R-SLAM: Optimizing Eye Tracking from Rolling Shutter Video of the Retina

Jay Shenoy*, James Fong*, Jeffrey Tan, Austin Roorda, and Ren Ng
University of California, Berkeley
{jayshenoy, james.fong, tanjeffreyz02, aroorda, ren}@berkeley.edu
* Equal contribution.

Abstract

We present a method for optimization-based recovery of eye motion from rolling shutter video of the retina. Our approach formulates eye tracking as an optimization problem that jointly estimates the retina's motion and appearance using convex optimization and a constrained version of gradient descent. By incorporating the rolling shutter imaging model into the formulation of our joint optimization, we achieve state-of-the-art accuracy both offline and in real-time. We apply our method to retina video captured with an adaptive optics scanning laser ophthalmoscope (AOSLO), demonstrating eye tracking at 1 kHz with accuracies below one arcminute—over an order of magnitude higher than conventional eye tracking systems.

1. Introduction

Eye tracking is the process of determining the eye's gaze direction over time, often using specialized optics and software. In the real-time setting, precise tracking is hindered by the constant and often ballistic motion of the eye, consisting of drift, microsaccades, and saccades [22, 25]. Studying these ballistic movements in the offline setting requires accurate, high-frequency motion estimation algorithms, which are useful in ophthalmology and biomedicine. A precise model of these high-frequency movements is also relevant to computer vision because it could inform the development of new algorithms that aim to replicate cognitive tasks such as object recognition and scene understanding; for instance, microsaccades have been linked to complex visual tasks in humans like reading [11]. Unfortunately, most eye tracking systems are only accurate to 0.5 degrees of visual angle, making the exact dynamics of microsaccades and saccades an open question.

Eye trackers that infer gaze direction using measurements from the eye's cornea, pupil and lens have both accuracy and precision limitations due to their reliance on individual subjective calibration procedures, and from gaze estimation errors caused by changes in pupil size or wobble of the crystalline lens.

In principle, eye tracking approaches based on imaging the retina can overcome these limits. Further, they add information by linking gaze with retinal structures. The adaptive optics scanning laser ophthalmoscope (AOSLO) is a device that images the retina at high resolution, with current systems capturing 30 FPS rolling shutter video 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 [35]. 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 rolling shutter capture process.

In this paper, 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 [33]), 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 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.

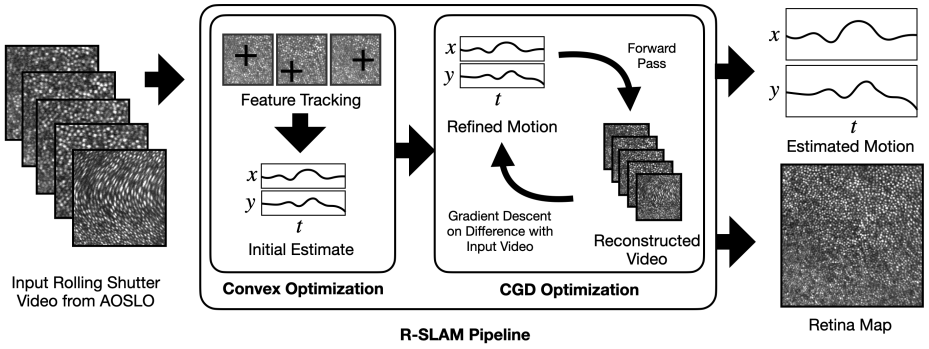

Figure 1: Our offline tracking algorithm's pipeline. R-SLAM receives as input distorted video of the retina, then proceeds to compute an initial motion estimate using convex optimization. We then use constrained gradient descent to jointly optimize the retina's motion and map, the latter of which can be used for real-time tracking.

- 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.

