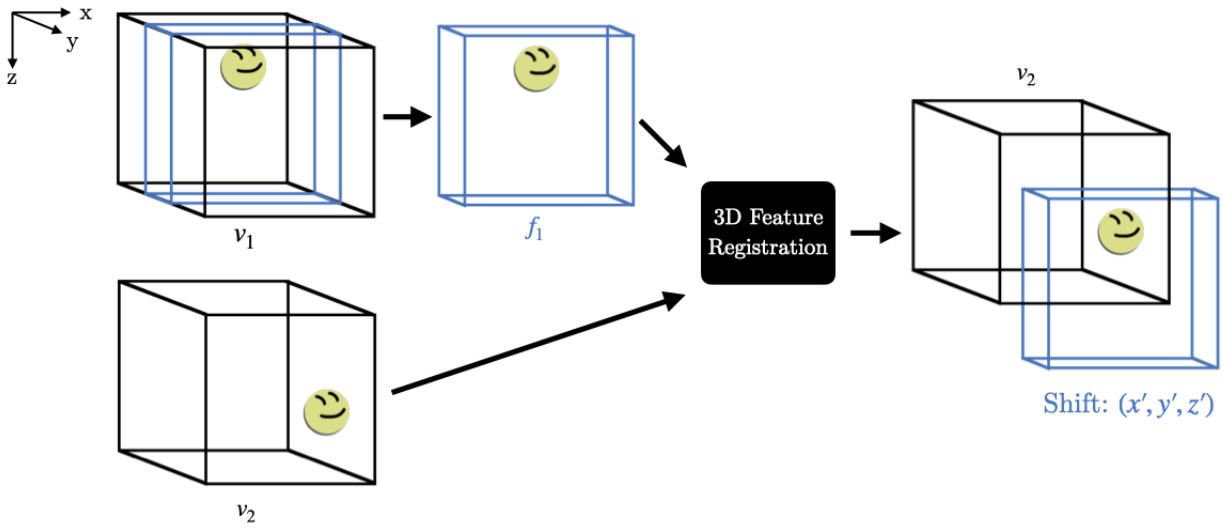


Figure 3.2: High-level 3D feature registration pipeline. Here, registration provides us the spatial displacement (x', y', z') of feature f_1 between volumes v_1 and v_2. Figure 3.3 illustrates this pipeline in more detail.

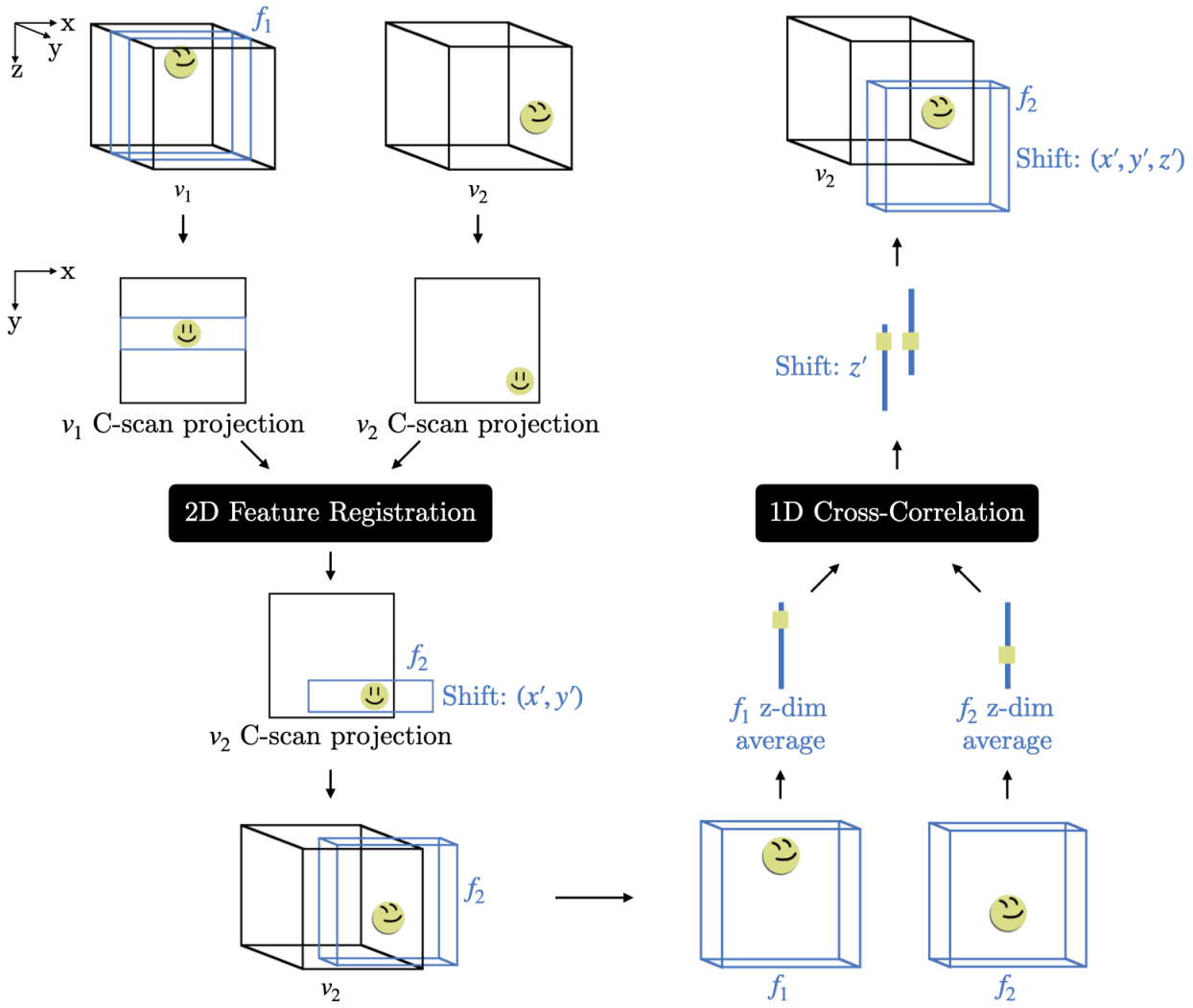
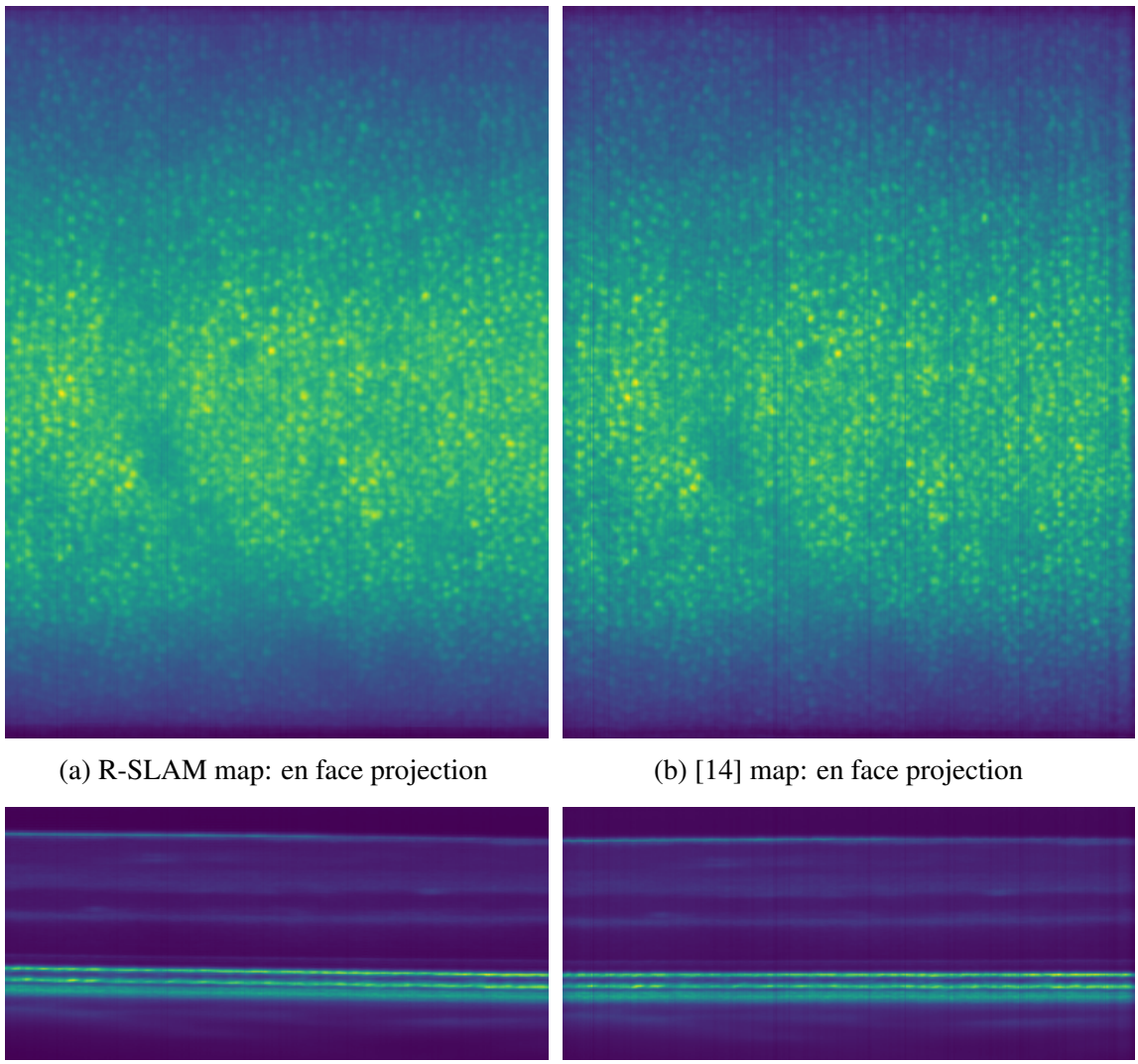


