FoveaCam: A MEMS Mirror-Enabled Foveating Camera

Brevin Tilmon, *Member, IEEE*, Eakta Jain, Silvia Ferrari, *Senior Member, IEEE*, Sanjeev Koppal, *Senior Member, IEEE*

Abstract—Most cameras today photograph their entire visual field. In contrast, decades of active vision research have proposed *foveating* camera designs, which allow for selective scene viewing. However, active vision's impact is limited by slow options for mechanical camera movement. We propose a new design, called *FoveaCam*, and which works by capturing reflections off a tiny, fast moving mirror. FoveaCams can obtain high resolution imagery on multiple regions of interest, even if these are at different depths and viewing directions. We first discuss our prototype and optical calibration strategies. We then outline a control algorithm for the mirror to track target pairs. Finally, we demonstrate a practical application of the full system to enable eye tracking at a distance for frontal faces.

Index Terms—Novel cameras, eye-tracking, MEMS mirrors

1 Introduction

Most cameras today capture images without considering scene content. In contrast, human eyes have fast mechanical movements that control how the scene is imaged in detail by the fovea, where visual acuity is highest. This concentrates computational (i.e. neuronal) resources in places where they are most needed. Foveation and related ideas have been studied in robotics and active vision [1], [2], [3], [4], although these have been constrained by relatively slow pan-zoom-tilt (PZT) cameras and robot motion.

In this paper, we present a foveating camera design called *FoveaCam*, that distributes resolution onto regions of interest by imaging reflections off a scanning micro-electro mechanical system (MEMS) mirror. While MEMS mirrors are widely used in computational cameras for modulating illumination [5], [6], we use them to modulate *viewing direction*, much like catadioptric cameras [7].

MEMS mirrors are compact, have low-power performance and are fast. Speed, in particular, allows the capture of near-simulatenous imagery of dynamic scenes from different viewing directions. In fact, the mirror moves faster than the exposure rate of most video cameras, removing any visual cues that viewpoint has changed from frame to frame. Effectively, the images from single passive camera in Fig. 1 are interleaved from multiple virtual cameras, each corresponding to a different mirror position.

Leveraging the fast mirror speed to multiplex the viewpoint over multiple regions-of-interest (ROI) is only possible with a fast control strategy to decide which parts of the scene to capture at high resolution. We have adapted an efficient robot planning algorithm for MEMS mirror control, which can be optionally integrated with a target tracker. Instead of planning slow robot motion and varying PZT on the robots' onboard cameras, our new control algorithms enable quick, computationally light-weight, MEMS-mirror based changes of camera viewpoint for dynamic scenes.

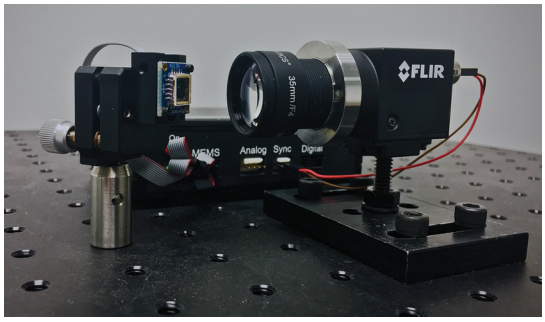


Fig. 1: Foveating camera setup

We illustrate our camera's utility through *remote* eye-tracking, showing that multiplexing resolution with FoveaCam results in higher fidelity eye-tracking. Remote eye tracking is an application that highlights the advantages of our novel camera design and control algorithms. Human eyes are a relatively small ROI, compared to faces and bodies, and eyes exhibit small and fast movements. Thus remote eye tracking tests those features of our camera that may, in the future, be used to enable other challenging dynamic imaging applications. In summary, our contributions are:

1) A novel sensor for dynamic scenes that temporally distributes its angular resolution over the FOV using reflections off a fast MEMS mirror. We discuss the system's optical characteristics and calibration.
2) An extension of the unicycle model for robot control to change the MEMS mirror path for pairs of targets. Our control algorithm is based on new closed form solutions for differential updates of the camera state.
3) A proof-of-concept gaze tracking application, created by fine tuning a recent eye-tracking neural network, demonstrating that our system enables better eye tracking at 3m range compared to a high-

• *B. Tilmon (btilmon@ufl.edu) and S. Koppal are with the Department of Electrical and Computer Engineering, E. Jain is with the Department of Computer and Information Science and Engineering, University of Florida. S. Ferrari is with the School of Mechanical and Aerospace Engineering, Cornell University.*

resolution commercial smartphone camera at the same location and with the same resolution.

1.1 Related work

Active vision and adaptive sampling: Ideas from visual attention [3], [4], have influenced robotics and vision, and information theoretic approaches are used to model adaptive 3D sensing for SLAM and other applications [8], [9], [10], [11]. Efficient estimation algorithms have been shown for adaptive visual and non-visual sensing on robots and point-zoom-tilt (PZT) cameras [12], [13], [14]. We propose to use active vision to drive the MEMS mirror directly in the camera, allowing for foveating over regions of interest.

MEMS/Galvo mirrors for vision and graphics: MEMS mirror modulation has been used for structured light [15], displays [16] and sensing [5]. We use MEMS mirrors to modulate viewing direction. MEMS mirrors used in LIDARs, such as from NASA and ARL [17], [18], [19], are run at resonance, while we control the MEMS scan pattern for novel imaging strategies. Such MEMS uses have been shown [20] for highly reflective fiducials in both fast 3D tracking and VR applications [21], [22]. We do not use special reflective fiducials and utilize active vision algorithms for MEMS mirror control. [23] shows a MEMS mirror-modulated 3D sensor with the potential for foveation, but without the adaptive algorithms that we discuss. In vision and graphics galvo mirrors are used with active illumination for light-transport [24], seeing around corners [25] and reconstruction with light curtains [6]. In contrast, the foveating camera presented here passively uses mirrors to image regions of interest in real-world scenes, compared to calibration-target oriented work [26], [27]. Our research is closest to [28] which was focused on static scenes, while we focus on dynamic scenes and control algorithms.

Selective imaging and adaptive optics: Our approach is similar in spirit to optical selective imaging with liquid crystal displays (LCDs) [29] and digital micro-mirror devices (DMDs) [5]. Because we use 2D scanning MEMS mirrors, we are able to allow the angular selectivity of [29] with the MEMS-enabled speed of [5]. Our design is the first to use a MEMS mirror to image dynamic scenes, although foveated designs have been proposed for static scenes, such as [30], [31]. Another related approach that uses fast optics for incident viewing is atmospheric sensing through turbulence with fast adaptive optics [32] with the difference being that we will show fast adaptive scene-specific imaging. Further, while we use a small MEMS mirror with many advantages of high-speed and low wear-and-tear, similar approaches have been tried with motor-driven mirrors [33].

Compressed sensing: Our approach of selectively imaging what is related to optically filtering light-fields for imaging tasks [34], [35], [36] and compressive sensing [37]. While there exist CS techniques for creating foveated imagery [31], [38], achieved sometimes during image capture, our goal is to distill visual information inside the camera, with MEMS mirror control, without requiring computationally intensive post-capture processing such as L1 optimization. Finally our approach involves fast modulation of the viewpoint, whereas fast temporal illumination modeling has enabled light-transport imaging [39], [40], [41], [42] and transient imaging [43], [44].

Remote gaze tracking: Previous efforts have built eye-trackers for use at either close distances or remotely using pan-zoom-tilt (PZT) cameras for applications such as home entertainment [45], [46], smart offices [11], outdoor advertising [47] and driver monitoring [48]. Depth and pose from stereo pairs has enabled gaze tracking from longer distances [49], [50]. We are the first to use a MEMS-mirror based foveating camera design for remote eye tracking. In our experiments, we track gaze from two people at 3m distance, separated by about a meter, which is currently not possible with any other technique. Further, our technique can easily accommodate multiple people with a single camera of high enough frame rate, since the MEMS mirror can move at KHz rates. In contrast, for methods that rely on PZT for dynamic scenes, frames are lost by the motorized sensors, unless each target is allocated a dedicated camera.