2. Related Work

2.1. Conventional Eye Tracking Systems

Eye trackers that infer gaze from the pupil position or the pupil position relative to the corneal reflection are inexpensive, compact, and convenient, but are typically accurate to only 0.5 degrees of visual angle. Most modern systems perform real-time pupil fitting and gaze tracking in a simultaneous fashion [26]. For instance, Tobii is a commercial eye tracker that estimates eye position and gaze direction by shining near-infrared light onto the pupil, with accuracies between 30-66 arcminutes [4, 17]. Similarly, Pupil Labs offers a video-based eye tracker that uses computer vision to fit a 3D model of the eye for tracking, accurate to about 1 degree (60 arcminutes) of visual angle [1, 2]. Both Tobii and Pupil Labs operate at frequencies under 250 Hz [4, 2], precluding the precise tracking of ballistic eye movements like saccades and microsaccades. The EyeLink achieves real-time tracking speeds of 2000 Hz using a video-based system [3]. Recent work by Angelopoulos *et al*. [6] beats the EyeLink by pushing real-time tracking speeds to 10,000 Hz using near-eye event cameras. However, as with Pupil and Tobii, both the EyeLink and the work of Angelopoulos *et al*. offer tracking accuracies of 27 arcminutes at best [6, 16].

Dual Purkinje image (DPI) trackers track eye movements by comparing the relative location of the corneal reflex with a reflection from the back surface of the crystalline lens [13]. DPI systems are more precise than pupil trackers, but are still limited in accuracy owing to the need for individual subjective calibration. It has been reported that iterative subjective calibration procedures can be implemented to generate DPI accuracies of around 2 arcminutes [28], but no amount of calibration can prevent artifacts from lens movement inside the eye [10, 14].

Other eye trackers include electrooculography [24], scleral search coils [29] and optical lever methods in which the deflection of a laser light is measured from a small mirror that is fixed to the eye via a specialized contact lens [31]. Because of the invasiveness and infrequent use of these methods, they will not be discussed further or compared with the R-SLAM approach.

2.2. 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*. [30], 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 [12, 32].

There are several techniques that attempt to dewarp rolling shutter AOSLO video. Stevenson *et al*. [32] per-

form 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*. [7] 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*. [9] use a similar method to [7], 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) [10]. We make use of this phenomenon in analyzing tracking error in the absence of ground truth motion data in Section 4.2. 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.

2.3. Rolling Shutter Correction for Frame Dewarping

A variety of algorithms exist to correct for rolling shutter, but most of them assume 3D world geometry [21, 36, 37]. Baker *et al*. [8] 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 [8] 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 [8] uses an l_1 loss to remove outliers from the set of tracked features (which we handle using RANSAC [20]). Thirdly, we refine the initial output of the convex step using gradient-based optimization over a different objective function.

3. 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. One unit of time is the inverse of the AOSLO's FPS.

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 $u, v \in [0, 1]$.

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:

$$\hat{M}, \hat{R} = \underset{M,R}{\arg\min} ||F(M, R) - V||^2 \quad (1)$$

3.1. Conceptual Overview

Equation 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 3.4.

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 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 3.2. Surprisingly, this convex optimization not only finds a good $\hat{M}_0$ to initialize CGD with, but it