Figure 3.3: Illustration of 3D feature registration, which proceeds by 2D registration to resolve the xy shifts followed by 1D correlation to determine the z shift. Here, we wish to locate 3D feature f_1 from volume v_1 within volume v_2. First, we take the C-scan projection of both volumes, and then register f_1 within v_2 in 2D, which provides the relative shift (x', y'). Using this information, we extract the 3D feature f_2 from v_2, and average f_1 and f_2 along the x and y dimensions to produce one-dimensional vectors aligned with the z-axis. These 1D vectors are illustrated with nonzero width for clarity. We apply 1D cross-correlation to these vectors, producing the relative z-dimensional shift z'. (x', y', z') is then the relative displacement of feature f_1 within volume v_2.



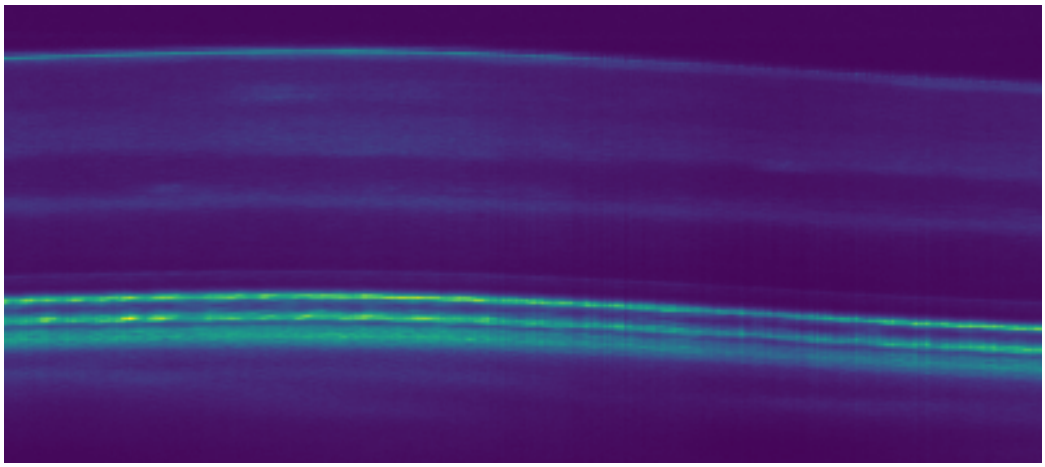
(a) R-SLAM map: en face projection

(b) [14] map: en face projection

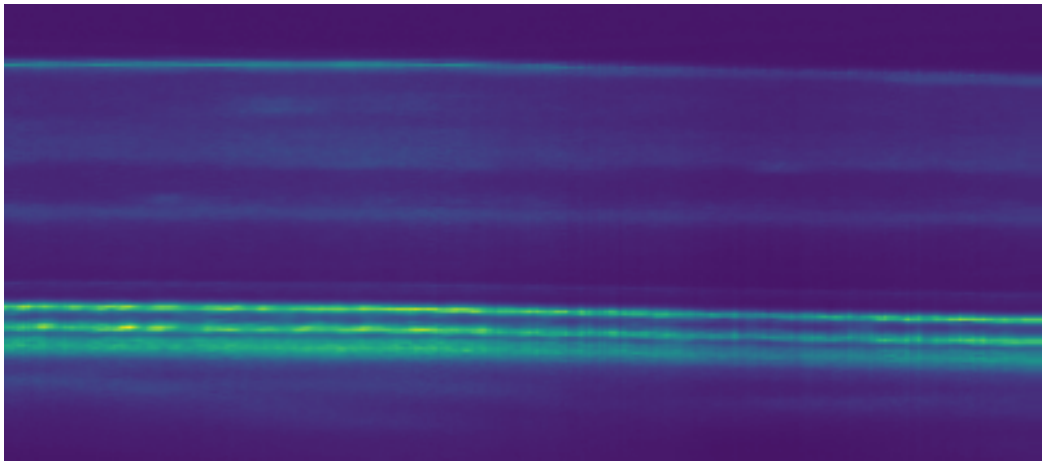
(c) R-SLAM map: slow B-scan projection

(d) [14] map: slow B-scan projection

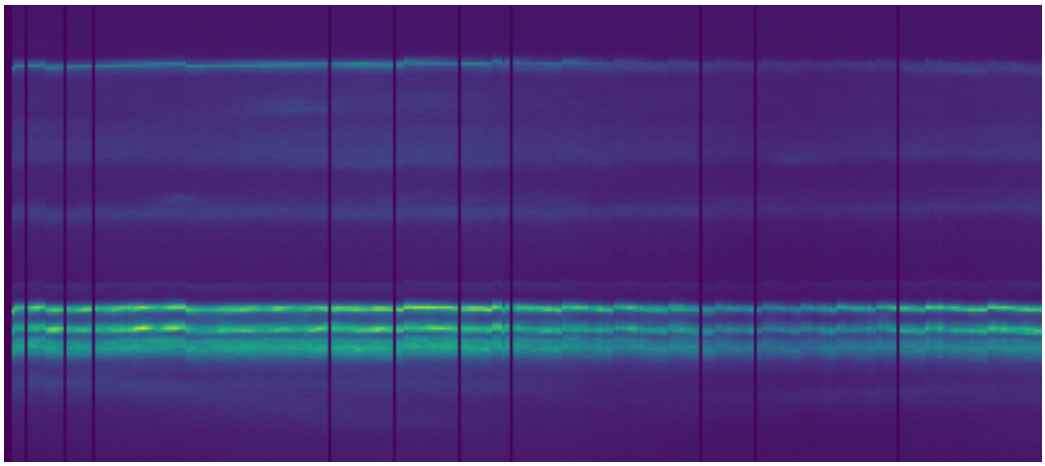
Figure 3.4: Comparison of retina maps generated by R-SLAM and a segmentation-based method [14]. Both maps are produced by stabilizing and averaging all the input volumes. Figures 3.4a and 3.4c are projections of R-SLAM's map, and 3.4b and 3.4d are projections of the segmentation-based