Large FOV cameras A natural argument against foveated imaging is to use a large field of view sensor. [51] demonstrated a camera for gigapixel imaging using a ball lens that overcomes lens resolution limits induced by aberrations. This camera uses a camera array on a PZT style motor. The proposed gigapixel camera fulfills a different role than we intend to fill with FoveaCam. The compactness and low bandwidth nature of FoveaCam lends itself towards mobile and resource constrained environments, where the camera array and PZT motor from [51] may prove burdensome.

Fast tracking with galvanometer mirrors Tracking with large galvo mirrors has been shown by [52]. This system tracks an object through a FOV via optical flow, and does not distribute resolution spatially to other objects. Galvo mirrors are very large and prone to over heating. Furthermore many galvo mirrors only rotate along one dimension, and two galvo mirrors are required for two dimensional tracking such as in [52]; a single MEMS mirror can rotate in two dimensions due to the gimbal-swivel design. Finally, [52] construct a very large high-bandwidth system whereas ours can run in embedded environments. Our advantages with FoveaCam include compactness, low bandwidth, 2D tracking, and a robust control algorithm for tracking multiple targets in a scene. Again, our advantage lies in resource-constrained applications.

2 Optical Setup and Calibration

We use a MEMS (micro-electro mechanical) swiveling mirror to direct the foveating camera viewpoint. The advantages of the MEMS mirror are speed and compactness. Figure 2 demonstrates that since the MEMS tilt angle ω changes the virtual camera viewpoint, we are able to generate multiple viewpoints at the inherent speed of the MEMS (typically in tens of KHz).

In our experiments, we assume the mirror fills the field-of-view (FOV) of the camera as in Fig. 2. We do this using a simple triangle-based scaling equation. We setup the equation by ray tracing reflection points to behind the mirror, yielding the virtual camera location. The system can then be solved using thin lens equations to determine the distance an object needs to be from the virtual camera to fill θ and have focus. From the figure, and from simple triangles, the camera FOV is $\theta = 2\ atan(\frac{s}{2f})$, where s is the sensor's longest dimension and f is the camera focal length.

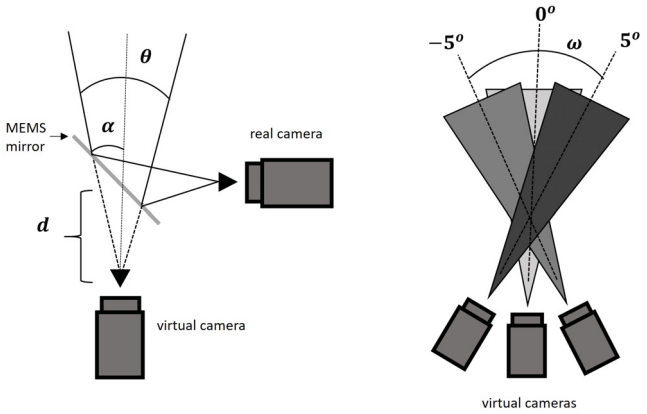


Fig. 2: Moving mirror creates virtual views

Assuming a mirror tilt α to the horizontal given by $\frac{\pi}{4}$, then full fill of the mirror requires the satisfaction of the following equations, where M is the largest mirror dimension and d is the mirror-to-camera distance along the optical axis,

$$d = \frac{M}{2}\ \sin(\alpha)\ \cot(\frac{\theta}{2}). \tag{1}$$

We pick focal lengths and camera resolutions to target imaging human heads at 5m-10m distances. In particular, for a $M = 3.0mm$ Mirrorcle mirror, we use a $f = 35mm$ lens and CMOS OV2710 1/2.7" $s = 5mm$ camera sensor, whose FOV is filled when the mirror-camera distance is 35mm. This enables a typical human head to fill θ when standing 2050mm from the virtual camera, allowing multiple people to be in the scene at 5m-10m distances while maintaining focus and high resolution on subjects. We chose α to be 45 degrees so an orthogonal relationship between the camera and virtual camera is upheld to ensure the virtual views do not see the real camera or system optics.

2.1 Navigating the design space

Using the equation above, we developed a simple calibration procedure for a user who provides sensor dimensions, camera lens properties, and MEMS mirror size, the model calculates the necessary MEMS mirror-camera distance to minimize vignetting, the optimal distance for a face to fill a desired field of view of the image, and the maximum field of view given the tilt of the MEMS mirror.

Our model predicts the distance a face needs to be from the virtual camera in order to fill either the horizontal or vertical fields of view of the camera, and the expected resolution of a face bounding box at this distance. We show experiments for validating these calibrations in Table 1 where the ground-truth resolution is determined by using a face classifier and counting pixels within the predicted face bounding box. Our model can be calibrated for any desired object size that fits within the FOV.

2.2 Resolution calibration

We used a 305mm x 305mm USAF 1951 Standard Layout chart from Applied Image Inc. to validate the resolution of our system across distances. To be visible in Near Infrared Wavelengths, we obtained a custom print of this standard

Distance (m)	Mirror in Camera (%)	No Mirror (%)
2	8.85	9.33
3	6.75	2.95
4	12.26	6.52
5	8.89	2.51

TABLE 1: Model Field of View Error (%)

pattern. We determined system resolution by inspecting the contrast between the last visible and the first non-visible group of lines. The frequency of the last group and the chart size provides the resolution.

To show the resolution robustness of our system, we compare experiments with the resolution chart for three cases: the mirror in our system, our foveating camera with no mirror in system, and an iPhone 6 Plus rear facing 12MP camera. Figure 3 shows our data at 4 meters for the three cases. Note our camera uses a 1/3" sensor, .003mm pixel size, and 35mm lens resulting in a 1080x1920 resolution while the iPhone 6s Plus uses a 1/3" sensor, .00122mm pixel size and a 4.15mm lens resulting in a 3024x2268 resolution.

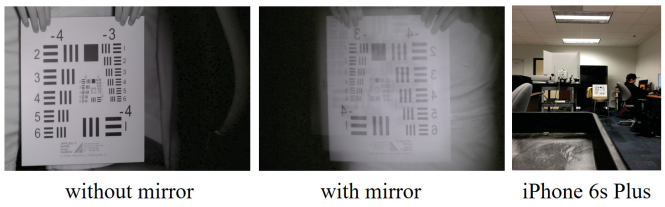
Our experiments show that imaging the mirror gives a resolution loss (lower frequency) compared to imaging without the mirror, and this is expected due to blur caused by the MEMS mirror cover glass and adding an element to the light path in general. Our system with or without the MEMS mirror still outperforms the iPhone 6 Plus. The average system resolution of the iPhone 6 Plus is .00097 cycles/mm, the average system resolution when imaging the mirror is 0.010 cycles/mm, and the average system resolution when imaging without the mirror is .018 cycles/mm. A higher cycles/mm means the system was able to detect higher frequencies (distinct lines) on the chart.

2.3 Cover glass calibration

The particular setup we use is tuned to near-infrared (NIR 850nm) data, which is commonly used for gaze tracking and iris detection due to its invariance to eye-color. For such a wavelength-sensitive application, further calibration is needed to deal with the cover glass that protects the MEMS mirror from dust and other particulates. Removing the coverglass would give an unobstructed light path, but would jeopardise the MEMS mirror safety. Unfortunately the cover-glass generates additional reflections or "ghosting". There are two primary reflections, we will call them the mirror and cover reflections. Here we discuss calibration preliminaries needed before our camera could be used for gaze tracking applications. While there are many techniques to computationally remove these artifacts [53], we wish to optically remove them to improve SNR and maintain our speed of capture. Fig. 4 shows our setup with the following additions that allows for ghosting removal:

Angle of Incidence: The Brewster angle of our cover glass (i.e., the angle at which polarized light completely transmits through a transparent surface) was determined to be 22.5 degrees from cover-glass manufacturer specifications, and by adjusting the mirror and camera angles to be near this value, we achieved partial cover reflection dampening.

Notch Filters: The cover glass transmits wavelengths between 675nm to 1040nm and reflects wavelengths outside



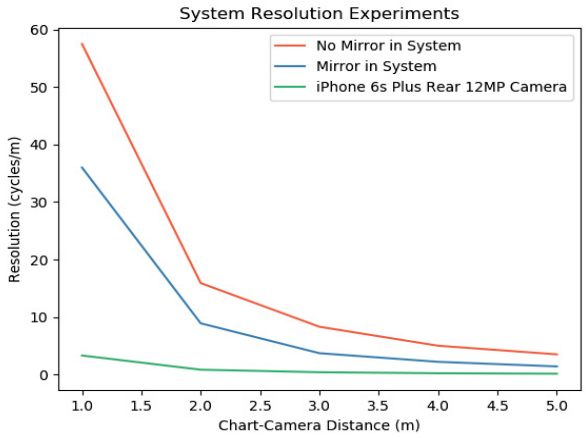


Fig. 3: Resolution experiments. At the top we show images of the resolution chart, with our camera, with and without mirror reflection, and a smartphone camera. The graph depicts the resolution change with distance, showing that the mirror/cover-glass reduces resolution, but that it is still better than a smartphone camera.