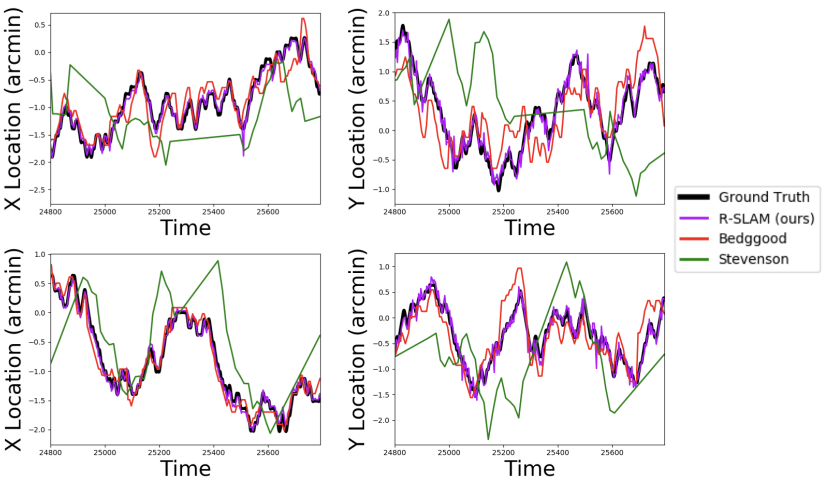

Figure 2: (Left) Comparison of offline tracking techniques on two different simulated AOSLO videos. Note that the simulated motion was set to a high level to stress-test all methods. R-SLAM, Azimipour *et al*., and Bedggood *et al*. are able to track the motion and offer a fair comparison, but the Stevenson *et al*. algorithm was not suited to track this magnitude of motion. Stevenson was able to track the real AOSLO videos (see Table 1) albeit with evident reference frame artifacts (Figure 4). R-SLAM achieves the most faithful reconstruction of the ground truth motion, particularly in the vertical (y) direction.

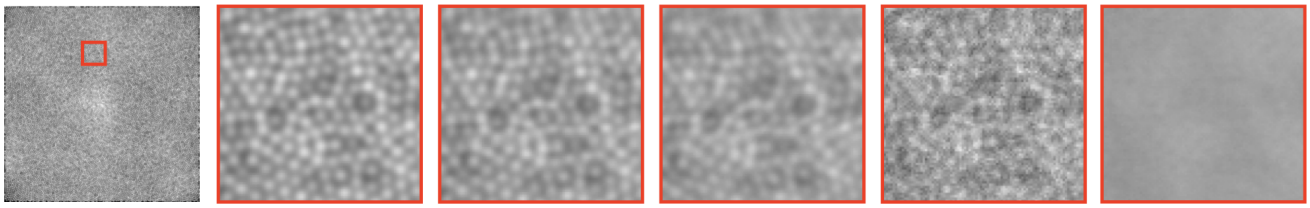

Figure 3: Comparison of different techniques for offline estimation of the retina map from simulated retina video. Stevenson fails to stabilize the input video, Azimipour (using only one frame) contains significant noise because it only stabilizes a single frame, and Bedggood *et al*. (equivalent to 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.

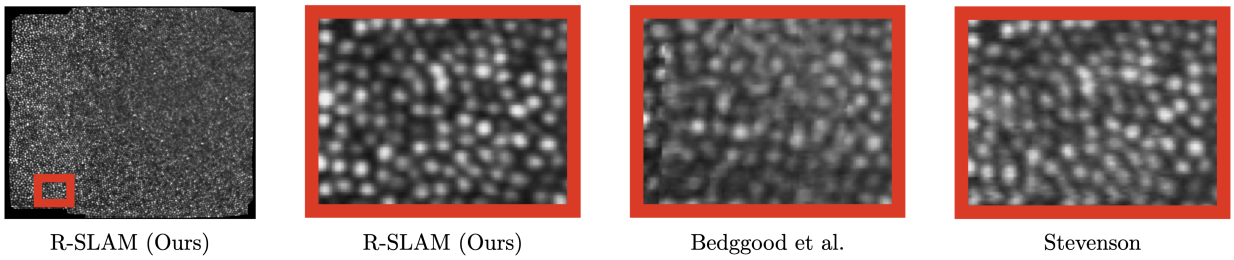

Figure 4: Comparison of different techniques for offline estimation of the retina map from real AOSLO video. Bedggood *et al*. suffers from blurry/distorted cone cells, and Stevenson contains sharpness issues and duplicated cone cells towards the bottom of the image. Only R-SLAM properly resolves all cone cells in the image.

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 3.2.1.

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 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 3.3.