map. The slow B-scan projections are cropped to the photoreceptor layer for clarity. Both techniques perform similarly, with R-SLAM achieving slightly better visual quality by avoiding some of the vertical artifacts found in the segmentation-based map.



(a) Input volume 1

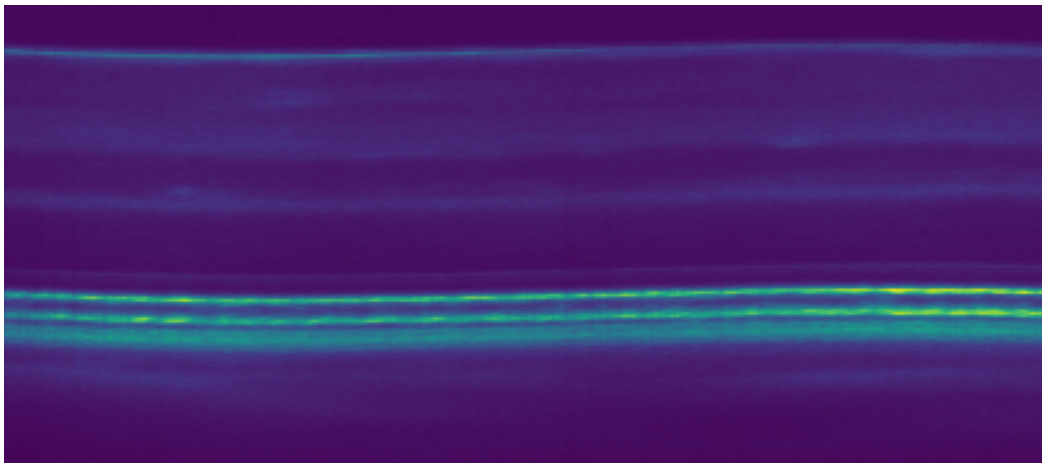


(b) R-SLAM stabilization

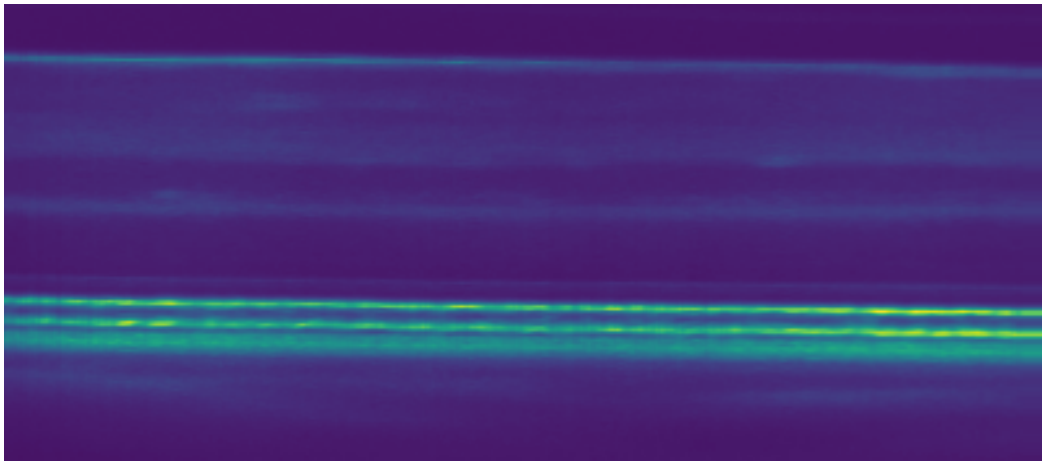


(c) [14] stabilization

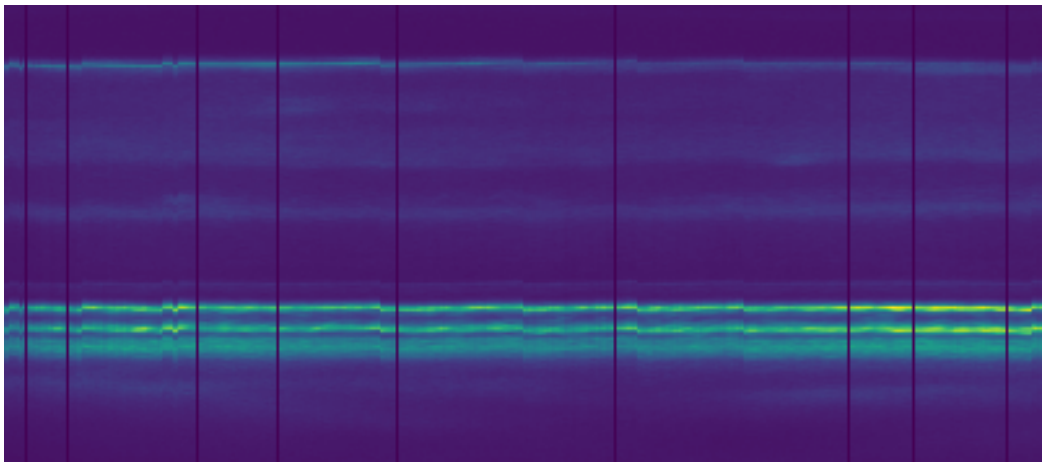
Figure 3.5: Comparison of R-SLAM and a segmentation-based method [14] at stabilizing distorted input volume 1. Shown are the slow B-scan projections of the input volume and the outputs of R-SLAM and [14]. R-SLAM does a better job of straightening out the photoreceptor layer, which is assumed to be locally planar. By contrast, the stabilized volume output by [14] contains sawtooth artifacts that appear as discontinuities in the photoreceptor layer.



(a) Input volume 2



(b) R-SLAM stabilization



(c) [14] stabilization

Figure 3.6: Comparison of R-SLAM and a segmentation-based method [14] at stabilizing distorted input volume 2. Shown are the slow B-scan projections of the input volume and the outputs of R-SLAM and [14]. As in figure 3.5, R-SLAM is more effective at flattening the photoreceptor layer, while the segmentation-based stabilization exhibits sawtooth artifacts.

