FOVEATED COMPUTATIONAL IMAGING

By

BREVIN TILMON

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2023

"You can't feel speed, only acceleration"
- David Holz, Midjourney

ACKNOWLEDGEMENTS

I would first like to thank my PhD advisor, Sanjeev Koppal, for his unwavering patience and guidance. This work would not have been possible without his support. I would like to thank my mother, father, and brother for giving me a firm foundation to explore life with. I would like to thank Nerea for giving beautiful color to my life. I would like to thank Jagger, Coleton, Aaron, and Dylan for their friendship over the years.

I would like to thank Uland Wong, Michael Dille, Shuochen Su, Michael Hall, Jian Wang and Sizhuo Ma among others for wonderful research internships at NASA, Meta, and Snap. These experiences significantly grew my engineering and research aptitude in unique ways and I am grateful. I would like to thank my colleagues old and new in the Florida Optics and Computational Sensor Lab for their support and friendship.

Finally, I would like to thank the Florida beaches and ocean for always being there for me after a deadline.

TABLE OF CONTENTS

page

LIST OF TABLES

Tables page

LIST OF FIGURES

Figures page

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

FOVEATED COMPUTATIONAL IMAGING

By

Brevin Tilmon

December 2023

Chair: Sanjeev Jagannatha Koppal
Major: Electrical and Computer Engineering

Foveation is an emergent trait of biological vision systems that concentrates sensing resources only where necessary for survival. Inspired by the evolution of foveation over millions of years, this dissertation presents foveated algorithms, cameras, and depth sensors that use similar ideas as biological vision systems to improve computer vision and machine learning performance without using more sensing resources. The results from this dissertation suggest that foveated sensing is a promising direction for the design of imaging sensors in the future.

CHAPTER 1
INTRODUCTION

1.1 Foveation

Foveation has naturally evolved in biological vision systems, principally as an adaptive mechanism for survival. The majority of animals, including humans, possess a retina in the eye where the center is densely packed with cones in contrast to the peripheral regions. This non-uniform distribution of cones is not a design flaw but rather an evolutionary feature to enhance visual acuity precisely in the direct line of sight. Essentially, this specialized arrangement gives the eye super-resolution capabilities.

Due to limitations set forth by evolutionary constraints, animals are inherently equipped with a fixed, non-expandable number of cones in their eyes. Consequently, the eye has cleverly evolved to utilize this constraint to its advantage by more tightly packing these limited cones in the center of the visual field. This elegant design amplifies the eye's maximum resolving capabilities, allowing for sharper and more detailed vision where it matters most. While the exact reasons for this evolutionary development are still a matter of scientific exploration, it is widely believed that this adaptation serves functions such as facilitating more effective hunting, enabling nuanced communication, and providing for a diverse set of other visual tasks that would be substantially more challenging if the fixed number of cones were dispersed uniformly across the entire field of view.

This dissertation applies these insights from foveation in biological systems to artificial imaging systems. By doing so, the objective is to substantially boost the performance of downstream computer vision technologies without the need for additional sensing resources. Specifically, we can draw parallels between the cones in the biological retina and the pixels or laser power in artificial imaging systems like cameras, LiDAR, and radar. Adopting the principles of foveation in these technologies has a multitude of theoretical as well as practical implications. It not only provides a new paradigm for the design and functionality of imaging sensors, but also offers avenues for more efficient and resource-effective systems, ranging from everyday digital cameras to advanced LiDAR and radar systems utilized in a wide array of applications.

1.2 Dissertation Organization

In addition to the introduction and conclusion, there are three technical chapters that outline the contributions in this dissertation. These technical chapters are outlined below.

In Chapter 2, we investigate the limitations and bottlenecks faced by contemporary camera systems that typically capture their entire visual field indiscriminately [115, 114]. While the realm of active vision research over the past several decades has proposed a plethora of foveating camera designs aimed at selective scene viewing, the impact of such innovations has largely been constrained by the slow, often cumbersome options available for mechanical camera movement. To address this pressing issue, we introduce a novel foveating camera design called FoveaCam. FoveaCams obtain high-resolution imagery concentrated on multiple regions of interest. We discuss the complete hardware and software design of the FoveaCam hardware prototype. We then show that FoveaCam improves machine learning based eye tracking performance at long ranges. We then show extremely long range results with FoveaCam adaptively zooming onto multiple regions of interest 1000 meters away.

In Chapter 3, we address the limitations of active depth sensing, particularly the trade-offs between sensing range, power consumption, and eye safety [117]. We propose a foveating active depth sensor that focuses light patterns only on specific regions of interest where depth information is crucial and where traditional passive stereo methods fall short. Through a comparative analysis with existing methods like full-frame projection, line scanning, and point scanning, our adaptive approach proves to consume the least power while maintaining optimal eye-safety distance and achieving the same maximum sensing range. We validate these findings with two hardware prototypes: one using a phase-only spatial light modulator (SLM) and another utilizing a micro-electro-mechanical (MEMS) mirror combined with a diffractive optical element (DOE). Experimental results confirm that our method efficiently estimates higher-quality geometry while maintaining eye safety.