3.2. 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 3.2.1. 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 5. We define the following loss for our estimate of M^* given a single $G \in \mathcal{G}$:

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

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:

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

By combining the inter-frame, tracking-based objective of Equation 2 with the intra-frame objective of Equation 3, we arrive at our overall convex optimization formulation:

$$\hat{M}_0 := \arg\min_M \lambda_B L(M) + \lambda_T \sum_{G \in \mathcal{G}} L(M, G) \tag{4}$$

Here, $\lambda_B, \lambda_T \geq 0$ are hyperparameter weights. Equation 4 is a convex quadratic program (QP) with no constraints, and so we can readily use existing convex optimization software packages [5, 15] to compute an optimal solution for $\hat{M}_0$. A proof of convexity is provided in the supplementary material. $\hat{M}_0$ is a discrete motion trace. We form a continuous representation of $\hat{M}_0$ via linear interpolation.

3.2.1 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 normalized cross-correlation [23] 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 un-

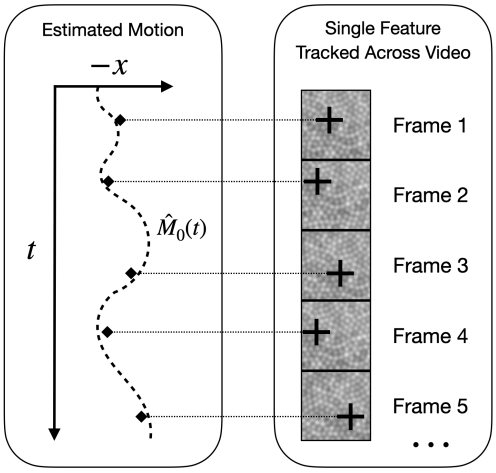

Figure 5: The relationship between tracked video features and eye motion trace, used as a constraint in our convex optimization formulation (Equation 4). Left: horizontal position of the retina as a function of time. The noisy motion samples from a single feature are shown as diamonds. The dashed curve 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.

derlying 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.

3.3. 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. This process is visualized at time 1:40 in the supplementary video.

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

R-SLAM jointly estimates the retinal 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 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 1 is not sufficiently constrained. To remedy this, we expect the following to hold true for M^*, V^*:

$$||S(M^*, V) - R^*||^2 \leq \epsilon. \tag{5}$$

In Equation 5, ϵ serves as a measure of the noise in the process used record V. This constraint can be added to Equation 1 to produce the new optimization problem:

$$\begin{aligned} &\arg\min_{M,R} ||F(M, R) - V||^2 \\ &\text{s.t. } ||S(M, V) - R||^2 \leq \epsilon. \end{aligned} \tag{6}$$

Recall that M^*, R^* are the desired optimal solutions. To make the constraint in Equation 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 retinal 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:

$$\begin{aligned} &||S(M, V) - R||^2 \leq \epsilon = 0 \\ \implies &||S(M, V) - R||^2 = 0 \\ \implies &S(M, V) = R. \end{aligned} \tag{7}$$

We approximate Equation 6 with the new objective:

$$\arg\min_M ||F(M, S(M, V)) - V||^2. \tag{8}$$

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 8 ensures consistency between $\hat{M}$, $\hat{R}$, and V, enabling faster convergence. We iteratively optimize Equation 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 [27], 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).

3.5. 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 retinal map. We make two important improvements. First, we use a more accurate retinal 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 [20] to filter out outliers, which is more robust than determining strip registration quality directly with the peak values output by normalized cross-correlation.

4. 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.

4.1. 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 [7]. Then, we compute a pair of three-second videos per mosaic using artificial eye motion traces derived from the random walk model in [18], 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 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.

4.2. 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*. [34]. 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 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 2.2, 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) [19]. Similar-appearing spectral artifacts are reported in prior work [10, 32]. 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.