Chapter 4

Conclusion

In this work, we have demonstrated that R-SLAM is effective at tracking and mapping the retina in both the 2D AOSLO and 3D AO-OCT domains. Our methods outperform prior work in this area, producing motion traces and retina maps that are highly accurate and contain enough detail to resolve fine cellular structure. Through extensive evaluation, we validate that our optimization-based approach is able to solve the underconstrained inverse problem of estimating the retina's map and motion from retina recordings, indicating the efficacy of this inverse approach as a framework for reasoning about retinal imaging. R-SLAM is both theoretically sound, relying on a probabilistic optimization algorithm that is proven to be convex, as well as efficient in practice, with robust methods for feature tracking and fast solvers for the convex optimization procedure.

R-SLAM is not without its limitations. We only test our method on data acquired during fixation, in which the subject focuses on a target and only exhibits smaller movements such as drift, tremor, and microsaccades. Another limitation of our method is that we model the retina as a planar surface, which is approximately true for the portion of the retina imaged during fixational acquisitions but does not hold for larger fields of view. Further research should be done to extend our approach to handle saccades and, more generally, retina recordings in which the subject is free to gaze in any direction.

These limitations are not inherent issues with R-SLAM, but rather the result of practical choices we make to process retinal imagery acquired during fixation. Future work can be done to extend R-SLAM to handle larger saccades and retinal curvature, for example, while still working within the same framework of inverse reconstruction. The novelty of R-SLAM is that it offers an optimization-based technique for performing retina tracking and mapping at higher accuracy than seen in prior work. Our method enables the collection of ground truth eye motion traces that offer useful information about the behavior of the human visual system, and it opens up new possibilities for the cellular-scale display technologies of the future.

Bibliography

[1] Akshay Agrawal et al. “A rewriting system for convex optimization problems”. In: *Journal of Control and Decision* 5.1 (2018), pp. 42–60.

[2] Mehdi Azimipour et al. “Intraframe motion correction for raster-scanned adaptive optics images using strip-based cross-correlation lag biases”. In: *PLOS ONE* 13.10 (Oct. 2018). Ed. by Phillip Bedggood, e0206052. DOI: 10.1371/journal.pone.0206052. URL: https://doi.org/10.1371/journal.pone.0206052.

[3] S. Baker et al. “Removing rolling shutter wobble”. In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2010, pp. 2392–2399. DOI: 10.1109/CVPR.2010.5539932.

[4] Phillip Bedggood and Andrew Metha. “De-warping of images and improved eye tracking for the scanning laser ophthalmoscope”. In: *PLOS ONE* 12.4 (Apr. 2017). Ed. by Marinko Sarunic, e0174617. DOI: 10.1371/journal.pone.0174617. URL: https://doi.org/10.1371/journal.pone.0174617.

[5] Norick R. Bowers, Alexandra E. Boehm, and Austin Roorda. “The effects of fixational tremor on the retinal image”. In: *Journal of Vision* 19.11 (Sept. 2019), p. 8. DOI: 10.1167/19.11.8. URL: https://doi.org/10.1167/19.11.8.

[6] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. DOI: 10.1017/CBO9780511804441.

[7] Steven Diamond and Stephen Boyd. “CVXPY: A Python-embedded modeling language for convex optimization”. In: *Journal of Machine Learning Research* 17.83 (2016), pp. 1–5.

[8] Nhan Hieu Do. *Parallel processing for adaptive optics optical coherence tomography (AO-OCT) image registration using GPU*. 2016. DOI: 10.7912/C2QS31. URL: http://hdl.handle.net/1805/10904.

[9] R. Engbert et al. “An integrated model of fixational eye movements and microsaccades”. In: *Proceedings of the National Academy of Sciences* 108.39 (Aug. 2011), E765–E770. DOI: 10.1073/pnas.1102730108. URL: https://doi.org/10.1073/pnas.1102730108.

[10] JM Findlay. “Frequency analysis of human involuntary eye movement”. In: *Kybernetik* 8.6 (1971), pp. 207–214.

[11] Martin A. Fischler and Robert C. Bolles. “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography”. In: *Commun. ACM* 24.6 (June 1981), pp. 381–395. ISSN: 0001-0782. DOI: 10.1145/358669.358692. URL: https://doi.org/10.1145/358669.358692.

[12] Matthias Grundmann et al. “Calibration-free rolling shutter removal”. In: *2012 IEEE International Conference on Computational Photography (ICCP)*. IEEE, Apr. 2012. DOI: 10.1109/iccphot.2012.6215213. URL: https://doi.org/10.1109/iccphot.2012.6215213.

[13] J.P. Lewis. “Fast Normalized Cross-Correlation”. In: *Ind. Light Magic* 10 (Oct. 2001).

[14] Zhenghan Li et al. “Correcting intra-volume distortion for AO-OCT using 3D correlation based registration”. In: *Opt. Express* 28.25 (Dec. 2020), pp. 38390–38409. DOI: 10.1364/OE.410374. URL: http://opg.optica.org/oe/abstract.cfm?URI=oe-28-25-38390.

[15] Jorge Otero-Millan, Stephen L. Macknik, and Susana Martinez-Conde. “Fixational eye movements and binocular vision”. In: *Frontiers in Integrative Neuroscience* 8 (July 2014). DOI: 10.3389/fnint.2014.00052. URL: https://doi.org/10.3389/fnint.2014.00052.

[16] Vimal Prabhu Pandiyan et al. “High-speed adaptive optics line-scan OCT for cellular-resolution optoretinography”. In: *Biomed. Opt. Express* 11.9 (Sept. 2020), pp. 5274–5296. DOI: 10.1364/BOE.399034. URL: http://opg.optica.org/boe/abstract.cfm?URI=boe-11-9-5274.

[17] Vimal Prabhu Pandiyan et al. “Reflective mirror-based line-scan adaptive optics OCT for imaging retinal structure and function”. In: *Biomedical Optics Express* 12.9 (Aug. 2021), p. 5865. DOI: 10.1364/boe.436337. URL: https://doi.org/10.1364/boe.436337.

[18] Vimal Prabhu Pandiyan et al. “The optoretinogram reveals the primary steps of phototransduction in the living human eye”. In: *Science Advances* 6.37 (2020), eabc1124. DOI: 10.1126/sciadv.abc1124. eprint: https://www.science.org/doi/pdf/10.1126/sciadv.abc1124. URL: https://www.science.org/doi/abs/10.1126/sciadv.abc1124.

[19] Adam Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[20] Austin Roorda et al. “Adaptive optics scanning laser ophthalmoscopy”. In: *Opt. Express* 10.9 (May 2002), pp. 405–412. DOI: 10.1364/OE.10.000405. URL: http://www.opticsexpress.org/abstract.cfm?URI=oe-10-9-405.

[21] Christy K. Sheehy et al. “High-speed, image-based eye tracking with a scanning laser ophthalmoscope”. eng. In: *Biomedical optics express* 3.10 (Oct. 2012). 173117[PII], pp. 2611–2622. ISSN: 2156-7085. DOI: 10.1364/BOE.3.002611. URL: https://doi.org/10.1364/BOE.3.002611.

[22] Jay Shenoy et al. “R-SLAM: Optimizing Eye Tracking From Rolling Shutter Video of the Retina”. In: *ICCV*. 2021.

[23] Scott B. Stevenson and Austin Roorda. “Correcting for miniature eye movements in high resolution scanning laser ophthalmoscopy”. In: *Ophthalmic Technologies XV*. Ed. by Fabrice Manns et al. Vol. 5688. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series. Apr. 2005, pp. 145–151. DOI: 10.1117/12.591190.

