Fast Foveating Cameras for Dense Adaptive Resolution

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

Abstract—Traditional cameras field of view (FOV) and resolution predetermine computer vision algorithm performance. These trade-offs decide the range and performance in computer vision algorithms. We present a novel foveating camera whose viewpoint is dynamically modulated by a programmable micro-electromechanical (MEMS) mirror, resulting in a natively high-angular resolution wide-FOV camera capable of densely and simultaneously imaging multiple regions of interest in a scene. We present calibrations, novel MEMS control algorithms, a real-time prototype, and comparisons in remote eye-tracking performance against a traditional smartphone, where high-angular resolution and wide-FOV are necessary, but traditionally unavailable.

Index Terms—Computational photography, computer vision

1 Introduction

Contemporary cameras indiscriminately sample their environment. Human eyes, however, adaptively distribute resolution to their direct line-of-sight at the fovea, which contains the highest visual acuity. Resolution adaptivity allows for efficiently expending resources only where necessary: our direct line-of-sight. In this paper, we bring resolution adaptivity to cameras by way of a foveating camera, capable of quickly, densely and simultaneously imaging multiple regions of interest in a scene.

Our foveating camera adaptively distributes resolution by viewing reflections off a programmable micro-electromechanical system (MEMS) mirror capable of fast 2D modulation, see Fig. 1. Recent advances in MEMS and galvonometer mirror sensors have enabled significant progress in fast, scene-aware, adaptive depth sensing [1], [2], [3], [4] and illumination [5], [6], [7]. Inserting mirrors into sensors optical path extends the sensors FOV to desired directions and densities that are usually only attainable through physically moving the sensor itself. This paper extends our work from [8] by describing a real-time prototype capable of two object tracking in Section 3, along with extensive simulations showing the benefit of foveating cameras at low resolutions for remote eye-tracking over a wide-FOV smartphone in Section 4.

Contemporary sensors focus on maximizing FOV at the expense of resolution to globally sample. This is at odds with computer vision algorithms which depend on rich visual data to accurately infer and reconstruct sparsely sampled image data. We propose using foveating cameras to quickly distribute a resolution-rich, low-FOV over a wide-FOV at prohibitive distances normally unobtainable without using large sensors or super-resolution algorithms that may not suffice for constrained robotic platforms. To enable accurate and simultaneous resolution distribution on targets, we adapt an efficient robot planning algorithm for MEMS mirror control that can optionally be integrated with a target tracker. Our new control algorithms enable quick, computationally light-weight updates of the MEMS mirror to simultaneously change the camera viewpoint between regions of interest.

We show the benefit of quickly modulating a high-resolution low FOV over a wide FOV with a MEMS mirror through remote eye-tracking, where both high resolution on facial features and wide FOV are preferrable but not possible with traditional cameras. We implement a recent convolutional neural network for eye-tracking and, through extensive simulations between our foveating camera and wide FOV smartphone using data collected in our lab, show the benefit of foveating cameras for resolution-FOV sensitive domains, such as remote eye-tracking. In summary, our contributions are:

(1) A novel sensor capable of *simultaneously* imaging several regions of interest in a scene by distributing its angular resolution with a fast MEMS mirror. We discuss optical and electronic calibrations along with a real-time prototype.
(2) An extension to 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) Demonstrating the benefit of adaptive resolution through increased remote eye-tracking performance compared to a conventional wide FOV smartphone with equivalent angular resolution.

- *Brevin Tilmon and Sanjeev Koppal are with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611 USA. E-mail: btilmon@ufl.edu, sjkoppal@ece.ufl.edu.*
- *Eakta Jain is with the Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL 32611 USA. E-mail: ejain@cise.ufl.edu.*
- *Silvia Ferrari is with the School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY 14850 USA. E-mail: sf375@cornell.edu.*

Manuscript received 11 June 2020; revised 29 Mar. 2021; accepted 31 Mar. 2021. Date of publication 15 Apr. 2021; date of current version 4 Aug. 2022. (Corresponding author: Brevin Tilmon.)
Recommended for acceptance by V. Koltun.
Digital Object Identifier no. 10.1109/TPAMI.2021.3071588

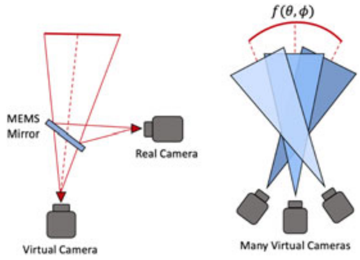


Fig. 1. Moving mirror creates many virtual views.

1.1 Related Work

Active vision and adaptive sampling. Ideas from visual attention [9], [10], have influenced robotics and vision, and information theoretic approaches are used to model adaptive 3D sensing for SLAM and other applications [11], [12]. Efficient estimation algorithms have been shown for adaptive visual and non-visual sensing on robots and point-zoom-tilt (PZT) cameras [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 [7], displays [15] and sensing [16]. 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, similar to [2]. 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 contrast, the foveating camera presented here passively uses mirrors to image regions of interest in real-world scenes, compared to calibration-target oriented work [24], [25].

Selective Imaging and Adaptive Optics. Our approach is similar in spirit to optical selective imaging with liquid crystal displays (LCDs) [26] and digital micro-mirror devices (DMDs) [16]. Because we use 2D scanning MEMS mirrors, we are able to allow the angular selectivity of [26] with the MEMS-enabled speed of [16]. 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 [27], [28], [29]. 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 [30].

Compressed Sensing. Our approach of selectively imaging what is related to optically filtering light-fields for imaging tasks [31], [32] and compressive sensing [33]. While there exist CS techniques for creating foveated imagery [28], [34], 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 [3], [35], [36], [37] and transient imaging [38].

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 [39], [40], outdoor advertising [41] and driver monitoring [42]. Depth and pose from stereo pairs has enabled gaze tracking from longer distances [43], [44]. 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.

Equiangular Cameras. A natural argument against foveated imaging is to use a large field of view equiangular sensor to image at the same high angular resolution as our foveating camera. We note that the current high cost of camera sensor fabrication actually encourages the use of our foveating camera, because we can modulate an equally dense angular resolution field of view using a camera sensor that is quadratically smaller, and therefore more inexpensive, than an equiangular camera that has the same angular resolution. Please see Section 1 of the supplementary, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2021.3071588 and Fig. 15 in Discussion for further explanations and simulation.