Fig. 4: The coverglass induces ghosting and double images. With a combination of using Brewster's angle, NIR notch filter and absorbing baffles, we eliminate the ghosting.

this range. Working within these specifications, we use a notch filter that transmits wavelengths between 800nm to 900nm and reflects wavelengths outside this range. Combined with a 850nm NIR light source to illuminate the scene, this reduces the reflection further.

Blocking cover reflection We insert an absorbing black paper in the center of the virtual camera field of view to remove incident light that would be reflected by the cover glass. We run the MEMS mirror outside this region.

3 Controlling the MEMS mirror motion

Given an optically calibrated foveating camera, as described by the previous section, we wish to move the MEMS mirror to best capture the scene. As in Fig. 1, our camera captures reflections off the MEMS mirror, whose azimuth and elevation are given by changes in control voltages over time, $(\theta(V(t)), \phi(V(t))$ over the mirror FOV ω_{mirror}.

3.1 Problem Setup

Let the system bandwidth be M pixels/second. Given an integer $k > 0$, we use a camera that captures $\frac{M}{k}$ pixel images at k images/second, in the foveating sensor. Since the mirror moves quickly, new active vision control is possible to distribute the k instances of the viewing cone within a second. The advantage of MEMS mirrors are its speed, allowing the mirror scan to quickly attend to a region-of-interest.

Consider a virtual plane Π perpendicular to the optical axis and parallel to the MEMS mirror in a resting, horizontal state, i.e. $(\theta = 0, \phi = 0)$. Every angular pose of the MEMS mirror (θ, ϕ) corresponds to a location (x, y) on Π given by perspective scaling. Consider a scene with two targets. For the purpose of this paper, we focus on targets that are the faces of two people. Long range eye tracking is possible if the mirror moves quickly between the two face locations.

To do this, consider a tight 1D sinusoid of amplitude $\frac{L_r}{2}$, bounded by the face locations. W.l.o.g consider one of these locations to be the "anchor" of the system, (x_r, y_r), while its orientation is given by the angle α_r w.r.t an arbitrary reference vector, such as one parallel to the lower edge of the MEMS mirror. We denote the *state* of the sensor by the triplet $q_r = (x_r, y_r, \alpha_r)$, and this state exists in a space of possible configurations given by the sensor hardware limits for 1D motion, $\mathbf{U} = (L_{min}, L_{max}) \times (\omega_{min}, \omega_{max})$. *The problem of control requires a solution that changes the state q_r of the sensor to enable target imaging.*

3.2 Control Algorithm Overview

To change the state to match the people's motion around the scene, we define a control vector $\mathbf{u_r} = (v_r, \omega_r)$ for a new desired motion, by specifying the velocity v_r by which the length of the 1D motion should change and the angular velocity ω_r by which the angle of the 1D motion should change. In the supplementary material, summarized briefly in Sect. 3.3, we use an optional Kalman filter to estimate the current state of the MEMS mirror's 1D motion and the face locations, given a previous state and face locations and the desired control vector. A reader who wishes to use a secondary strategy, such as face detection or thermal sensing, to track the probability distribution of targets over time can skip this section. Our contribution mainly lies in the subsequent sections Sect. 3.4 and Sect. 3.6, where we discuss how to come up with a control vector, given previously captured imagery from our sensor. Our model and control algorithm are adapted from the unicycle model of robot control [54].

3.3 Optional Kalman filter for state and target tracking

A probability distribution of the targets over time is necessary to control the viewing direction of the MEMS mirror in our camera. For experiments in Sect. 4 we have used a vision-based face-tracker as a proxy for this filter. For completeness we have provided the description of a Kalman filter tracker in the supplementary material. That supplementary material defines a control matrix $B_r(k)$ to update the state vector using the control vector $u_r(k)$:

$$q_r(k+1) = \mathbf{I_3} q_r(k) + B_r(k) u_r(k) + Q_r, \qquad (2)$$

where Q_r is the covariance matrices of the MEMS controller noise and $\mathbf{I_3}$ is the identity representing the state transition for a calibrated, controlled sensor (i.e. only our control vector and noise matters in changing the state).

Let the left and right face locations, on plane Π, be $q_f = [x_{lf}\ y_{lf}\ x_{rf}\ y_{rf}]$. Adding the face locations to the sensor state gives a full state vector, $q(k) = [q_r^T(k)\ q_f^T(k)]^T$. Since we have no control over the location of the faces, the full control vector $u(k) = [u_r(k)\ 0]^T$. The full prediction is

$$q(k+1) = F\ q(k) + B(k)\ u(k) + w, \tag{3}$$

where F is a target motion matrix, based on optical flow equations, derived in the supplementary material and w represents the process noise in the MEMS controller and the target motion and is denoted as covariance matrices Q_r and Q_t. Let the covariance matrix of the state vector (MEMS mirror + target faces) be $P_k = [P_r(k)\ 0; 0\ P_t(k)]$, where $P_r(k)$ is the covariance matrix representing the uncertainty in the MEMS mirror state and $P_t(k)$ is the covariance matrix representing the uncertainty in the target location. Then the change in uncertainty is

$$P(k+1) = [B_r(k)^T P_r B_r(k)\ 0; 0\ P_t] + [Q_r(k)\ 0; 0\ Q_t(k)], \tag{4}$$

where the untracked noise is represented in the MEMS controller and the target as covariances Q_r and Q_t.

The update step for the entire system is given by two types of sensor measurements. The first is the proprioceptive sensor based on the voltage measurements made directly with a USB oscilloscope that receives the same voltages sent to the MEMS. The second is a camera that views the reflections of the mirror and applies a standard face recognition classifier to each location, determining a probability distribution of left and right face locations across the FOV. From these two measurements we can propose both the estimated state vector and its covariance matrix, $[z(k), R(k)]$. Note that the measurement function (usually denoted as $H(k)$) is the identity in our setup since all the probability distributions share the same domain, i.e. the 2D plane Π created in front of the sensor. The remaining Kalman filter equations are

$$K^{'} = P(k+1)(P(k+1) + R(k+1))^{-1} \tag{5}$$

$$q^{'}(k+1) = q(k+1) + K^{'}(z(k+1) - q(k+1)) \tag{6}$$

$$P^{'}(k+1) = P(k+1) - K^{'}P(k+1) \tag{7}$$

3.4 A metric for good mirror control

We define a metric for control as the difference between the groundtruth (unknown) state $q(k)$ and the current state as predicted by the filter $q^{'}(k+1)$. However, if there is no face detection, then the filter cannot be applied and we default to the previous state moved by the control vector, given by $q(k+1)$. Let P_d be the probability that all faces were detected successfully.

$$M_k = P_d \mathbf{E}[e^{'}(k+1)^T\ e^{'}(k+1)] + (1-P_d)\mathbf{E}[e(k+1)^T\ e(k+1)]. \tag{8}$$

where

$$e^{'}(k+1) = q(k) - q^{'}(k+1). \tag{9}$$

$$e(k+1) = q(k) - q(k+1). \tag{10}$$

$$\tag{11}$$

Using the trace trick, similar to [54], we can convert M_k into an expression using the covariance matrices,

$$M_k = tr[P(k+1)] - P_d(tr[P(k+1)] - tr[P^{'}(k+1)]). \tag{12}$$

Since $tr[P(k+1)] - tr[P^{'}(k+1)]$ is always positive (due to uncertainty reduction of a Kalman filter), maximizing P_d reduces the error M_k. This is our *metric for good performance*, which should illuminate how to control the MEMS mirror with the control vector u_r.