[24] Sebastian Thrun and John J. Leonard. “Simultaneous Localization and Mapping”. In: *Springer Handbook of Robotics*. Ed. by Bruno Siciliano and Oussama Khatib. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 871–889. ISBN: 978-3-540-30301-5. DOI: 10.1007/978-3-540-30301-5_38. URL: https://doi.org/10.1007/978-3-540-30301-5_38.

[25] Yiyi Wang et al. “Human foveal cone photoreceptor topography and its dependence on eye length”. In: *Elife* 8 (2019), e47148.

[26] Qiang Yang et al. “Design of an integrated hardware interface for AOSLO image capture and cone-targeted stimulus delivery”. In: *Opt. Express* 18.17 (Aug. 2010), pp. 17841–17858. DOI: 10.1364/OE.18.017841. URL: http://www.opticsexpress.org/abstract.cfm?URI=oe-18-17-17841.

[27] B. Zhuang, L. Cheong, and G. Lee. “Rolling-Shutter-Aware Differential SfM and Image Rectification”. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. Los Alamitos, CA, USA: IEEE Computer Society, Oct. 2017, pp. 948–956. DOI: 10.1109/ICCV.2017.108. URL: https://doi.ieeecomputersociety.org/10.1109/ICCV.2017.108.

[28] B. Zhuang et al. “Learning Structure-And-Motion-Aware Rolling Shutter Correction”. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 4546–4555. DOI: 10.1109/CVPR.2019.00468.

Appendix A

Additional Experiments and Proofs

A.1 Additional Experiments

We provide further evidence of how R-SLAM achieves better tracking quality over prior work, both with real AOSLO video and in simulation. Since real videos lack ground truth motion data, we use the quality of the generated map as a proxy for tracking performance since map artifacts indicate poor tracking. With simulated videos, we directly compare each method's output motion trace against the synthetic ground truth trace.

A.1.1 Offline Map Generation With Real AOSLO Video

We compare the retina maps output by R-SLAM with those of other offline mapping techniques in figures A.1, A.2, and A.3. Each method is allowed to drop frame data when generating maps because we wish to evaluate map quality as opposed eye tracking quality. The three input videos are real AOSLO recordings published by Wang *et al*. [25], which are attached to this supplementary submission. In each case, R-SLAM outperforms both Bedggood *et al*. [4] and Stevenson [23] in producing maps that are artifact-free and contain clear cone cells. The method of Azimipour *et al*. [2] is excluded because it is nearly identical to that of Bedggood *et al*. [4].

A.1.2 Real-time Tracking on Simulated Video

We offer additional visualizations of the performance of various real-time AOSLO trackers on simulated video. Each real-time tracker differs in how it generates the reference image of the retina. Figures A.4 and A.5 show the real-time tracking performance of R-SLAM, Bedggood *et al*. [4], and Stevenson [23] on two videos, evaluated against the ground truth. In each case, R-SLAM more faithfully estimates the ground truth motion.

A.2 Proof of Convexity for Motion Initialization Objective

We prove that the objective in Equation 2.4 is convex by showing that the optimization function can be expressed as a Tikhonov regularized problem [6], which is known to be convex:

$$\min_{\mathbf{x}} ||\mathbf{Ax} - \mathbf{b}||^2 + ||\mathbf{Dx}||^2 \tag{A.1}$$

Where $\mathbf{A}$ and $\mathbf{D}$ are matrices and $\mathbf{b}$ is a vector. $\mathbf{x}$ is a vector which we are optimizing.

The loss functions from the main paper are reproduced here for clarity:

$$L(M, G) = \sum_{j=1}^{|G|-1} ||(M(t_j) - M(t_{j+1})) - (p_{j+1} - p_j)||^2 \tag{A.2}$$

$$L(M) = \sum_{i=1}^{|T|-1} \frac{||M(t_{i+1}) - M(t_i)||^2}{t_{i+1} - t_i} \tag{A.3}$$

$$\lambda_B L(M) + \lambda_T \sum_{G \in \mathcal{G}} L(M, G) \tag{A.4}$$

With Equation A.4 being the function we want to reduce to the form given in Equation A.1. The first term will turn into the regularization $||\mathbf{Dx}||^2$ and the second term will turn into the least-squares $||\mathbf{Ax} - \mathbf{b}||^2$.

The following subsections illustrate how this is done.

A.2.1 Proof that the first term is reducible to $||\mathbf{Dx}||^2$

M is a vector-valued function, but we will split it into its X and Y components to make this proof easier to follow. Split $L(M)$ into $L(M) = L_X(M) + L_Y(M)$, where $L_X(M)$ and $L_Y(M)$ operate on the individual components X and Y which make up M:

$$L_X(M) = \sum_{i=1}^{|T|-1} \frac{[X(t_{i+1}) - X(t_i)]^2}{t_{i+1} - t_i} \tag{A.5}$$

$$L_Y(M) = \sum_{i=1}^{|T|-1} \frac{[Y(t_{i+1}) - Y(t_i)]^2}{t_{i+1} - t_i} \tag{A.6}$$

We will only examine $L_X(M)$ in detail, but the same reasoning applies to $L_Y(M)$.

Let $x \in \mathbb{R}^{|T|}$ be a vector. Let the ith element of x be $X(t_i)$. We can therefore rewrite L_X as follows:

$$L_X(x) = \sum_{i=1}^{|T|-1} \frac{[x_{i+1} - x_i]^2}{t_{i+1} - t_i} \tag{A.7}$$

Let $d(t_a, t_b) \in \mathbb{R}^{|T|}$ be a vector of zeros, except with a 1 in the index where element $X(t_a)$ appears in x and a -1 where element $X(t_b)$ appears in x. Conceptually, d is a vector which lets us express a scalar *difference* as a vector dot product: $X(t_a) - X(t_b) = d^\top x$.

Then we can rewrite L_X as follows:

$$L_X(x) = \sum_{i=1}^{|T|-1} \frac{(d(t_{i+1}, t_i)^\top x)^2}{t_{i+1} - t_i} \tag{A.8}$$

Expanding terms and rearranging:

$$L_X(x) = \sum_{i=1}^{|T|-1} \frac{x^\top d(t_{i+1}, t_i) d(t_{i+1}, t_i)^\top x}{t_{i+1} - t_i} \tag{A.9}$$

$$L_X(x) = x^\top \left(\sum_{i=1}^{|T|-1} \frac{d(t_{i+1}, t_i) d(t_{i+1}, t_i)^\top}{t_{i+1} - t_i} \right) x \tag{A.10}$$

Collapsing the inner summation into a single matrix, we are left with:

$$L_X(x) = x^\top H_X x \tag{A.11}$$

where $H_X \in \mathbb{R}^{|T| \times |T|}$ is a symmetric matrix.

We can apply the same process to L_Y with $y \in \mathbb{R}^{|T|}$ where $y_i = Y(t_i)$:

$$L_Y(y) = y^\top H_Y y \tag{A.12}$$

Combining terms into block matrix form:

$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix} \tag{A.13}$$

$$\mathbf{H} = \begin{bmatrix} H_X & 0 \\ 0 & H_Y \end{bmatrix} \tag{A.14}$$

We have

$$\lambda_B L(M) = \lambda_B \mathbf{x}^\top H \mathbf{x} \tag{A.15}$$

Notice that since $L(M) \geq 0$ by Equation A.3, this shows that H is positive semidefinite. Therefore, let $D' = H^{1/2}$ be the matrix square root of H (such that $H^\top H = D'$). Let $\mathbf{D} = \sqrt{\lambda_B} D'$. We have:

$$\lambda_B L(M) = ||\mathbf{D}\mathbf{x}||^2 \tag{A.16}$$

as desired.