Fast Object Tracking With Galvanometer Mirrors. Unlike MEMS-modulated projectors and lidars, the mirror size is not correlated with light throughput when imaging with cameras like we do. Using the same camera sensor and lens, the mirror size affects the *incident aperture* size and how close the mirror must be to the camera to fill the camera field of view. However, once the camera field of view is filled exactly, the amount of light illuminating the camera sensor is the same no matter the mirror size that fills the camera field of view by similar triangles. Although the MEMS mirror size does not affect light throughput, the incident aperture that lets light hit the mirror plays a noise role we plan to explore in future work. Tracking with large galvo mirrors has been shown by [6]. This system adaptively illuminates an object moving through a FOV via optical flow, and does not distribute illumination to other objects.

2 Foveating Camera Prototype

Our prototype in Fig. 2 consists of a Mirrorcle Technologies 3.6 mm bonded MEMS mirror, Mirrorcle USB MEMS controller, FLIR BFS-U3-16S2C-CS with a 35 mm f/8 Edmund Optics lens, and custom synchronization circuitry for triggering the sensors. We chose the camera parameters and mirror size that make the camera FOV roughly the same size as a face at from 3 m to 5 m.

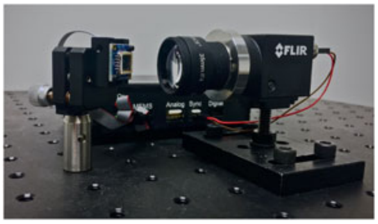


Fig. 2. Foveating camera prototype.

Using thin lens equations and our camera parameters, we can form a simulation for finding the camera-mirror distance D that minimizes vignetting from the mirror by

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

where θ is the cameras FOV for a given lens and sensor, M is the MEMS mirror diameter and α is the angle between the mirror and camera optical axis, which is typically $\pi/4$ by the manufacturer provided Brewster's angle, or the angle that polarized light completely passes through a transparent surface. Note we only take into account the mirror's coverglass Brewster angle, since the coverglass stays constant while the mirror moves. We used our simulation model mainly to determine what optics enable imaging a typical human head for eye-tracking purposes discussed in later sections.

Resolution Calibration. Our MEMS mirror comes with a protective coverglass that reflects light in the visible ranges and passes in the Near Infrared Ranges (800–900 nm), we wish to only image light reflected by the mirror and not the coverglass, therefore we aim to suppress the reflected coverglass light. We suppress coverglass ghosting by fitting a 800–900 nm bandpass filter to the camera, orienting the camera and MEMS mirror coverglass at $\pi/4$ satisfying Brewster's angle according to manufacturer specifications, and inserting an absorbing cover over ghosting-inducing reflective packaging surrounding the MEMS mirror. These calibrations enable much higher image quality as seen in Fig. 3.

In Section 4 we examine eye-tracking performance as a function of resolution for both our foveating camera and iPhone 6 smartphone. Our foveating camera uses a 1/3″ sensor, .003 mm pixel size, and 35 mm lens resulting in a 1080 × 1920 resolution while the iPhone 6 uses a 1/3″ sensor, .0015 mm pixel size and a 4.15 mm lens resulting in a 3024 × 2268 resolution. In Section 4 we simulate various optics and show associated performance for both cameras. We opt for a 35 mm lens because this enables us to tightly image faces at around three meters given our camera sensor.

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. 2, 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 FOVω_{mirror}.

Fig. 3. The coverglass induces ghosting and double images. With a combination of using Brewster's angle, NIR notch filter and absorbing film, we eliminate the ghosting.

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.

Consider a virtual plane Π perpendicular to the optical axis and parallel to the MEMS mirror in a resting, horizontal state, i.e., $(\theta = 0, \phi = 0)$. Π is a fixed distance from the MEMS mirror and is placed at the working distance of the camera where the subjects being imaged are. Every angular pose of the MEMS mirror (θ, ϕ) corresponds to a location (x, y) on Π given by perspective scaling. 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. We later discuss how to adapt this two target model to multiple targets.

Our goal is to move the mirror across a 1D line segment of length L_r that maximizes the chances of overlapping with the targets. Let one of the end points be denoted by (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, $\mathbf{U} = (L_{min}, L_{max}) \times (\omega_{min}, \omega_{max})$. $\mathbf{U}$ relates to (x_r, y_r, α_r) because (x_r, y_r, α_r), lying in plane Π, are constrained by $\mathbf{U}$. We encourage the reader to see Figure 1 in the supplementary material, available online, for a visual aid. *The problem of control requires a solution that changes the state q_r of the sensor to enable target imaging.*

L_{min} and L_{max} correspond to the min and max distance between regions of interest in Π with $L_{min} = 0$ and $L_{max} < 2\omega_{max}$. $\omega_{min} = 0^o$ and ω_{max} is given by

$$\omega_{max} = tan(\omega_{mems}) + FOV_{fovea} \tag{2}$$

where ω_{mems} corresponds to the MEMS mirror maximum tilt and FOV_{fovea} corresponds to the field of view for the camera imaging the MEMS mirror.

There are two ways to control the MEMS mirror to move in a 1D motion. The first is *point-to-point* and the second is using resonance, creating a *Lissajous* pattern.

Point-to-Point Algorithm. Given prior knowledge of the objects location in the scene, which can be given by a

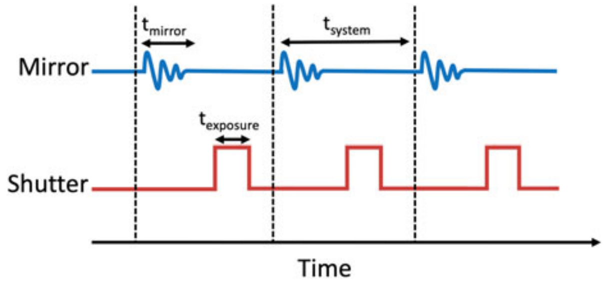


Fig. 4. Timing description of our foveating camera. t_{mirror} is the mirror settling time used to delay the camera trigger signal, $t_{exposure}$ is the camera exposure time and t_{system} is the total time of one frame capture after all motion and processing delays.

co-located sensor or known initialization, it is then possible to image each object through updating the respective dictionary mirror coordinates to keep the moving objects in frame.