3.5 Updating the control vector

The conclusion of the previous section's discussion can be depicted as a control law,

$$max_{\mathbf{u_r}} P_d \tag{13}$$

where P_d is defined as the probability that all the faces are detected, and is given by integrating the probability of seeing a face over the MEMS mirror path given by the state of the sensor, $q_r(k) = (x_r(k), y_r(k), \alpha_r(k))$. We now discuss a gradient-based iterative update to the control vector, given the sensor state and uncertainty.

.

Calculating P_d as a slice Given a parameter s, we can express the locations along which the probability P_d must be integrated as,

$$P_d(q_r(k)) = \int_{s=0}^{L} f_t(x_r(k) + s\cos\alpha_r(k), y_r(k) + s\sin\alpha_r(k))ds \tag{14}$$

where f_t is the probability distribution function of the faces in the canonical plane Π. The distribution f_t comes from the estimates of face location, which could be from the Kalman filter or from another process, and can be modeled as a pair of bi-variate Gaussian distributions, of equal weight (i.e. the mixing parameter is 0.5), such that $f_t(x,y) = f_l(x,y) + fr(x,y)$, where each Gaussian component centered at the two previously estimated left and right face locations given by $q_f(k-1) = [x_{lf}(k-1)\ y_{lf}(k-1)\ x_{rf}(k-1)\ y_{rf}(k-1)]$.

In other words, P_d is an integral along a slice through two bivariate Gaussian distributions. For each left and right case, we know the correlation matrix of both 2D gaussians, from the Kalman filter, given by $[\sigma_{1l}, \sigma_{2l}, \rho_l]$ for the left and $[\sigma_{1r}, \sigma_{2r}, \rho_r]$. Therefore the term $f_t(x_r(k) + s\cos\alpha_r(k), y_r(k) + s\sin\alpha_r(k))$ can be split into two components, where $x = x_r(k) + s\cos\alpha_r(k)$ and $y = y_r(k) + s\sin\alpha_r(k)$, the first given by $f_l(x,y)$

$$\frac{1}{2\pi\sigma_{1l}\sigma_{2l}\sqrt{1-\rho_l^2}} e^{-\frac{\frac{(x-x_{lf})^2}{\sigma_{1l}^2} - \frac{2\rho_l(x-x_{lf})(y-y_{lf})}{\sigma_{1l}\sigma_{2l}} + \frac{(y-y_{lf})^2}{\sigma_{2l}^2}}{2(1-\rho_l^2)}} \tag{15}$$

and the second given by $f_r(x,y)$

$$\frac{1}{2\pi\sigma_{1r}\sigma_{2r}\sqrt{1-\rho_r^2}} e^{-\frac{\frac{(x-x_{rf})^2}{\sigma_{1r}^2} - \frac{2\rho_l(x-x_{rf})(y-y_{rf})}{\sigma_{1r}\sigma_{2r}} + \frac{(y-y_{rf})^2}{\sigma_{2r}^2}}{2(1-\rho_r^2)}}. \tag{16}$$

Algorithm 1: Gradient-based update of control vector u_r

Input: Kalman filter outputs, valid space **U**, epsilon error threshold ϵ, learning rate η and initial control vector u_r

Output: Updated control vector u_r

while *1* **do**
  $u_r^{tmp} = u_r + \eta \frac{\delta P_d(q_r(k+1))}{\delta \mathbf{u_r}}$
  if $u_r^{tmp} \notin \mathbf{U}$
    return
  else if $\|u_r^{tmp} - u_r\| < \epsilon$
    return
  else
    $u_r = u_r^{tmp}$
  endif
end
return u_r

3.6 Arguments for using gradient descent

In this section we argue that maximizing the value P_d, can be tackled with gradient descent. First we show that P_d has at most two global maxima, by linking it to the well known Radon transform. Second we show that this formulation of P_d is bounded.

Global maxima: P_d is obtained by slicing through the two Gaussians at a line segment given by $q_r = (x_r, y_r, \alpha_r)$. By reconstituting this as a slice through a line with y intercept $y_{rad} = y_r + x_r * (tan(\alpha_r))$ and slope $s_{rad} = tan(\alpha_r)$, we notice that P_d is the Radon transform of a bi-variate distribution. For each Gaussian distribution individually, this transform has been shown to be unimodal with a global maxima and continuous [55] for a zero-mean Gaussian. Since translations and affine transformations do not affect the radon transform, these hold for any Gaussian distribution. For the sum of radon transforms of two such Gaussians, there can be at most two global maxima (if these are equal) and at least one maxima (if these overlap perfectly). Further, since the sum of two continuous functions is also continuous, the radon transform of the bi-variate distribution is also continuous. Finally, the Radon transform is computationally burdensome for a robot to compute at every frame, which supports using iterative gradient descent.

Bounded domain: Consider any slice through the bi-variate distribution. Consider a slice that has the centers of the two Gaussians on the *same* side of the slice. Then, by moving the slice towards the two centers, we can increase both components of P_d exponentially and monotonically. So such a slice cannot maximize P_d. From the above argument, the slice that maximizes P_d goes through a line segment between the centers of the two Gaussians. Note, we are saying that the optimal slice must intersect this line segment somewhere. In other words, the domain, within the Radan transform of bi-variate Gaussians, where we must search for the maximal slice, is bounded.

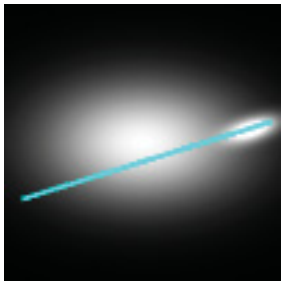


Fig. 5: Heterogeneity

Optimal path is not the line joining Gaussians' center: While the line joining the Gaussians' center is a useful heuristic, it is not a general solution since the length of the integral L could be smaller than the distance between the Gaussian centers. Secondly, the heuristic tends to work when the Gaussians are similar; if one Gaussian dominates, as in Fig. 5, then the optimal line can be different.

From these arguments of bounded domain and continuity, the application of gradient descent is a reasonable strategy for lightweight optimization of the control law.

3.7 Gradient descent

Gradients and algorithm We compute the Jacobian (i.e. derivatives) of $P_d(q_r(k+1))$, given by $\mathbf{u_r}$

$$\frac{\delta P_d(q_r(k+1))}{\delta \mathbf{u_r}} = \frac{\delta P_d(q_r(k+1))}{\delta q_r(k+1)} \frac{\delta q_r(k+1)}{\delta \mathbf{u_r}} \tag{17}$$

Since the second term is the sensor motion model $B_r(k)\delta t$, we just need to calculate the first term,

$$\frac{\delta P_d(q_r(k+1))}{\delta q_r(k+1)} = \begin{bmatrix} \frac{\delta}{\delta x_r} P_d(q_r(k+1)) \\ \frac{\delta}{\delta y_r} P_d(q_r(k+1)) \\ \frac{\delta}{\delta \alpha_r} P_d(q_r(k+1)) \end{bmatrix} \tag{18}$$

We can rewrite this by setting $x = x_r(k) + s\cos\alpha_r(k)$ and $y = y_r(k) + s\sin\alpha_r(k)$, and by splitting f_t into left and right Gaussians, as

$$\frac{\delta P_d(q_r(k+1))}{\delta q_r(k+1)} = \begin{bmatrix} \frac{\delta}{\delta x_r} \int_{s=0}^{L} f_l(x,y)ds \\ \frac{\delta}{\delta y_r} \int_{s=0}^{L} f_l(x,y)ds \\ \frac{\delta}{\delta \alpha_r} \int_{s=0}^{L} f_l(x,y)ds \end{bmatrix} + \begin{bmatrix} \frac{\delta}{\delta x_r} \int_{s=0}^{L} f_r(x,y)ds \\ \frac{\delta}{\delta y_r} \int_{s=0}^{L} f_r(x,y)ds \\ \frac{\delta}{\delta \alpha_r} \int_{s=0}^{L} f_r(x,y)ds \end{bmatrix} \tag{19}$$

These gradients can easily be calculated after every iteration of the Kalman filter, allowing for the closed form update of the MEMS mirror based on the movement of the faces, sensor state and uncertainty. In our experiments, we computed closed forms of these using a commercially available symbolic calculator, and the accompanying files representing the derivatives are provided in the supplementary material. In Algorithm 1 we use these gradients to update the control vector.