A.2.2 Proof that the second term is reducible to $||\mathbf{A}\mathbf{x} - \mathbf{b}||^2$

This section will proceed similar to Section A.2.1. We will analyze each X and Y component separately, then re-combine into block matrix form to yeild the final result.

We will split $L(M, G)$ into $L(M, G) = L_X(M, G) + L_Y(M, G)$:

$$L_X(M, G) = \sum_{j=1}^{|G|-1} [(X(t_j) - X(t_{j+1})) - (u_{j+1} - u_j)]^2 \tag{A.17}$$

$$L_Y(M, G) = \sum_{j=1}^{|G|-1} [(Y(t_j) - Y(t_{j+1})) - (v_{j+1} - v_j)]^2 \tag{A.18}$$

Again, we will only examine $L_X(M, G)$ in detail, but the same reasoning applies to $L_Y(M, G)$.

Let x and d be defined as in Section A.2.1. We have:

$$L_X(M, G) = \sum_{j=1}^{|G|-1} [d(t_j, t_{j+1})^\top x - c_j]^2 \tag{A.19}$$

where $c_j = u_{j+1} - u_j$.

Let $q_G \in \mathbb{R}^{|G|-1}$ be a vector where the jth element is c_j. Similarly, let $K_G \in \mathbb{R}^{(|G|-1)\times|T|}$ be a matrix where the jth row is $d(t_j, t_{j+1})$. Then we have:

$$L_X(M, G) = ||K_G x - q_G||^2 \tag{A.20}$$

by construction.

Let $\mathcal{G}$ have some arbitrary ordering, so that G_k is the kth element of $\mathcal{G}$.

Then, in block matrix form define the following matrix:

$$P_X = \begin{bmatrix} K_{G_1} \\ K_{G_2} \\ \vdots \\ K_{G_{|\mathcal{G}|}} \end{bmatrix} \tag{A.21}$$

Similarly, define the following vector:

$$b_X = \begin{bmatrix} q_{G_1} \\ q_{G_2} \\ \vdots \\ q_{G_{|\mathcal{G}|}} \end{bmatrix} \tag{A.22}$$

We have:

$$\sum_{G \in \mathcal{G}} L_X(M, G) = ||P_X x - b_X||^2 \tag{A.23}$$

Define $\mathbf{x}$ as before. We define P and b as follows in block matrix form:

$$P = \begin{bmatrix} P_X & 0 \\ 0 & P_Y \end{bmatrix} \tag{A.24}$$

$$b = \begin{bmatrix} b_X \\ b_Y \end{bmatrix} \tag{A.25}$$

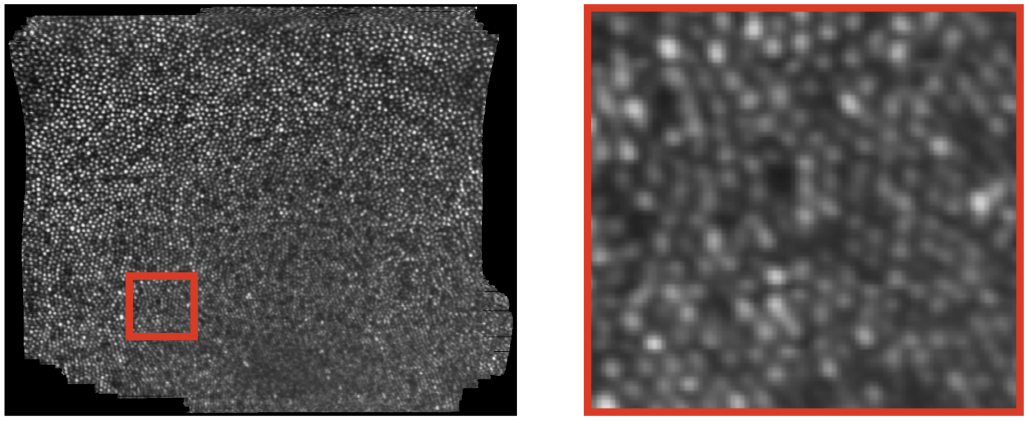
Therefore:

$$\sum_{G \in \mathcal{G}} L(M, G) = ||P\mathbf{x} - b||^2 \tag{A.26}$$

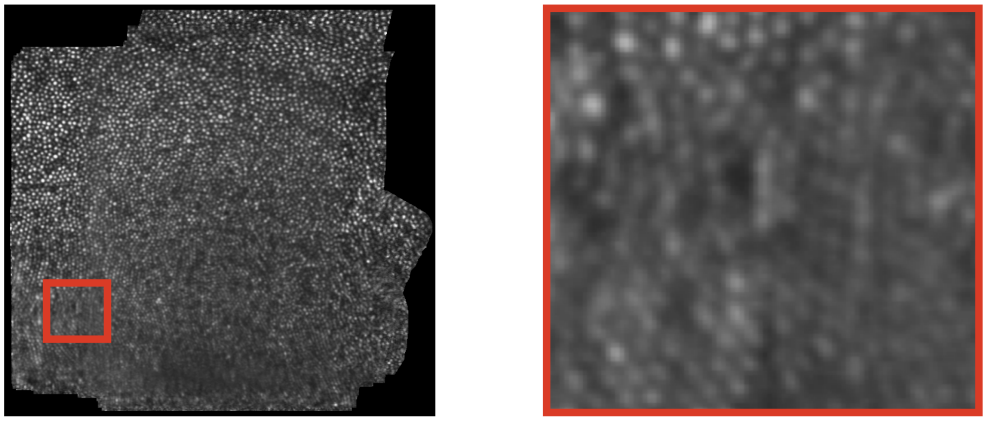
Taking $\mathbf{A} = \sqrt{\lambda_T} P$ and $\mathbf{b} = \sqrt{\lambda_T} b$, we have:

$$\lambda_T \sum_{G \in \mathcal{G}} L(M, G) = ||\mathbf{A}\mathbf{x} - \mathbf{b}||^2 \tag{A.27}$$

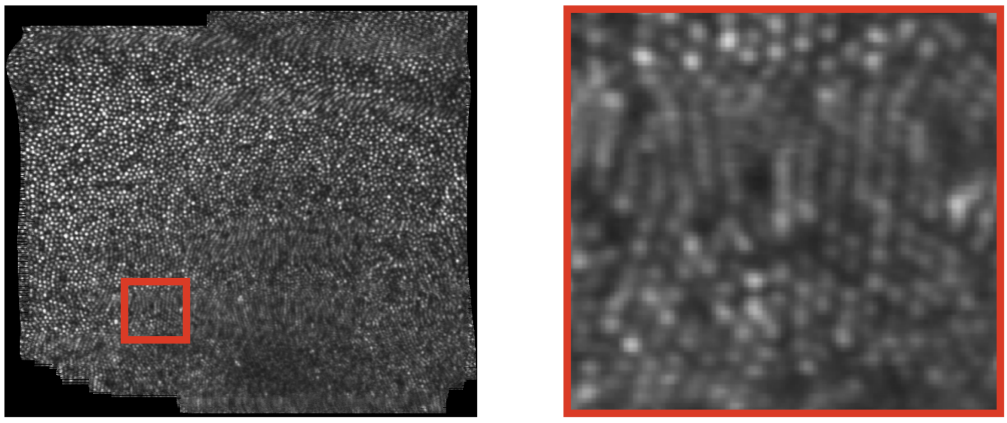
as desired.