Using the point-to-point control strategy, our foveating camera begins by initializing camera parameters and sending initial [x, y] coordinates to the MEMS controller. These coordinates indicate where the optical axis of the mirror points to on the arbitrary plane Π. The camera capture is triggered for every sample streamed to the mirror from the controller, and this trigger signal is delayed by 200 μs to allow the MEMS mirror to settle for clean images. The image is then processed, which explains the delay between mirror movement and camera exposure in Fig. 4, and the voltages sent to the MEMS mirror are scaled to satisfy a given criteria, such as keeping an object in frame at the given coordinate for that object. We add an additional delay before the mirror moves again for processing to finish. This strategy results in a total frame rate of $40Hz$ with the camera imaging 640×480 at 8-bit RGB. We use 640×480 resolution since real time imaging is infeasible at the native 1920×1080. See Fig. 4 for a visualization of our cameras timing.

Real-Time Demonstration. We now show a real-time demonstration of our foveating camera using the point-to-point control algorithm for tracking two red balls at 20 Hz for each ball in Figs. 5 and 6. The ghosting artifacts can be suppressed by inserting an absorbing filter around the MEMS mirror as mentioned in Section 2.

Lissajous Pattern. Our main control contribution is providing closed form differential updates to MEMS mirror 1D lissajous scanning motion. However, we only use lissajous scanning for the proof of concept control experiment in Section 5, and use point-to-point for the real time demonstration and data collection for eye tracking. The benefit of lissajous scanning over point-to-point is lower

Fig. 5. Helper smartphone image.

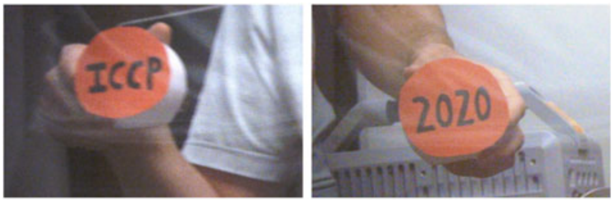


Fig. 6. Raw output of our foveating camera switching between two red objects based on point-to-point algorithm.

latency due to softer control on mirror coordinates. 1D lissajous scan patterns are realized by,

$$y(t) = Asin(2\pi f_x t + \phi_x). \tag{3}$$

Now consider a 1D lissajous wave 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.

Unlike the previous point-to-point method, the Lissajous pattern runs in resonance, which has both advantages and disadvantages. Speed and the lack of any settling time, as in Fig. 4 are an obvious advantage. However, since the MEMS mirror is in a balistic mode, images are obtained in the gaps between the target that must automatically be removed. Finally, the end points of the mirror motion may not be consistent, and therefore alignment must take place.

In the next section, we detail how to update q_r, the state of the mirror, which is general and impacts any technique for 1D mirror control for two targets.

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, available online, summarized briefly in Section 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. Proability distributions can be obtained from a co-located sensor instead of the Kalman filter. Our contribution mainly lies in the subsequent sections Sections 3.4 and 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 [45].

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 Section 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, available online. That supplementary material, available online, defines a control matrix $B_r(k)$ to update the state vector using the control vector $u_r(k)$

$$q_r(k+1) = \mathbf{I_3} q_r(k) + B_r(k) u_r(k) + Q_r, \tag{4}$$

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

$$q(k+1) = F\ q(k) + B(k)\ u(k) + w, \tag{5}$$

where F is a target motion matrix, based on optical flow equations, derived in the supplementary material, available online, 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 $P_k = [P_r(k)\quad 0; 0\ P_t(k)]$, 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

$$P(k+1) = [B_r(k)^T P_r B_r(k)\ 0; 0\ P_t] + [Q_r(k)\ 0; 0\ Q_t(k)], \tag{6}$$

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

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

$$q^{'}(k+1) = q(k+1) + K^{'}(z(k+1) - q(k+1)). \tag{8}$$

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

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 $q^{'}(k+1)$. This is useful to quantify the tracking performance of our system. 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 $q(k+1)$. The filter cannot be applied because the face detections are necessary to fully define our state space $q(k)$. Let P_d be the probability that all faces were detected successfully.

$$M_k = P_d \mathbf{E}[e^{'}(k+1)^T\ e^{'}(k+1)] + (1-P_d)\mathbf{E}[e(k+1)^T\ e(k+1)]. \tag{10}$$

where

$$e^{'}(k+1) = q(k) - q^{'}(k+1). \tag{11}$$

$$e(k+1) = q(k) - q(k+1). \tag{12}$$

(13)

Using the trace trick, similar to [45], we can convert M_k into an expression using the covariance matrices,

$$M_k = tr[P(k+1)] - P_d(tr[P(k+1)] - tr[P^{'}(k+1)]). \tag{14}$$

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,

$$max_{\mathbf{u_r}} P_d, \tag{15}$$

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,

$$P_d(q_r(k)) = \int_{s=0}^{L} f_t(x_r(k) + s\cos\alpha_r(k), y_r(k) + s\sin\alpha_r(k))ds, \tag{16}$$

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,

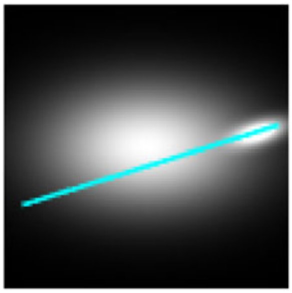


Fig. 7. Heterogeneity.

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 $f_l(x, y)$

$$\frac{1}{2\pi\sigma_{1l}\sigma_{2l}\sqrt{1-\rho_l^2}} e^{-\frac{\frac{(x-x_{lf})^2}{\sigma_{1l}^2} - \frac{2\rho_l(x-x_{lf})(y-y_{lf})}{\sigma_{1l}\sigma_{2l}} + \frac{(y-y_{lf})^2}{\sigma_{2l}^2}}{2(1-\rho_l^2)}}, \tag{17}$$

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

$$\frac{1}{2\pi\sigma_{1r}\sigma_{2r}\sqrt{1-\rho_r^2}} e^{-\frac{\frac{(x-x_{rf})^2}{\sigma_{1r}^2} - \frac{2\rho_l(x-x_{rf})(y-y_{rf})}{\sigma_{1r}\sigma_{2r}} + \frac{(y-y_{rf})^2}{\sigma_{2r}^2}}{2(1-\rho_r^2)}}. \tag{18}$$

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 [46] 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). 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 as in Fig. 8 Ic. 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 as in 8 II(c). In other words, the domain, within the Radan transform of bi-variate Gaussians, where we must search for the maximal slice, is bounded.