Simulations: In Fig 6 we show simulations of Algorithm 1 on 20 pairs of 2D Gaussians. In Fig 6I(a) we select four from these 20, showing the ground-truth "slice" that maximizes target probability, P_d , calculated from the radon transform. In Fig. 6I(b) we show the results of the experiments. For each Gaussian pair, we began the gradient descent at an initialization from the ground-truth, using a shift of mean zero and standard deviation σ such that $3 * \sigma$ varies from 0 to about a 25% of the image width. This means that at the extreme case, initialization could be anywhere in a 50% chunk of the image near the ground-truth. Fig. 6II shows similar experiments where we only allowed initializations in the constrainted domain of the segment between the maxima of the Gaussians. This reduces the overall error percentage slightly in Fig. 6II(b).

Fig. 6I-II(b) graphs show Euclidean distance between the converged slice and ground truth, averaged over five trials.

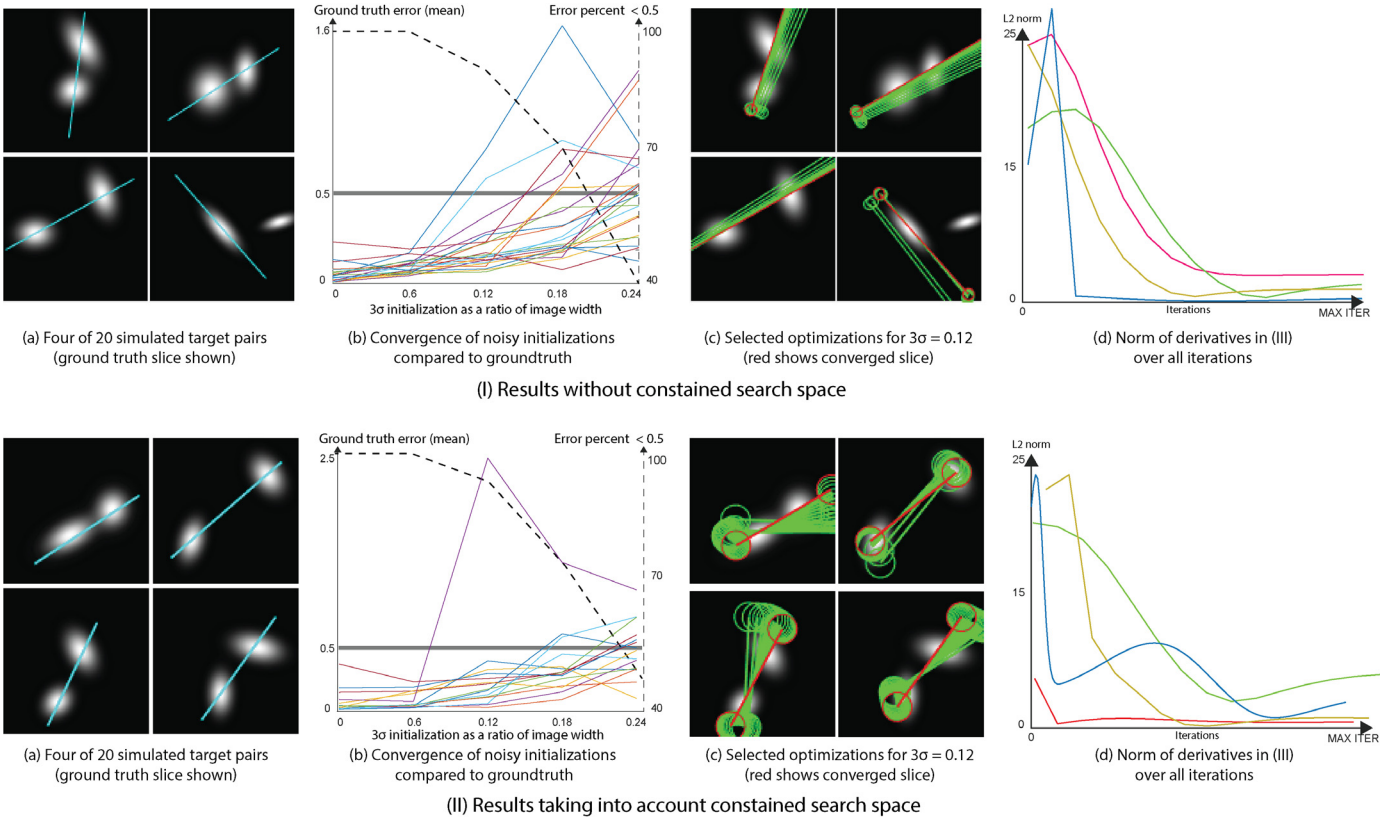


Fig. 6: Simulations of 1D slice optimization: In (I) and (II) we created simulations to test the iterative optimization in Algorithm 1. (I) is a free-form optimization, whereas (II) constrains the optimization along the proposed bounded region. Note that the percent error for (II) is slightly lower.

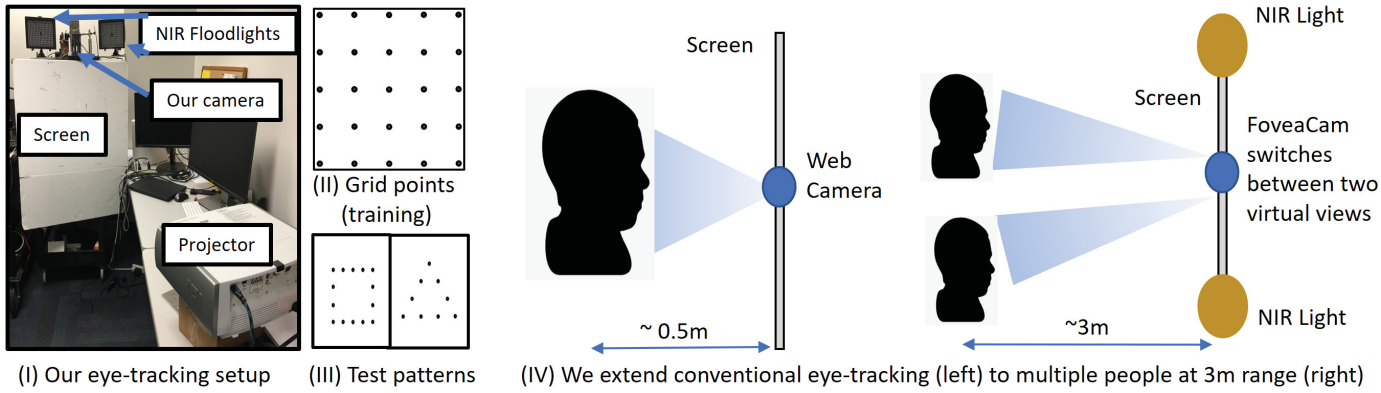


Fig. 7: Our eye tracking setup and train/test patterns.

Note that most results converge even for large deviations from the ground-truth. In Fig. 6I-II(c) we the convergence path for these examples, and in Fig. 6I-II(d) we show that the L2 norm of the gradients decreases as it converges.

Practical considerations: While we have provided gradients for optimization, other factors influence convergence. For example, learning rate in our experiments was fixed at a user-defined value, and this could be annealed during optimization. Failure cases of our setup are due to initializations that are too distant from either Gaussian and, therefore, have small gradients (i.e. local minima). Again, more capable optimization strategies, using our gradients, can result in better convergence.

4 Remote Eye-Tracking for Frontal Faces

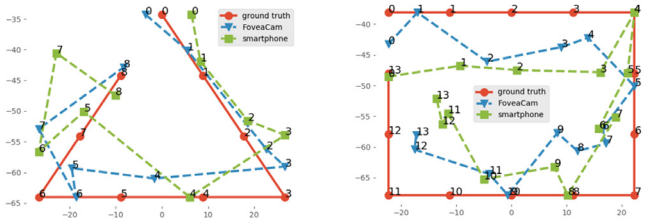
Remote eye-tracking for frontal faces has potential applications in situations where the faces are directly viewed by the camera, such as human-robot interaction, automobile safety, smart homes and in educational, classroom settings.

In this section, we describe our testbed for remote eye-tracking, where we compare our sensor (foveating camera) with near-co-located smartphone, viewing frontal faces at around 3m distance. We present a proof-of-concept remote eye-tracking system that uses our MEMS mirror enabled foveating camera to capture images, and the iTracker convolutional neural network [56] to analyze these images.