R-SLAM (Ours)

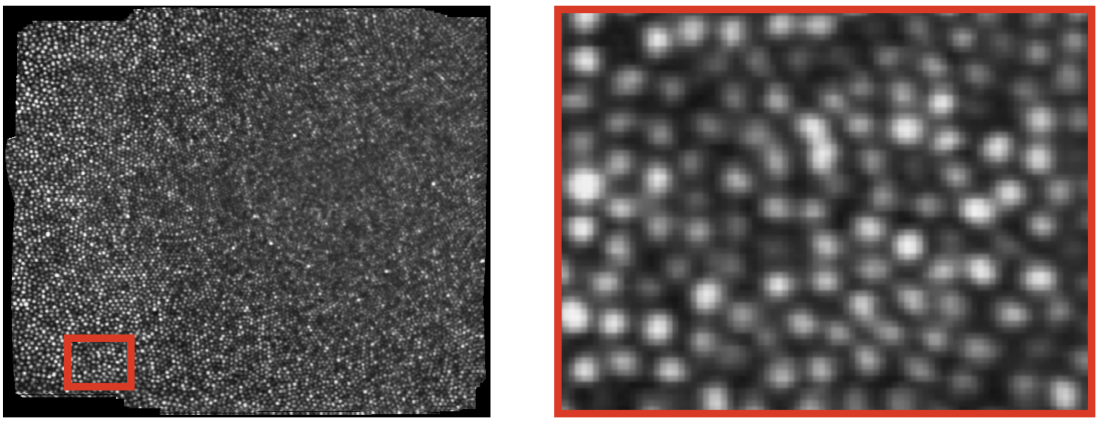


Bedggood

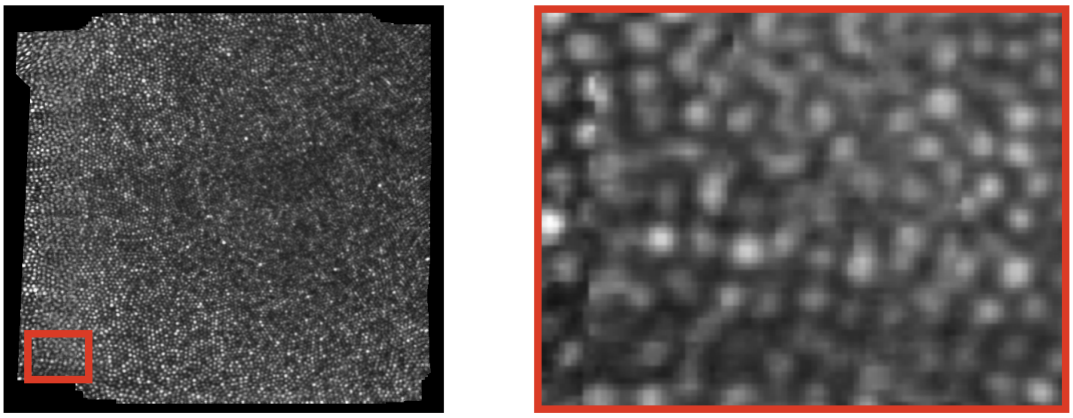


Stevenson

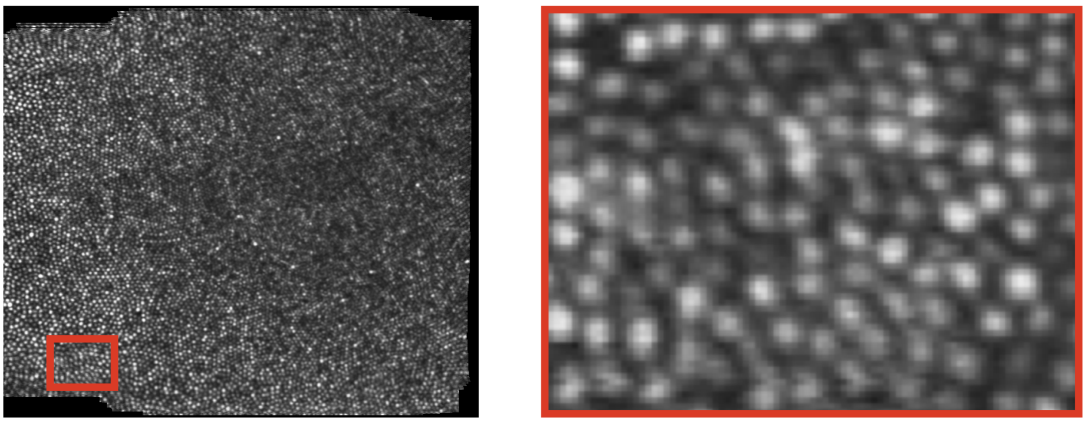
Figure A.1: R-SLAM does better than Bedggood *et al.* [4] at clearly resolving all the cone cells. R-SLAM also avoids the sorts of horizontal seam artifacts found in the map output by Stevenson [23].



R-SLAM (Ours)