Algorithm 1. Gradient-Based Update of Control Vector u_r

Input: Kalman filter outputs, valid space $\mathbf{U}$, 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 + η δP_d(q_r(k+1)) / δu_r
  if u_r^tmp ∉ U
    return
    else if ||u_r^tmp − u_r|| < ε
    return
    else
    u_r = u_r^tmp
  end if
end
return u_r
```

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. Second, the heuristic tends to work when the Gaussians are similar; if one Gaussian dominates, as in Fig. 7, 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 $\mathbf{u_r}$

$$\frac{\delta P_d(q_r(k+1))}{\delta \mathbf{u_r}} = \frac{\delta P_d(q_r(k+1))}{\delta q_r(k+1)} \frac{\delta q_r(k+1)}{\delta \mathbf{u_r}}. \tag{19}$$

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

$$\frac{\delta P_d(q_r(k+1))}{\delta q_r(k+1)} = \begin{bmatrix} \frac{\delta}{\delta x_r} P_d(q_r(k+1)) \\ \frac{\delta}{\delta y_r} P_d(q_r(k+1)) \\ \frac{\delta}{\delta \alpha_r} P_d(q_r(k+1)) \end{bmatrix}. \tag{20}$$

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

$$\frac{\delta P_d(q_r(k+1))}{\delta q_r(k+1)} = \begin{bmatrix} \frac{\delta}{\delta x_r} \int_{s=0}^{L} f_l(x,y)ds \\ \frac{\delta}{\delta y_r} \int_{s=0}^{L} f_l(x,y)ds \\ \frac{\delta}{\delta \alpha_r} \int_{s=0}^{L} f_l(x,y)ds \end{bmatrix} + \begin{bmatrix} \frac{\delta}{\delta x_r} \int_{s=0}^{L} f_r(x,y)ds \\ \frac{\delta}{\delta y_r} \int_{s=0}^{L} f_r(x,y)ds \\ \frac{\delta}{\delta \alpha_r} \int_{s=0}^{L} f_r(x,y)ds \end{bmatrix}. \tag{21}$$

These gradients can easily be calculated after every iteration of the Kalman filter, allowing for the closed form

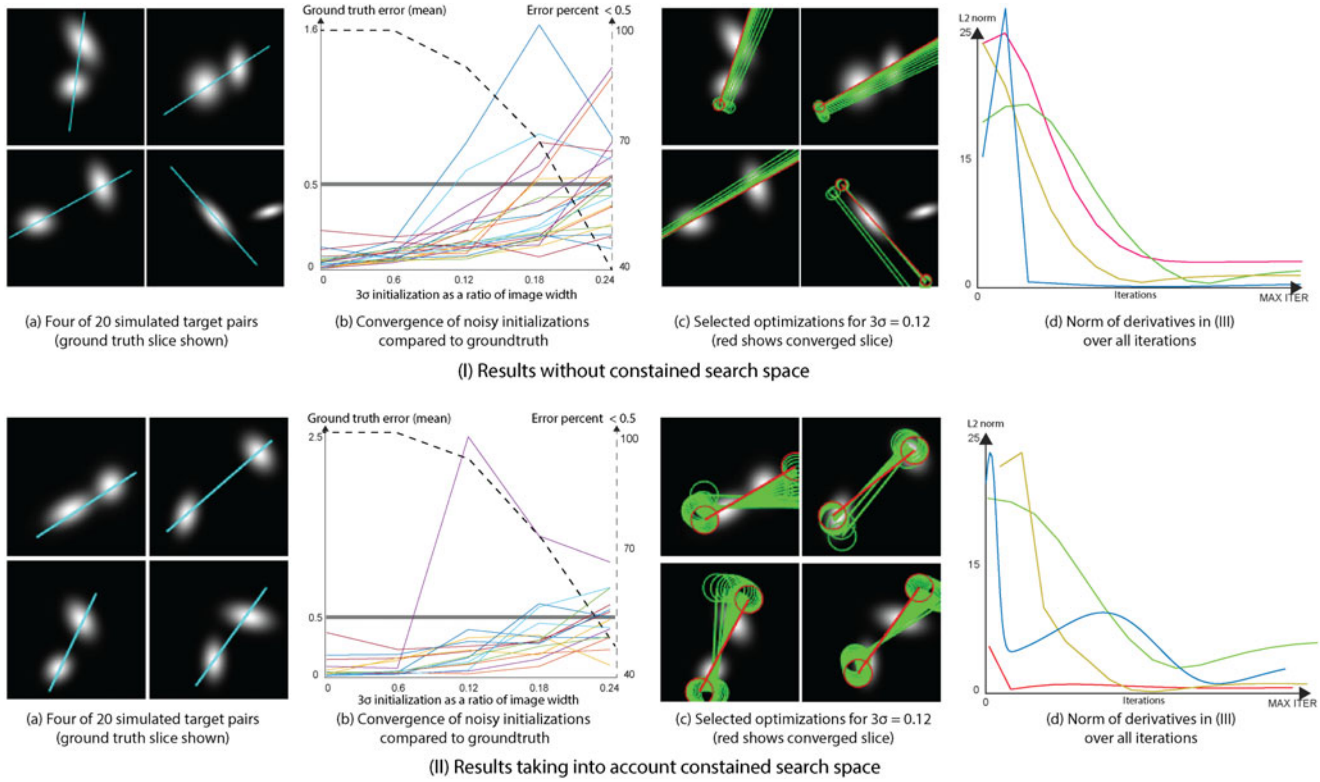


Fig. 8. Simulations of 1D slice optimization: In (I) and (II) we created simulations to test the iterative optimization in Algorithm 1. (I) is a free-form optimization, whereas (II) constrains the optimization along the proposed bounded region. Note that the percent error for (II) is slightly lower.

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, available online. In Algorithm 1 we use these gradients to update the control vector.

Simulations. In Fig. 8 we show simulations of Algorithm 1 on 20 pairs of 2D Gaussians. In Fig. 8 Ia we select four from these 20, showing the ground-truth "slice" that maximizes target probability, P_d , calculated from the radon transform. In Fig. 8 Ib 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 percent of the image width. This means that at the extreme case, initialization could be anywhere in a 50 percent chunk of the image near the ground-truth. Fig. 8 II 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. 8 IIb.

Fig. 8 I-IIb graphs show euclidean distance between the converged slice and ground truth, averaged over five trials. Note that most results converge even for large deviations from the ground-truth. In Fig. 8 I-IIc we show the convergence path for these examples, and in Fig. 8 I-IId 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 such as the learning rate. 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.

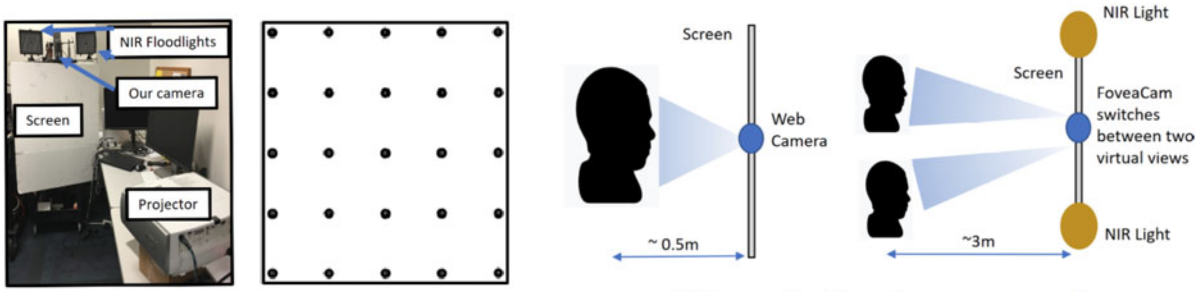
4 Remote Eye-Tracking for Frontal Faces

We demonstrate the benefit of modulating a dense low-FOV over a wide-FOV through remote eye-tracking, where both high angular resolution and wide FOV for multi-person imaging are necessary. 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 the eye tracking performance using the iTracker convolutional neural network [47] for both our foveating camera and a near-co-located smartphone. We also present a proof-of-concept remote eye-tracking system that uses our MEMS mirror enabled foveating camera to capture images.

4.1 Our Eye-Tracking Setup

Our setup, shown in Fig. 9, 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 a 5×5 grid of points spanning the width and height of the lambertian plane, as in the figure.



(I) Our eye-tracking setup (II) Dataset gaze patterns projected onto screen beneath camera (III) We extend traditional short-range eye-tracking (left) to two people at 3m(right)

Fig. 9. Our eye tracking setup and gaze pattern used for our finetuning dataset.

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 Section 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 [47] 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 to the camera; e.g., the output is (0, 0) for someone looking directly at the camera.

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

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, $25\,\mathrm{cm}$ in iTracker versus $3\,\mathrm{m}$ for us) which changes 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 3 m from the camera.

4.3 Data Collection for Fine-Tuning