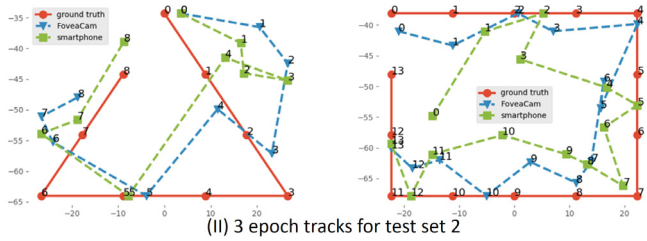
(I) Fine-tuning data at 1920x1080. Left is our camera and right is smartphone

(II) Test data at 1920x1080. Left is our camera and right is smartphone

Fig. 8: Our training/test data for fine-tuning.



(I) 3 epoch tracks for test set 1



(II) 3 epoch tracks for test set 2

Fig. 9: 3 epochs tracks for both test sets in Table 3

Camera	RMSE (3 epochs) (cm)
Smartphone (random initial.)	55.91
Smartphone (iTracker initial.)	6.88
Foveating camera (random initial.)	45.77
Foveating camera (iTracker initial.)	7.73

TABLE 2: Random initialization fails (Train/Val error)

4.1 Our eye-tracking setup

Our setup, shown in Fig. 7, consists of our foveating camera, placed between two NIR floodlights. The setup is at the top of a textureless lambertian plane of width approximately $100cm \times 100cm$. A video projector, placed at $2m$ distance, projects either a grid training pattern, or two test patterns, a triangle and a rectangle, as in the figure.

Two subjects at $3m$ distance from the camera, view the patterns, focusing on each dot for about 5 seconds. The smartphone camera has a FOV of $55°$ and views both subjects. Our camera has a FOV of $8.6°$ and alternates between the two subjects. In Sect. 5, we describe how to control the movement of the mirror due to subject motion, but in this section we will assume that only the eyes of the subjects move. *Therefore in all our experiments, for the same pixel bandwidth of* 1920×1080 *for our sensor and the smartphone, we are able to increase the angular resolution by a factor of* $\frac{55}{8.6} \approx 6$ *times.* This is the main advantage of the foveating camera. Now we discuss the impact of this increased resolution on eye-tracking performance.

4.2 Fine-tuning a gaze-tracking network

The iTracker convolutional neural network [56] takes in four inputs derived from a single capture of a face (both eyes, cropped face and face location), assumed to be captured on a smartphone, at arms length from the face. Each of these inputs goes into a dedicated Alexnet-inspired network, with the eye-layers sharing weights. The outputs of the layers are a 2D gaze location, relative the camera; e.g., the output is $(0, 0)$ for someone looking directly at the camera.

86% of the iTracker imagery is iPhone data trained on eye angles varying in y from 2cm(4.5 degrees) to 10cm (21.8 degrees) and x from -1cm(2.3 degrees) to 5cm(13.5 degrees). To maintain these angles at 3m for our data, we trained on patterns spanning x from -39cm to 39cm (7.4 degrees) and y from -21cm (4 degrees) to -82cm (15.3 degrees).

While this network has been trained on the GazeCapture dataset of around 1400 subjects in a variety of domains, it cannot be used directly on our setup (described next), since the geometry of the setup is different (i.e. subjects are much further away, $25cm$ in iTracker vs. $3m$ for us) which the perspective of how much the eyes appear to move for the same angle. Further, our data is in the NIR range, which is different domain than the data used in the paper. In all our results, we compare the original results with fine tuning with domain-specific data collected with our setup. All our training and testing was done at 3m from the camera.

4.3 Data collection for fine-tuning

We implemented the iTracker [56] in PyTorch with its pre-trained weights. The network performs poorly using the provided network weights at the same span of test points at 3m as the iPhone tests. This is expected since viewing a 12cm spanned x,y pattern (iPhone) at 3m gives less than 1 degree eye angle. Commercial eye trackers typically employ 1 degree eye angle tolerance or higher. To circumvent lack of eye angle, we fine tuned the network on data with the correct in-situ angular properties.

Experiments with four volunteers (3 male and 1 female, see Fig. 9(I)) enabled the collection of fine-tuning data in-situ with the device, in NIR, for the grid pattern in Fig. 7. Each data collection experiment lasted 20 minutes, and data was collected simultaneously for smartphone and foveating camera. We record 400 images per point, giving 10,000 images per subject or 40,000 total images. We use 33,000 images due to faulty face and eye detections being discarded to maintain high-fidelity data. We split our training set into 26,400 train and 6,600 validation, randomly shuffled.

For fine-tuning, we begin with identical weights to [56], except we lower our learning rate ten fold. We do not freeze any layers. The only parameter left in fine-tuning is the number of epochs, and too many epochs will result in

TEST SET 1 ERRORS

Rectangle			Triangle		
	Our RMSE (cm)	Smartphone RMSE (cm)		Our RMSE (cm)	Smartphone RMSE (cm)
3 epochs	**6.67**	6.94	3 epochs	**5.24**	7.15
10 epochs	7.05	**6.12**	10 epochs	**6.29**	6.34

TEST SET 2 ERRORS

Rectangle			Triangle		
	Our RMSE (cm)	Smartphone RMSE (cm)		Our RMSE (cm)	Smartphone RMSE (cm)
3 epochs	**5.14**	7.87	3 epochs	**7.44**	8.9
10 epochs	**6.26**	7.75	10 epochs	**6.86**	8.23

TABLE 3: Results on both test experiments

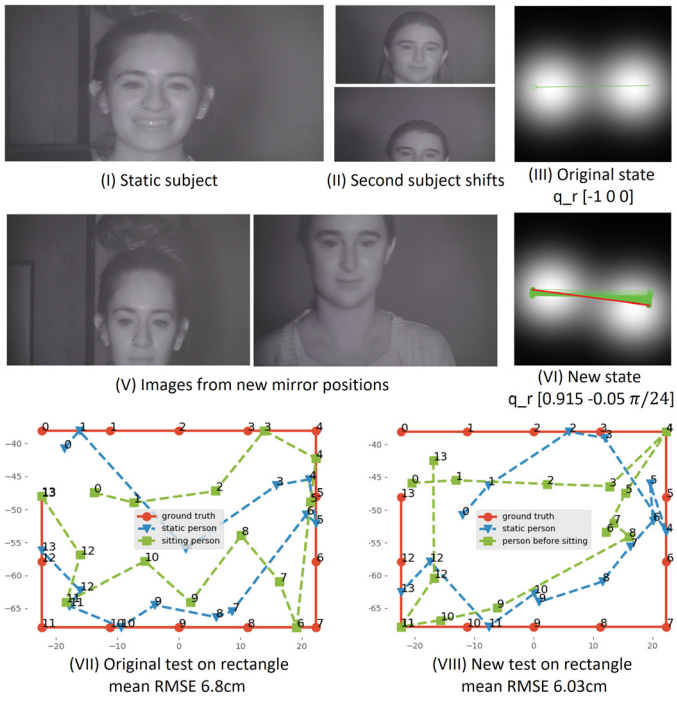


Fig. 10: Proof-of-concept control experiments where one subject moves slightly, and Algorithm 1 is used to reset the mirror positions. Eye tracks are provided for start state and the end state, after mirror motion.

over-fitting and loss of knowledge from the original 1200 iTracker experiments. In our experiments we have run the fine-tuning on two sets, 3 and 10 epochs.

4.4 Experimental results

In our experiments, the subjects were at $3m$ distance and eight people were involved overall, four in training and four in testing. To show that this relatively small fine-tuning dataset does not adversely affect our results, we show, in Table 2, that validation errors after 3 epochs for both our camera and the smartphone are much higher when starting from random weights, than from the pre-trained weights. So, our small dataset is simply used for fine-tuning and does not overfit after 3 epochs, and we do indeed utilize the 1400 users encapsulated in the pre-trained weights.

Our test dataset consists of two pairs of two subjects (see Fig. 9(II)) looking at the square and triangle patterns at both max mirror tilt(+7 degrees) and minimum mirror tilt (-7 degrees) giving a total of 1380 images for each pair. In Table 3 we show the results of both our camera and the smartphone camera for the cases of 3 epochs and 10 epochs. Table 3 also shows the test eye-tracks for the rectangle and triangle for these epochs. We note that the higher resolution we place on the face enables our images to reach a lower RMSE than the smartphone images, after 3 epochs. In fact, it takes the smartphone images more than three times the training time, i.e. 10 epochs, to reach a comparable accuracy to ours. Clearly, our method is able to capture more SNR due to its concentrated resolution on the region of interest.