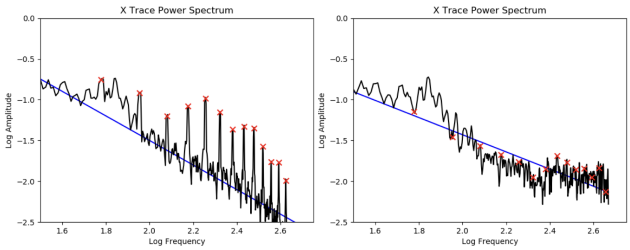

Figure 6: Comparison of power spectra of motion traces output by Stevenson [32] (left) and R-SLAM (right). In black is the power spectrum of the motion trace for a given video, in blue is the best linear regression fit, and in red are markers denoting the harmonics of 30 Hz (30 Hz, 60 Hz, 90 Hz, and 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.

	Method	Stevenson [32]	Bedggood *et al*. [9]	Azimipour *et al*. [7]	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 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 4.2). 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*. [7] 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.

5. Discussion

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

5.1. 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 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 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 1. Using the spectral error metric defined in Section 4.2, R-SLAM achieves the lowest error, which corresponds to the artifact-free power spectra shown in Figure 6.

5.2. 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*. [8] 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.

Acknowledgements

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

References

[1] Pupil Capture software. Pupil Labs, retrieved Mar 2021. https://docs.pupil-labs.com/core/software/pupil-capture.

[2] Pupil Labs VR/AR Add-On, Technical Specs & Performance. Pupil Labs, retrieved Mar 2021. https://pupil-labs.com/products/vr-ar/tech-specs/.

[3] Eyelink 1000 Plus, Aug 2020. SR Research, https://www.sr-research.com/eyelink-1000-plus/.

[4] HTC Vive Pro Eye - VR headset with eye tracking integration, Sep 2020. https://vr.tobii.com/integrations/htc-vive-pro-eye/.

[5] Akshay Agrawal, Robin Verschueren, Steven Diamond, and Stephen Boyd. A rewriting system for convex optimization problems. *Journal of Control and Decision*, 5(1):42–60, 2018.

[6] Anastasios N Angelopoulos, Julien NP Martel, Amit PS Kohli, Jorg Conradt, and Gordon Wetzstein. Event based, near eye gaze tracking beyond 10,000 Hz. *arXiv preprint arXiv:2004.03577*, 2020.

[7] Mehdi Azimipour, Robert J. Zawadzki, Iwona Gorczynska, Justin Migacz, John S. Werner, and Ravi S. Jonnal. Intraframe motion correction for raster-scanned adaptive optics images using strip-based cross-correlation lag biases. *PLOS ONE*, 13(10):e0206052, Oct. 2018.

[8] S. Baker, E. Bennett, S. B. Kang, and R. Szeliski. Removing rolling shutter wobble. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2392–2399, 2010.

[9] Phillip Bedggood and Andrew Metha. De-warping of images and improved eye tracking for the scanning laser ophthalmoscope. *PLOS ONE*, 12(4):e0174617, Apr. 2017.

[10] Norick R. Bowers, Alexandra E. Boehm, and Austin Roorda. The effects of fixational tremor on the retinal image. *Journal of Vision*, 19(11):8, Sept. 2019.

[11] Norick R. Bowers and Martina Poletti. Microsaccades during reading. *PLOS ONE*, 12(9):e0185180, Sept. 2017.

[12] Min Chen, Robert F. Cooper, Grace K. Han, James Gee, David H. Brainard, and Jessica I. W. Morgan. Multimodal automatic montaging of adaptive optics retinal images. *Biomedical Optics Express*, 7(12):4899, Nov. 2016.

[13] Hewitt D Crane and Carroll M Steele. Generation-V dual-Purkinje-image eyetracker. *Applied optics*, 24(4):527–537, 1985.

[14] Heiner Deubel and Bruce Bridgeman. Fourth Purkinje image signals reveal eye-lens deviations and retinal image distortions during saccades. *Vision research*, 35(4):529–538, 1995.

[15] Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

[16] Benedikt V. Ehinger, Katharina Groß, Inga Ibs, and Peter König. A new comprehensive eye-tracking test battery concurrently evaluating the pupil labs glasses and the EyeLink 1000. *PeerJ*, 7:e7086, July 2019.