The network performs poorly using the provided network weights at the same span of test points at 3 m as the iPhone tests. This is expected since viewing a 12 cm spanned x, y pattern (iPhone) at 3 m gives less than 1^o eye angle. Commercial eye trackers typically employ 1^o 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. 10 I) enabled the collection of fine-tuning data in-situ with the device, in NIR, for the grid pattern in Fig. 9 along with the smartphone. 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 randomly split the dataset into 23,000 train, 4,000 validation, and 6,000 test for our foveating camera and smartphone. For fine-tuning, we begin with identical weights to [47], except we lower our learning rate ten fold. We do not freeze any layers. We found 10 epoch fine-tuning to fit our dataset properly, and all results in this section are from 10 epoch fine-tuning.

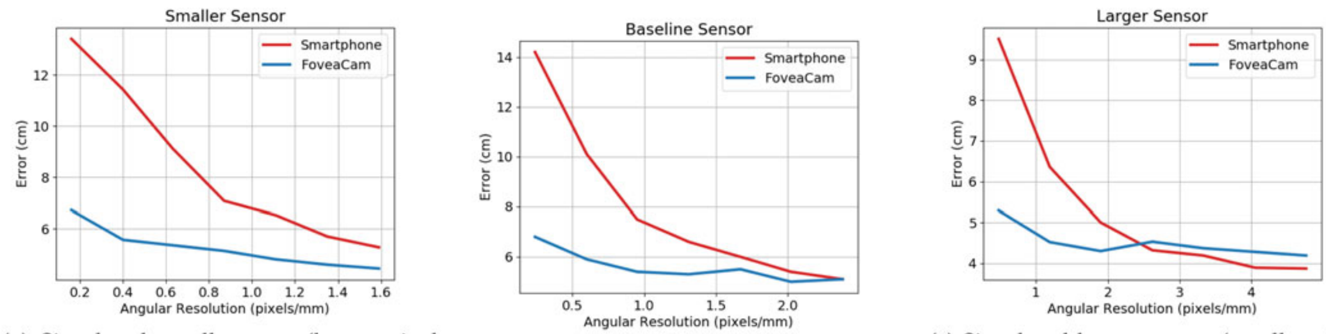
4.4 Experimental Results

In our experiments, the subjects were at $3\,\mathrm{m}$ distance and six people were involved overall, four for training and testing,

Fig. 10. Sample images from our dataset. Our foveating camera data is on the left and iPhone 6 smartphone is on the right. Some faces are off center due to this being a real experiment with the camera moving.

TABLE 1
Random Initialization Fails (Train/Val error)

Camera	L2 Error (10 epochs) (cm)
Smartphone (random initial.)	55.91
Smartphone (iTracker initial.)	6.5
Foveating camera (random initial.)	45.77
Foveating camera (iTracker initial.)	4.45



(a) Simulated small sensor (larger pixel size) and lenses error vs angular resolution.

(b) Baseline sensor and simulated lenses error vs angular resolution.

(c) Simulated larger sensor (smaller pixel size) and lenses error vs angular resolution

Fig. 11. Angular resolution experiment results.

two for the proof of concept experiment in Section 5. To show that this relatively small fine-tuning dataset does not adversely affect our results, we show, in Table 1, that validation errors after 10 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 10 epochs, and we do indeed utilize the 1,400 users encapsulated in the pre-trained weights.

Simulating Angular Resolution. We now show the benefit of our foveating camera by analyzing eye tracking error as a function of simulated angular resolutions for our foveating camera and smartphone. We introduce a simulation model to downsample and then upsample network inputs, changing the angular resolution of the inputs. We finetune, validate, and test the network all using simulated network inputs according to the below simulation model. The test data is reshuffled for each simulation while the network hyper-parameters remain identical to Section 4.2.

Simulation Model. We outline the equations and provide code for our simulation model in Section 2 of the supplementary, available online, but give the main idea here. We pick different camera parameters for both our foveating camera and smartphone. We then downsample and upsample the network inputs based on how many pixels from each new simulated camera are left over in the original field of view to simulate angular resolution loss.

Simulation Results. Our simulations show that we can maintain low eye-tracking error even at very small image resolutions. Figs. 11a, 11b, and 11c demonstrate the eye tracking performance drastically degrading for the smartphone as angular resolution decreases while our foveating camera error degrades more gradually and has a much lower extreme. Our foveating camera and smartphone converged to similar errors at high angular resolutions with the smartphone performing slightly better at image sizes above 2MP. Even though the simulated angular resolution of our foveating camera and smartphone are equivalent on Figs. 11a, 11b, and 11c x axis, downsampling and upsampling causes different degradation's for our foveating camera and smartphone since their images were sampled at different native angular resolutions. We do not see the smartphone outperform the foveating camera until we use a smaller pixel size (larger number of pixels) in Fig. 11c. The smartphone is able to beat the foveating camera because the foveating camera is degraded just enough after our resizing operation in comparison to the smartphone to make the resulting true angular resolution worse than the smartphone true angular resolution after our resizing operation.

See Figs. 12 and 13 for a visualization of the iTracker network output for our foveating camera and smartphone, respectively. The L2 error of our foveating camera was 5.18cm and the smartphone was 11.06 cm. The camera parameters for this visualization include increasing sensor pixel sizes by 1.5, using 5 mm and 2.5 mm lenses for our foveating camera and smartphone, respectively, giving a common vertical angular resolution of .4 pixels/mm between the smartphone and foveating camera, as in in Fig. 11a. Since our native FOV was 466 mm $\times$ 262 mm (W $\times$ H), the image size at these parameters was 332 $\times$ 104, this clearly shows the benefit of our camera: we are able to sacrifice significant resolution and maintain high performance.