5 Proof-of-Concept Control Experiment

Finally, we use the control from Section 3, along with the eye-tracking capability described in the previous section, to demonstrate a proof-of-concept capability of our sensor. In this experiment, one of the pair of persons from our test subjects are looking at the square test pattern.

We use a the bounding box from a simple face-tracker [57] as a proxy for the Kalman filter, and use the a user defined ratio k to map the maximum box dimension d_{max} to the variance $\sigma = k * d_{max}$ in a symmetric Gaussian centered on the box that approximates the probability distribution of the face. Combining this for both faces provides the probability distribution of the targets P_d, required in our control law. In all our experiments this ratio was $k \approx 3$. In Fig. 10, we show the initial state of the scene for the two test subjects and the corresponding gaze track for the triangle test data at the initial mirror position of [-1 0] for the left person and [1 0] for the person on the right and the control state is $q_r = [-1\ 0\ 0]$. Then, one person moves, as shown in the figure. We run Algorithm 1 from the initial starting point, as shown in the Fig. 10, which converges to mirror positions of [-.86 0.331] and [.915 -0.05] respectively with a state vector of $q_r = [.915\ \ -0.05\ \frac{\pi}{24}]$. Note that, at these new positions, both faces are clearly visible, and the gaze tracking experiment for the square test data, redone at this new mirror position, also produces good quality results ($6.8cm$ and $6.03cm$ RSME respectively).

6 Conclusion

We have demonstrated, for the first time, a foveating camera that captures dynamic scenes through viewing reflections off a MEMS mirror. FoveaCam is the first step towards fast foveating imaging that could be used, in the future, to enable other challenging dynamic imaging applications

such as tracking small robotic insects, smart education in very large classrooms and lip-reading from far-off distances.

ACKNOWLEDGMENTS

Eakta Jain has been supported by the National Science Foundation through NSF: 1566481. Sanjeev Koppal and Brevin Tilmon have been partially supported by the National Science Foundation through NSF IIS: 1909729 and the Office of Naval Research through ONR N00014-18-1-2663. Silvia Ferrari has been supported by N00014-19-1-2144.

REFERENCES

[1] F. Pittaluga, Z. Tasneem, J. Folden, B. Tilmon, A. Chakrabarti, and S. J. Koppal, "A mems-based foveating lidar to enable real-time adaptive depth sensing," *arXiv:2003.09545*, 2020.

[2] J. Aloimonos, I. Weiss, and A. Bandyopadhyay, "Active vision," *International journal of computer vision*, vol. 1, no. 4, pp. 333–356, 1988.

[3] S. Frintrop, E. Rome, and H. I. Christensen, "Computational visual attention systems and their cognitive foundations: A survey," *ACM Transactions on Applied Perception (TAP)*, vol. 7, no. 1, p. 6, 2010.

[4] N. Bruce and J. Tsotsos, "Attention based on information maximization," *Journal of Vision*, vol. 7, no. 9, pp. 950–950, 2007.

[5] S. K. Nayar, V. Branzoi, and T. E. Boult, "Programmable imaging: Towards a flexible camera," *International Journal of Computer Vision*, vol. 70, no. 1, pp. 7–22, 2006.

[6] J. Wang, J. Bartels, W. Whittaker, A. C. Sankaranarayanan, and S. G. Narasimhan, "Programmable triangulation light curtains," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 19–34.

[7] J. Gluckman and S. K. Nayar, "Planar catadioptric stereo: Geometry and calibration," in *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, vol. 1. IEEE, 1999, pp. 22–28.

[8] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.

[9] B. Charrow, G. Kahn, S. Patil, S. Liu, K. Goldberg, P. Abbeel, N. Michael, and V. Kumar, "Information-theoretic planning with trajectory optimization for dense 3d mapping." in *Robotics: Science and Systems*, vol. 11. Rome, 2015.

[10] E. Ristani and C. Tomasi, "Features for multi-target multi-camera tracking and re-identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6036–6046.

[11] T. Darrell, B. Moghaddam, and A. P. Pentland, "Active face tracking and pose estimation in an interactive room," in *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 1996, pp. 67–72.

[12] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[13] H. Wei, P. Zhu, M. Liu, J. P. How, and S. Ferrari, "Automatic pan–tilt camera control for learning dirichlet process gaussian process (dpgp) mixture models of multiple moving targets," *IEEE Transactions on Automatic Control*, vol. 64, no. 1, pp. 159–173, 2018.

[14] C. Ding, B. Song, A. Morye, J. A. Farrell, and A. K. Roy-Chowdhury, "Collaborative sensing in a distributed ptz camera network," *IEEE Transactions on Image Processing*, vol. 21, no. 7, pp. 3282–3295, 2012.

[15] R. Raskar, G. Welch, M. Cutts, A. Lake, L. Stesin, and H. Fuchs, "The office of the future: A unified approach to image-based modeling and spatially immersive displays," in *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*. ACM, 1998, pp. 179–188.

[16] A. Jones, I. McDowall, H. Yamada, M. Bolas, and P. Debevec, "Rendering for an interactive 360 light field display," *ACM Transactions on Graphics (TOG)*, vol. 26, no. 3, p. 40, 2007.

[17] T. P. Flatley, "Spacecube: A family of reconfigurable hybrid on-board science data processors," 2015.

[18] B. L. Stann, J. F. Dammann, M. Del Giorno, C. DiBerardino, M. M. Giza, M. A. Powers, and N. Uzunovic, "Integration and demonstration of mems–scanned ladar for robotic navigation," in *Proc. SPIE*, vol. 9084, 2014, p. 90840J.

[19] K. T. Krastev, H. W. Van Lierop, H. M. Soemers, R. H. M. Sanders, and A. J. M. Nellissen, "Mems scanning micromirror," Sep. 3 2013, uS Patent 8,526,089.

[20] A. Kasturi, V. Milanovic, B. H. Atwood, and J. Yang, "Uav-borne lidar with mems mirror-based scanning capability," in *Proc. SPIE*, vol. 9832, 2016, p. 98320M.

[21] V. Milanović, A. Kasturi, J. Yang, and F. Hu, "A fast single-pixel laser imager for vr/ar headset tracking," in *Proc. of SPIE Vol*, vol. 10116, 2017, pp. 101 160E–1.

[22] V. Milanović, A. Kasturi, N. Siu, M. Radojičić, and Y. Su, "memseye for optical 3d tracking and imaging applications," in *Solid-State Sensors, Actuators and Microsystems Conference (TRANSDUCERS), 2011 16th International*. IEEE, 2011, pp. 1895–1898.

[23] T. Sandner, C. Baulig, T. Grasshoff, M. Wildenhain, M. Schwarzenberg, H.-G. Dahlmann, and S. Schwarzer, "Hybrid assembled micro scanner array with large aperture and their system integration for a 3d tof laser camera," in *MOEMS and Miniaturized Systems XIV*, vol. 9375. International Society for Optics and Photonics, 2015, p. 937505.

[24] T. Hawkins, P. Einarsson, and P. E. Debevec, "A dual light stage." *Rendering Techniques*, vol. 5, pp. 91–98, 2005.

[25] M. O ´ Toole, D. B. Lindell, and G. Wetzstein, "Confocal non-line-of-sight imaging based on the light-cone transform," *Nature*, vol. 555, no. 7696, p. 338, 2018.

[26] T. Tang, Y. Huang, C. Fu, and S. Liu, "Acceleration feedback of a ccd-based tracking loop for fast steering mirror," *Optical Engineering*, vol. 48, no. 1, p. 013001, 2009.

[27] N. Chen, B. Potsaid, J. T. Wen, S. Barry, and A. Cable, "Modeling and control of a fast steering mirror in imaging applications," in *2010 IEEE International Conference on Automation Science and Engineering*. IEEE, 2010, pp. 27–32.

[28] H. Hua and S. Liu, "Dual-sensor foveated imaging system," *Applied optics*, vol. 47, no. 3, pp. 317–327, 2008.

[29] A. Zomet and S. K. Nayar, "Lensless imaging with a controllable aperture," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 1. IEEE, 2006, pp. 339–346.