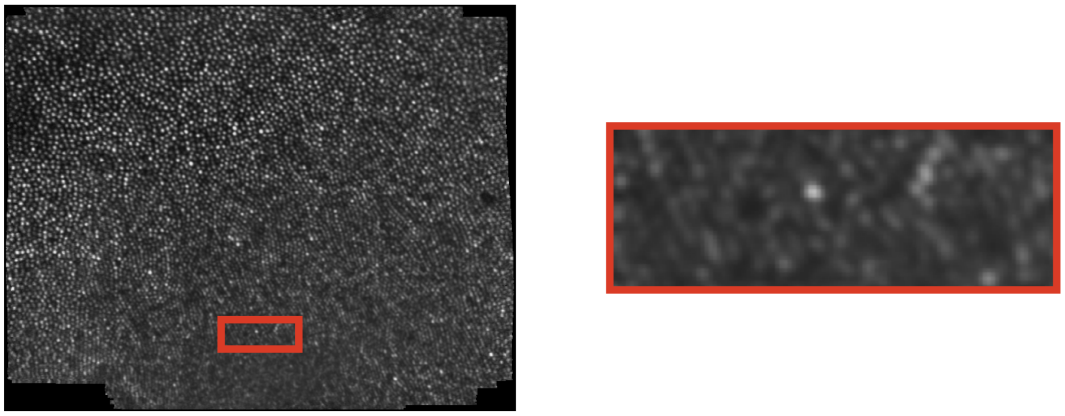


Bedggood

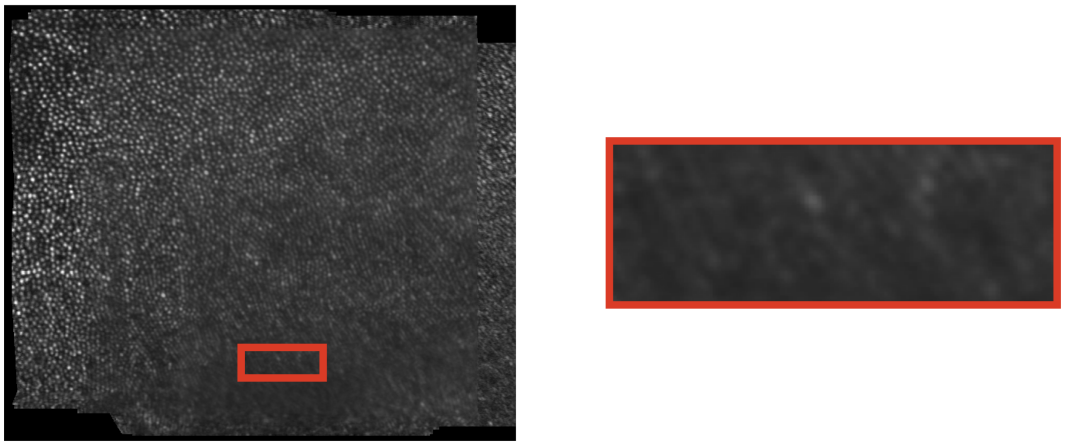


Stevenson

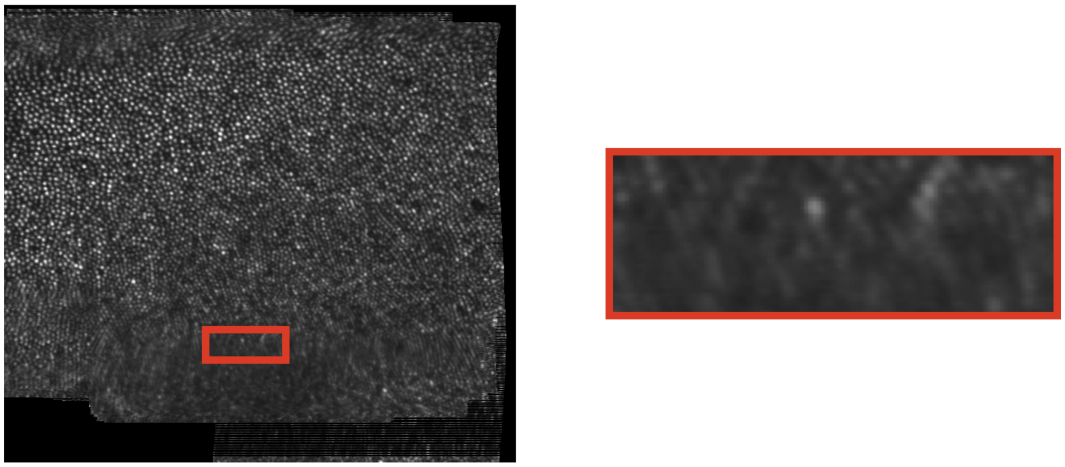
Figure A.2: R-SLAM avoids the vertical seams and blurry artifacts found in the map output by Bedggood [4]. R-SLAM's map is also far less noisy than that of Stevenson [23]. Note that these maps are the same as those found in figure 2.4.



R-SLAM (Ours)



Bedggood



Stevenson

Figure A.3: R-SLAM does a better job of resolving cone cells and removing blurry distortions, particularly near the bottom of the zoomed inset. The region displayed in the inset is close to the foveal center, where the cones are the smallest. Accurate registration is critical within the fovea in order to achieve adequate resolution of cone cells.

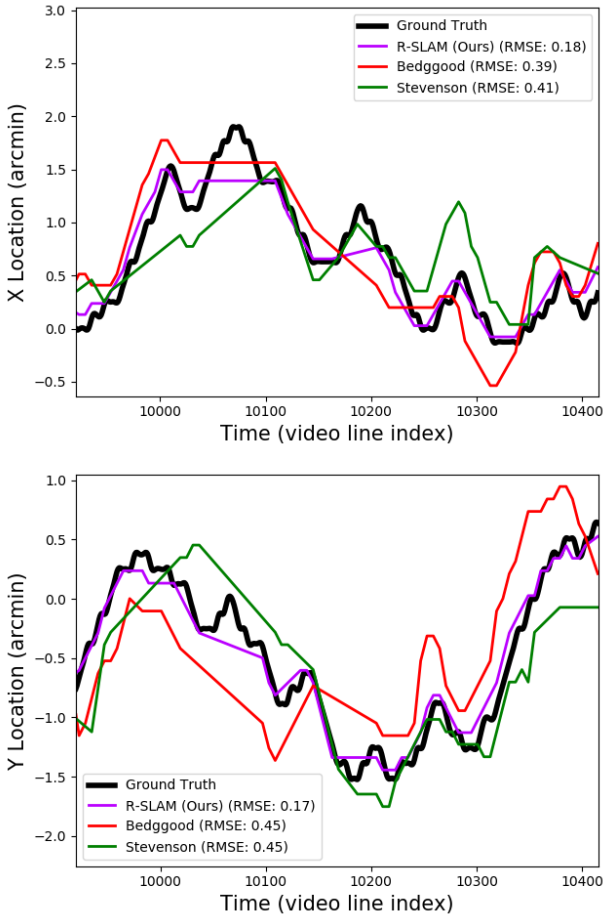


Figure A.4: Comparison of real-time trackers on a simulated input video with a synthetic motion trace. R-SLAM achieves lower root-mean-square error (RMSE) than the other two methods in both directions. Visually, R-SLAM is closer to the synthetic ground truth motion (black), which is more apparent in the vertical (y) direction.

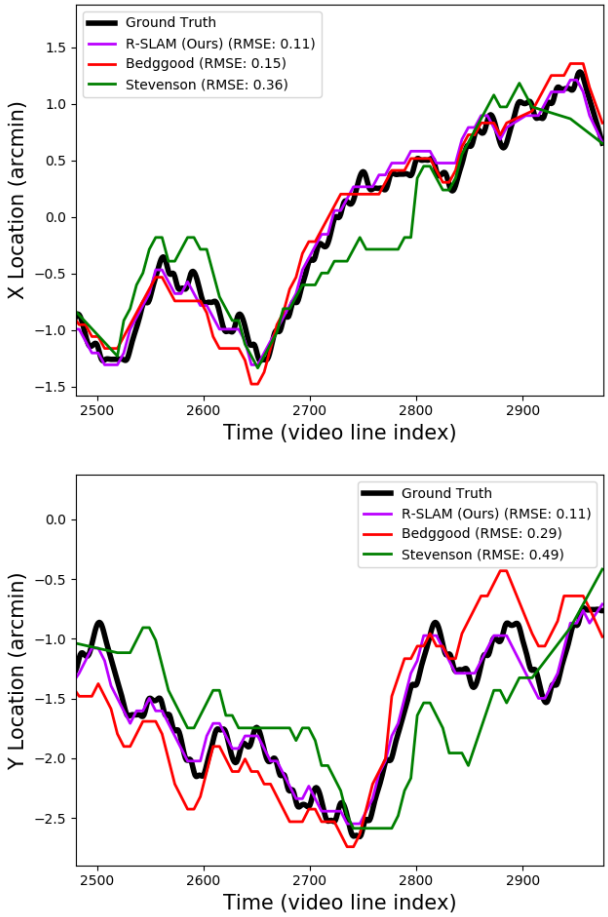


Figure A.5: Comparison of real-time trackers on a second simulated input video with a synthetic motion trace. R-SLAM outperforms the other two methods in terms of root-mean-square error (RMSE) from the ground truth. While R-SLAM's motion trace looks visually similar to that of Bedggood [4] in the horizontal (x) direction, R-SLAM visibly outperforms Bedggood in the vertical direction.