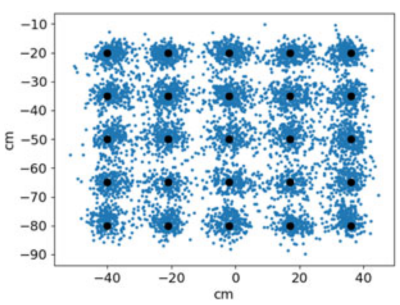


Fig. 12. Raw network output for our foveating camera test data with 5.15cm error. The ground truth locations are black dots.

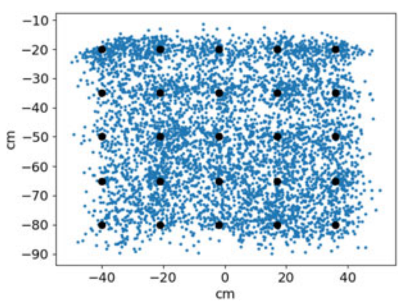


Fig. 13. Raw network output for smartphone test data with 11.06 cm error. Ground truth locations are black dots.

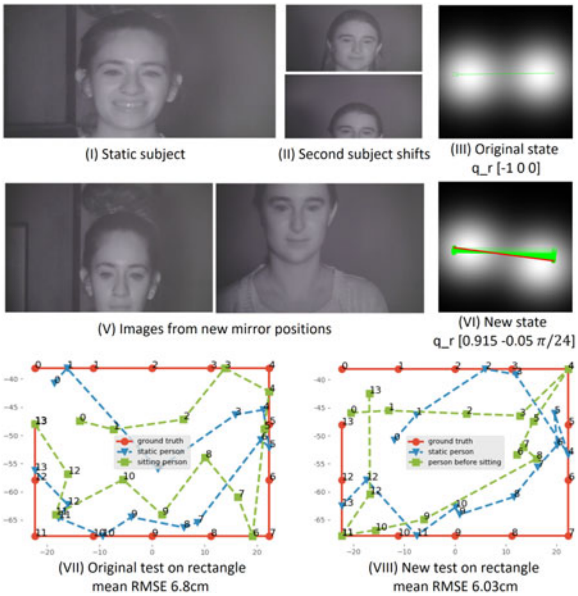


Fig. 14. Proof-of-concept control experiments where one subject moves slightly, and Algorithm 1 is used to reset the mirror positions. Eye tracks are provided for start state and the end state, after mirror motion.

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 not used in training, validating, or testing the network are looking at a square pattern the network has not seen. This pattern has a smaller span than the 5×5 grid used in Section 4.

We use a the bounding box from a simple face-tracker [48] as a proxy for the Kalman filter, and use the a user defined ratio $k \approx 3$ 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 Fig. 14, we show the initial state of the scene for the two test subjects and the corresponding gaze track for the square 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. Algorithm 1 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 pattern, redone at this new mirror position, also produces good quality results ($6.8\,\mathrm{cm}$ and $6.03\,\mathrm{cm}$ L2 error respectively).

6 Conclusion

Limitations. Multi-object tracking with 2D lissajous scanning is an area of focus moving forward, and we hope to provide derivations for these mirror position updates in future work to move towards a generalized control law.

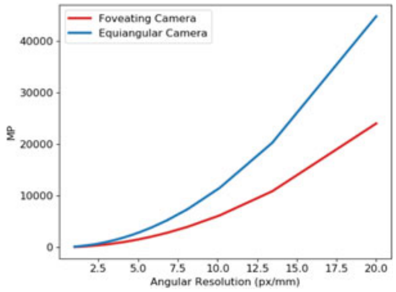


Fig. 15. As angular resolution increases, foveating cameras can obtain the same angular resolution with quadratically less pixels than their potentially gigapixel camera counterpart.

Integrating an auto-focusing element such as a liquid lens into our camera would improve the shallow depth of field of our camera caused by the MEMS mirror size. Liquid lenses are easily embedded into camera systems and would allow for increased imaging distance and depth of field.

While our camera is fairly compact at 15cm $\times$ 10cm $\times$ 10 cm, we acknowledge this size will need to be reduced before foveating cameras could easily be integrated into robotic imaging systems. This could be accomplished by replacing the bulky MEMS controller with a MEMS driver board giving dimensions of 3.5cm x 4cm x 1cm, and using optics and sensor housing that optimize for smallest possible form factor. We also note that using a MEMS driver board instead of MEMS controller should increase the frame rate of the camera due to less software overhead from the ease of use the controller provides.

Discussion. Further comparison with competing sensors and datasets is necessary to further show our camera's performance. We provide initial simulations comparing our foveating image capture technique to an equivalent full frame camera in Fig. 15 and in the supplementary, available online.

Foveating cameras change a cameras viewpoint such that it can see multiple regions of interest at resolutions and speeds typically not possible. Quickly modulating a pixel-dense camera viewpoint has direct applications to robotics, augmented reality and autonomous vehicles where densely sampling specific regions could help complete 3D reconstructions, aid long range visual navigation tracking, and increase safety by increased sampling on critical regions of interest.

Acknowledgments

The work of Brevin Tilmon and Sanjeev Koppal was supported in part by the National Science Foundation under Grant NSF IIS: 1909729 and in part by the Office of Naval Research under Grant ONR N00014-18-1-2663. The work of Eakta Jain was supported by the National Science Foundation under Grant NSF: 1566481. The work of Silvia Ferrari was supported by the Office of Naval Research under Grant N00014-19-1-2144.

References

[1] Z. Tasneem, D. Wang, H. Xie, and S. J. Koppal, "Directionally controlled time-of-flight ranging for mobile sensing platforms," in *Proc. Robot., Sci. Syst.*, 2018, pp. 1–10.

[2] 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,” 2020, *arXiv: 2003.09545*.

[3] S. Achar, J. R. Bartels, W. L. Whittaker, K. N. Kutulakos, and S. G. Narasimhan, “Epipolar time-of-flight imaging,” *ACM Trans. Graph.*, vol. 36, no. 4, 2017, Art. no. 37.

[4] Z. Tasneem, C. Adhivarahan, D. Wang, H. Xie, K. Dantu, and S. J. Koppal, “Adaptive fovea for scanning depth sensors,” *Int. J. Robot. Res.*, vol. 39, pp. 837–855, 2020.