[17] John Elvesjo, Marten Skogo, and Gunnar Elvers. Method and installation for detecting and following an eye and the gaze direction thereof, Aug 2009.

[18] R. Engbert, K. Mergenthaler, P. Sinn, and A. Pikovsky. An integrated model of fixational eye movements and microsaccades. *Proceedings of the National Academy of Sciences*, 108(39):E765–E770, Aug. 2011.

[19] JM Findlay. Frequency analysis of human involuntary eye movement. *Kybernetik*, 8(6):207–214, 1971.

[20] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981.

[21] Matthias Grundmann, Vivek Kwatra, Daniel Castro, and Irfan Essa. Calibration-free rolling shutter removal. In *2012 IEEE International Conference on Computational Photography (ICCP)*. IEEE, Apr. 2012.

[22] R. John Leigh and David S. Zee. *Neurology of Eye Movements*. Oxford University Press, 2015.

[23] J.P. Lewis. Fast normalized cross-correlation. *Ind. Light Magic*, 10, 10 2001.

[24] Michael F Marmor, Mitchell G Brigell, Daphne L McCulloch, Carol A Westall, and Michael Bach. ISCEV standard for clinical electro-oculography (2010 update). *Documenta Ophthalmologica*, 122(1):1–7, 2011.

[25] Susana Martinez-Conde, Stephen L. Macknik, and David H. Hubel. The role of fixational eye movements in visual perception. *Nature Reviews Neuroscience*, 5(3):229–240, Mar. 2004.

[26] Carlos H. Morimoto and Marcio R. M. Mimica. Eye gaze tracking techniques for interactive applications. *Comput. Vis. Image Underst.*, 98(1):4–24, Apr. 2005.

[27] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[28] Martina Poletti and Michele Rucci. A compact field guide to the study of microsaccades: Challenges and functions. *Vision research*, 118:83–97, 2016.

[29] David A Robinson. A method of measuring eye movement using a scleral search coil in a magnetic field. *IEEE Transactions on bio-medical electronics*, 10(4):137–145, 1963.

[30] Austin Roorda, Fernando Romero-Borja, William J. Donnelly III, Hope Queener, Thomas J. Hebert, and Melanie C.W. Campbell. Adaptive optics scanning laser ophthalmoscopy. *Opt. Express*, 10(9):405–412, May 2002.

[31] Robert M Steinman, Genevieve M Haddad, Alexander A Skavenski, and Diane Wyman. Miniature eye movement. *Science*, 181(4102):810–819, 1973.

[32] Scott B. Stevenson and Austin Roorda. Correcting for miniature eye movements in high resolution scanning laser ophthalmoscopy. In Fabrice Manns, Per G. Soederberg, Arthur

Ho, Bruce E. Stuck, and Michael Belkin, editors, *Ophthalmic Technologies XV*, volume 5688 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 145–151, Apr. 2005.

[33] Sebastian Thrun and John J. Leonard. *Simultaneous Localization and Mapping*, pages 871–889. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[34] Yiyi Wang, Nicolas Bensaid, Pavan Tiruveedhula, Jianqiang Ma, Sowmya Ravikumar, and Austin Roorda. Human foveal cone photoreceptor topography and its dependence on eye length. *Elife*, 8:e47148, 2019.

[35] Qiang Yang, David W. Arathorn, Pavan Tiruveedhula, Curtis R. Vogel, and Austin Roorda. Design of an integrated hardware interface for AOSLO image capture and cone-targeted stimulus delivery. *Opt. Express*, 18(17):17841–17858, Aug 2010.

[36] B. Zhuang, L. Cheong, and G. Lee. Rolling-shutter-aware differential sfm and image rectification. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 948–956, Los Alamitos, CA, USA, oct 2017. IEEE Computer Society.

[37] B. Zhuang, Q. Tran, P. Ji, L. Cheong, and M. Chandraker. Learning structure-and-motion-aware rolling shutter correction. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4546–4555, 2019.