[30] G. Sandini and G. Metta, "Retina-like sensors: motivations, technology and applications," in *Sensors and sensing in biology and engineering*. Springer, 2003, pp. 251–262.

[31] S. Liu, C. Pansing, and H. Hua, "Design of a foveated imaging system using a two-axis mems mirror," in *International Optical Design Conference 2006*, vol. 6342. International Society for Optics and Photonics, 2006, p. 63422W.

[32] J. M. Beckers, "Adaptive optics for astronomy: principles, performance, and applications," *Annual review of astronomy and astrophysics*, vol. 31, no. 1, pp. 13–62, 1993.

[33] T. Nakao and A. Kashitani, "Panoramic camera using a mirror rotation mechanism and a fast image mosaicing," in *Proceedings 2001 International Conference on Image Processing (Cat. No. 01CH37205)*, vol. 2. IEEE, 2001, pp. 1045–1048.

[34] R. Raskar, A. Agrawal, and J. Tumblin, "Coded exposure photography: motion deblurring using fluttered shutter," in *ACM transactions on graphics (TOG)*, vol. 25, no. 3. ACM, 2006, pp. 795–804.

[35] R. Ng, "Fourier slice photography," in *ACM transactions on graphics (TOG)*, vol. 24, no. 3. ACM, 2005, pp. 735–744.

[36] A. Levin, R. Fergus, F. Durand, and W. T. Freeman, "Image and depth from a conventional camera with a coded aperture," *ACM transactions on graphics (TOG)*, vol. 26, no. 3, p. 70, 2007.

[37] M. B. Wakin, J. N. Laska, M. F. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. F. Kelly, and R. G. Baraniuk, "An architecture for compressive imaging," in *Image Processing, 2006 IEEE International Conference on*. IEEE, 2006, pp. 1273–1276.

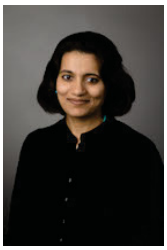
[38] I. B. Ciocoiu, "Foveated compressed sensing," *Circuits, Systems, and Signal Processing*, vol. 34, no. 3, pp. 1001–1015, 2015.

[39] M. Gupta, S. K. Nayar, M. B. Hullin, and J. Martin, "Phasor imaging: A generalization of correlation-based time-of-flight imaging," *ACM Transactions on Graphics (ToG)*, vol. 34, no. 5, p. 156, 2015.

[40] M. O ´ Toole, F. Heide, L. Xiao, M. B. Hullin, W. Heidrich, and K. N. Kutulakos, "Temporal frequency probing for 5d transient analysis of global light transport," *ACM Transactions on Graphics (ToG)*, vol. 33, no. 4, p. 87, 2014.

[41] M. O ´ Toole, S. Achar, S. G. Narasimhan, and K. N. Kutulakos, "Homogeneous codes for energy-efficient illumination and imaging," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, p. 35, 2015.
[42] S. Achar, J. R. Bartels, W. L. Whittaker, K. N. Kutulakos, and S. G. Narasimhan, "Epipolar time-of-flight imaging," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, p. 37, 2017.
[43] A. Velten, D. Wu, B. Masia, A. Jarabo, C. Barsi, C. Joshi, E. Lawson, M. Bawendi, D. Gutierrez, and R. Raskar, "Imaging the propagation of light through scenes at picosecond resolution," *Communications of the ACM*, vol. 59, no. 9, pp. 79–86, 2016.
[44] F. Heide, M. B. Hullin, J. Gregson, and W. Heidrich, "Low-budget transient imaging using photonic mixer devices," *ACM Transactions on Graphics (ToG)*, vol. 32, no. 4, p. 45, 2013.
[45] D.-C. Cho, W.-S. Yap, H. Lee, I. Lee, and W.-Y. Kim, "Long range eye gaze tracking system for a large screen," *IEEE Transactions on Consumer Electronics*, vol. 58, no. 4, pp. 1119–1128, 2012.
[46] C. Hennessey and J. Fiset, "Long range eye tracking: bringing eye tracking into the living room," in *Proceedings of the Symposium on Eye Tracking Research and Applications*. ACM, 2012, pp. 249–252.
[47] M. Khamis, A. Hoesl, A. Klimczak, M. Reiss, F. Alt, and A. Bulling, "Eyescout: Active eye tracking for position and movement independent gaze interaction with large public displays," in *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. ACM, 2017, pp. 155–166.
[48] O. Palinko, A. L. Kun, A. Shyrokov, and P. Heeman, "Estimating cognitive load using remote eye tracking in a driving simulator," in *Proceedings of the 2010 symposium on eye-tracking research & applications*. ACM, 2010, pp. 141–144.
[49] D. Beymer and M. Flickner, "Eye gaze tracking using an active stereo head," in *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, vol. 2. IEEE, 2003, pp. II–451.
[50] D. Geisler, D. Fox, and E. Kasneci, "Real-time 3d glint detection in remote eye tracking based on bayesian inference," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 7119–7126.
[51] O. Cossairt, D. Miau, and S. Nayar, "Gigapixel Computational Imaging," in *IEEE International Conference on Computational Photography (ICCP)*, Mar 2011.
[52] H. O. Kohei Okumura and M. Ishikawa, "High-speed gaze controller for millisecond-order pan/tilt camera," *ICRA*, 2011.
[53] Y. Shih, D. Krishnan, F. Durand, and W. T. Freeman, "Reflection removal using ghosting cues," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3193–3201.
[54] H. Wei, W. Lu, P. Zhu, G. Huang, J. Leonard, and S. Ferrari, "Optimized visibility motion planning for target tracking and localization," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 76–82.
[55] E. W. Weisstein, "Radon transform–gaussian. From MathWorld—A Wolfram Web Resource," last visited on 10/8/2019. [Online]. Available: \url{http://mathworld.wolfram.com/RadonTransformGaussian.html}
[56] K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik, and A. Torralba, "Eye tracking for everyone," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
[57] M. Jones and P. Viola, "Fast multi-view face detection," *Mitsubishi Electric Research Lab TR-20003-96*, vol. 3, no. 14, p. 2, 2003.

Brevin Tilmon received the B.S. degree in engineering physics from Murray State University and is currently a PhD student in electrical engineering at the University of Florida. He is interested in developing adaptive imaging systems and computer vision/computational photography algorithms.



Eakta Jain is an Assistant Professor of Computer and Information Science and Engineering at the University of Florida. She received her PhD and MS degrees in Robotics from Carnegie Mellon University, working in the Graphics Lab. Her B.Tech. degree is in Electrical Engineering from IIT Kanpur. She has worked in industrial research at Texas Instruments Research and Development labs, Disney Research Pittsburgh, and the Walt Disney Animation Studios. Eaktas research interests are in building human-centered computer graphics algorithms to create and manipulate artistic content, including traditional hand animation, comic art, and films. Her work has been presented at venues such as ACM SIGGRAPH, and has won multiple awards.

Silvia Ferrari is John Brancaccio Professor of Mechanical and Aerospace Engineering at Cornell University. Prior to that, she was Professor of Engineering and Computer Science at Duke University, and Founder and Director of the NSF Integrative Graduate Education and Research Traineeship (IGERT) and Fellowship program on Wireless Intelligent Sensor Networks (WISeNet). Currently, she is the Director of the Laboratory for Intelligent Systems and Controls (LISC) at Cornell University, and her principal research interests include robust adaptive control of aircraft, learning and approximate dynamic programming, and optimal control of mobile sensor networks. She received the B.S. degree from EmbryRiddle Aeronautical University and the M.A. and Ph.D. degrees from Princeton University. She is a senior member of the IEEE, and a member of ASME, SPIE, and AIAA. She is the recipient of the ONR young investigator award (2004), the NSF CAREER award (2005), and the Presidential Early Career Award for Scientists and Engineers (PECASE) award (2006).

Sanjeev Koppal is an assistant professor at the University of Florida ECE department. Prior to joining UF, he was a researcher at the Texas Instruments Imaging Research and Development lab. Sanjeev obtained his Masters and Ph.D. degrees from the Robotics Institute at Carnegie Mellon University. After CMU, he was a post-doctoral research associate in the School of Engineering and Applied Sciences at Harvard University. He received his B.S. degree from the University of Southern California in 2003. His interests span computer vision, computational photography and optics, novel cameras and sensors, 3D reconstruction, physics-based vision and active illumination.