[5] K. Henderson, X. Liu, J. Folden, B. Tilmon, S. Jayasuriya, and S. J. Koppal, “Design and calibration of a fast flying-dot projector for dynamic light transport acquisition,” *IEEE Trans. Comput. Imag.*, vol. 6, pp. 529–543, 2020.

[6] H. O. Kohei Okumura and M. Ishikawa, “High-speed gaze controller for millisecond-order pan/tilt camera,” *Proc. IEEE Int. Conf. Robot. Automat.*, 2011, pp. 6186–6191.

[7] J. Wang, J. Bartels, W. Whittaker, A. C. Sankaranarayanan, and S. G. Narasimhan, “Programmable triangulation light curtains,” in *Proc. Eur. Conf. Comput.*, 2018, pp. 19–34.

[8] B. Tilmon, E. Jain, S. Ferrari, and S. Koppal, “FoveaCam: A MEMS Mirror-Enabled Foveating Camera,” in *Proc. IEEE Int. Conf. Comput. Photography*, 2020, pp. 1–11.

[9] S. Frintrop, E. Rome, and H. I. Christensen, “Computational visual attention systems and their cognitive foundations: A survey,” *ACM Trans. Appl. Perception*, vol. 7, no. 1, 2010, Art. no. 6.

[10] N. Bruce and J. Tsotsos, “Attention based on information maximization,” *J. Vis.*, vol. 7, no. 9, 2007, Art. no. 950.

[11] B. Charrow *et al.*, “Information-theoretic planning with trajectory optimization for dense 3D mapping,” in *Proc. Robot., Sci. Syst.*, 2015, pp. 1–10.

[12] E. Ristani and C. Tomasi, “Features for multi-target multi-camera tracking and re-identification,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6036–6046.

[13] C. Cadena *et al.*, “Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age,” *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016.

[14] 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 Trans. Autom. Control*, vol. 64, no. 1, pp. 159–173, Jan. 2019.

[15] A. Jones, I. McDowall, H. Yamada, M. Bolas, and P. Debevec, “Rendering for an interactive 360 light field display,” *ACM Trans. Graph.*, vol. 26, no. 3, pp. 40–es, 2007.

[16] S. K. Nayar, V. Branzoi, and T. E. Boult, “Programmable imaging: Towards a flexible camera,” *Int. J. Comput. Vis.*, vol. 70, pp. 7–22, 2006.

[17] T. P. Flatley, *Spacecube: A Family of Reconfigurable Hybrid on-Board Science Data Processors.* Washington, DC, USA: NASA, 2015.

[18] B. L. Stann *et al.*, “Integration and demonstration of MEMS–scanned LADAR for robotic navigation,” in *Proc. SPIE*, 2014, pp. 177–186.

[19] K. T. Krastev, H. W. Van Lierop, H. M. Soemers, R. H. M. Sanders, and A. J. M. Nellissen, “MEMS scanning micromirror,” U.S. Patent 8 526 089, Sep. 3, 2013.

[20] A. Kasturi, V. Milanovic, B. H. Atwood, and J. Yang, “Uav-borne lidar with MEMS mirror-based scanning capability,” *Proc. SPIE*, vol. 9832, 2016, Art. no. 98320M.

[21] V. Milanović, A. Kasturi, J. Yang, and F. Hu, “A fast single-pixel laser imager for VR/AR headset tracking,” *Proc. SPIE*, vol. 10116, 2017, Art no. 101160E.

[22] V. Milanović, A. Kasturi, N. Siu, M. Radojičić, and Y. Su, “MEMSEye for optical 3D tracking and imaging applications,” in *Proc. 16th Int. Solid-State Sensors, Actuators Microsystems Conf.*, 2011, pp. 1895–1898.

[23] T. Sandner *et al.*, “Hybrid assembled micro scanner array with large aperture and their system integration for a 3D ToF laser camera,” in *Proc. SPIE*, vol. 9375, 2015, Art no. 937505.

[24] T. Tang, Y. Huang, C. Fu, and S. Liu, “Acceleration feedback of a CCD-based tracking loop for fast steering mirror,” *Opt. Eng.*, vol. 48, no. 1, pp. 1–6, 2009.

[25] N. Chen, B. Potsaid, J. T. Wen, S. Barry, and A. Cable, “Modeling and control of a fast steering mirror in imaging applications,” in *Proc. IEEE Int. Conf. Automat. Sci. Eng.*, 2010, pp. 27–32.

[26] A. Zomet and S. K. Nayar, “Lensless imaging with a controllable aperture,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2006, pp. 339–346.

[27] G. Sandini and G. Metta, “Retina-like sensors: Motivations, technology and applications,” in *Sensors and Sensing in Biology and Engineering*. Vienna, Austria: Springer, 2003, pp. 251–262.

[28] S. Liu, C. Pansing, and H. Hua, “Design of a foveated imaging system using a two-axis MEMS mirror,” in *Proc. Int. Opt. Des. Conf.*, 2006, pp. 869–876.

[29] H. Hua and S. Liu, “Dual-sensor foveated imaging system,” *Appl. Opt.*, vol. 47, no. 3, pp. 317–327, 2008.

[30] T. Nakao and A. Kashitani, “Panoramic camera using a mirror rotation mechanism and a fast image mosaicing,” in *Proc. Int. Conf. Image Process.*, 2001, pp. 1045–1048.

[31] R. Ng, “Fourier slice photography,” in *ACM Trans. Graph.*, vol. 24, no. 3, pp. 735–744, 2005.

[32] A. Levin, R. Fergus, F. Durand, and W. T. Freeman, “Image and depth from a conventional camera with a coded aperture,” *ACM Trans. Graph.*, vol. 26, no. 3, pp. 70–es, 2007.

[33] M. B. Wakin *et al.*, “An architecture for compressive imaging,” in *Proc. Int. Conf. Image Process.*, 2006, pp. 1273–1276.

[34] I. B. Ciocoiu, “Foveated compressed sensing,” *Circuits, Syst., Signal Process.*, vol. 34, no. 3, pp. 1001–1015, 2015.

[35] M. Gupta, S. K. Nayar, M. B. Hullin, and J. Martin, “Phasor imaging: A generalization of correlation-based time-of-flight imaging,” *ACM Trans. Graph.*, vol. 34, no. 5, 2015, Art. no. 156.

[36] M. O. T’oole, 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 Trans. Graph.*, vol. 33, no. 4, 2014, Art. no. 87.

[37] M. O. T’oole, S. Achar, S. G. Narasimhan, and K. N. Kutulakos, “Homogeneous codes for energy-efficient illumination and imaging,” *ACM Trans. Graph.*, vol. 34, no. 4, 2015, Art. no. 35.

[38] A. Velten *et al.*, “Imaging the propagation of light through scenes at picosecond resolution,” *Commun. ACM*, vol. 59, no. 9, pp. 79–86, 2016.

[39] 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 Trans. Consum. Electron.*, vol. 58, no. 4, pp. 1119–1128, Nov. 2012.

[40] C. Hennessey and J. Fiset, “Long range eye tracking: Bringing eye tracking into the living room,” in *Proc. Symp. Eye Tracking Res. Appl.*, 2012, pp. 249–252.

[41] 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 *Proc. 30th Annu. ACM Symp. User Interface Softw. Technol.*, 2017, pp. 155–166.

[42] O. Palinko, A. L. Kun, A. Shyrokov, and P. Heeman, “Estimating cognitive load using remote eye tracking in a driving simulator,” in *Proc. Symp. Eye-Tracking Res. Appl.*, 2010, pp. 141–144.

[43] D. Beymer and M. Flickner, “Eye gaze tracking using an active stereo head,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2003, pp. II–451.

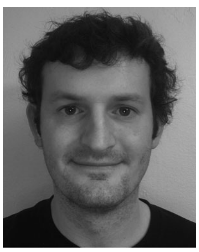
[44] D. Geisler, D. Fox, and E. Kasneci, “Real-time 3D glint detection in remote eye tracking based on Bayesian inference,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 7119–7126.

[45] H. Wei, W. Lu, P. Zhu, G. Huang, J. Leonard, and S. Ferrari, “Optimized visibility motion planning for target tracking and localization,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 76–82.

[46] E. W. Weisstein, “Radon transform–gaussian. From MathWorld—A Wolfram Web Resource,” Accessed: Aug. 10, 2019. [Online]. Available: http://mathworld.wolfram.com/RadonTransformGaussian.html

[47] K. Krafka *et al.*, “Eye tracking for everyone,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2176–2184.

[48] M. Jones and P. Viola, “Fast multi-view face detection,” Mitsubishi Electric Research Lab, Cambridge, MA, USA, Tech. Rep. TR-20003–96, vol. 3, no. 14, p. 2, 2003.



Brevin Tilmon is currently working toward the PhD degree from the Electrical and Computer Engineering Department, University of Florida. His research interests include computational photography, computer vision, and machine learning.

Eakta Jain received the BTech degree in electrical engineering from IIT Kanpur, and the MS and PhD degrees in robotics from the Graphics Lab, Carnegie Mellon University. She is currently an assistant professor of computer and information science and engineering at the University of Florida. She was in industrial research with Texas Instruments Research and Development labs, Disney Research Pittsburgh, and the Walt Disney Animation Studios. Her research interests include 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 (Senior Member, IEEE) received the BS degree from Embry–Riddle Aeronautical University, and the MA and PhD degrees from Princeton University. She is currently a John Brancaccio professor of mechanical and aerospace engineering at Cornell University. Before that, she was a professor of engineering and computer science with Duke University, and the founder and director with the NSF Integrative Graduate Education and Research Traineeship (IGERT) and Fellowship program on wireless intelligent sensor networks. She is currently the Director with the Laboratory for Intelligent Systems and Controls, Cornell University. Her research interests include robust adaptive control of aircraft, learning and approximate dynamic programming, and optimal control of mobile sensor networks. She is a senior member of the IEEE, and a member of ASME, SPIE, and AIAA. She was 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 received the BS degree from the University of Southern California in 2003, and the master’s and PhD degrees from the Robotics Institute, Carnegie Mellon University (CMU). He was a postdoctoral research associate with the School of Engineering and Applied Sciences, Harvard University. She is currently an assistant professor at the Department of Electrical and Computer Engineering, University of Florida. Before joining the University of Florida, he was a researcher with the Texas Instruments Imaging Research and Development lab. His research interests include computer vision, computational photography and optics, novel cameras and sensors, 3D reconstruction, physics-based vision, and active illumination.