In Chapter 4, we investigate an end-to-end learning based control approach for foveating imaging systems, termed SaccadeCam [116]. Our approach employs a self-supervised network

for foveating camera resolution, specifically tailored for monocular depth estimation. We show that it is possible to learn where to distribute pixels to boost monocular depth estimation compared to cameras that use the same resolution but undistributed. We show experiments in simulation on the Kitti self driving car dataset [36] and in the wild on a foveating camera prototype running neural networks for control on an embedded system.

CHAPTER 2
PASSIVE FOVEATED IMAGING

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 [96, 2, 33, 12], although these have been constrained by relatively slow pan-zoom-tilt (PZT) cameras and robot motion.

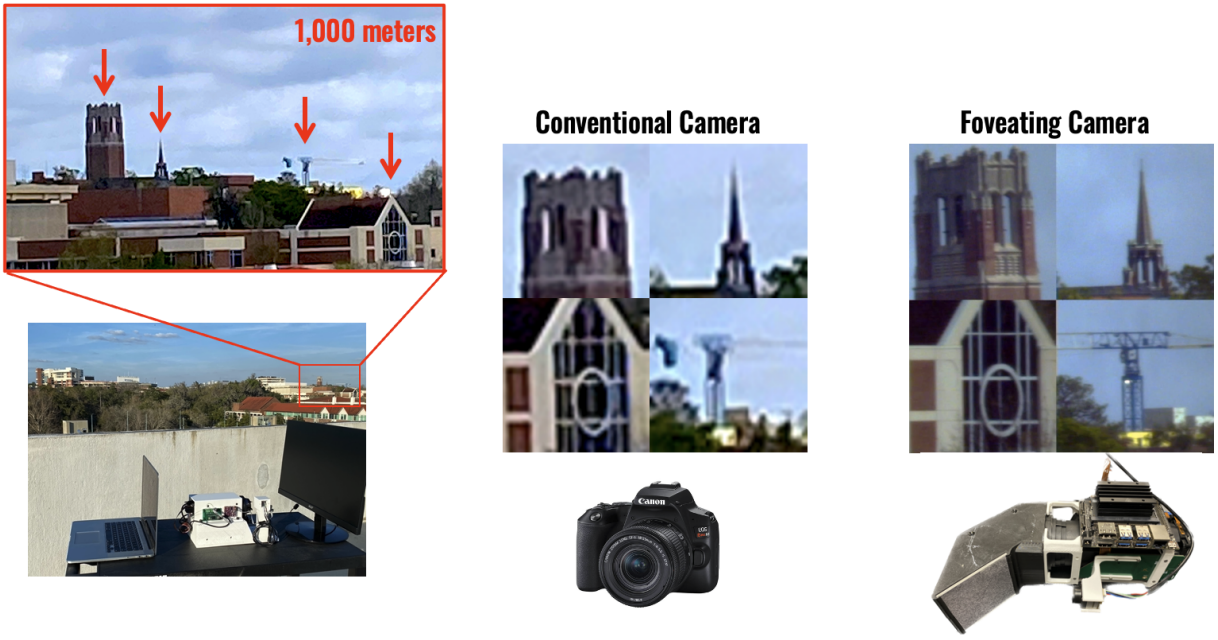


Figure 2-1. Real time foveation on multiple regions of interest [115, 114].

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 [84, 126], we use them to modulate *viewing direction*, much like catadioptric cameras [39].

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 a single passive camera 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.

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-resolution commercial smartphone camera at the same location and

with the same resolution.

2.1 Related Work

Active vision and adaptive sampling. Ideas from visual attention [33, 12], have influenced robotics and vision, and information theoretic approaches are used to model adaptive 3D sensing for SLAM and other applications [113, 17, 103, 25]. Efficient estimation algorithms have been shown for adaptive visual and non-visual sensing on robots and point-zoom-tilt (PZT) cameras [13, 130, 28]. 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 [101], displays [59] and sensing [84]. We use MEMS mirrors to modulate viewing direction. MEMS mirrors used in LIDARs, such as from NASA and ARL [31, 108, 69], are run at resonance, while we control the MEMS scan pattern for novel imaging strategies. Such MEMS uses have been shown [63] for highly reflective fiducials in both fast 3D tracking and VR applications [82, 81]. We do not use special reflective fiducials and utilize active vision algorithms for MEMS mirror control. [105] 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 [48], seeing around corners [87] and reconstruction with light curtains [126]. 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 [110, 18]. Our research is closest to [54] 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) [138] and digital micro-mirror devices (DMDs) [84]. Because we use 2D scanning MEMS mirrors, we are able to allow the angular selectivity of [138] with the MEMS-enabled speed of [84]. 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 [104, 74]. Another related approach that uses fast optics for incident viewing is

atmospheric sensing through turbulence with fast adaptive optics [9] 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 [83].

Compressed sensing. Our approach of selectively imaging what is related to optically filtering light-fields for imaging tasks [100, 85, 70] and compressive sensing [124]. While there exist CS techniques for creating foveated imagery [22, 74], 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 [45, 86, 90, 1] and transient imaging [123, 49].

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 [21, 50], smart offices [25], outdoor advertising [65] and driver monitoring [93]. Depth and pose from stereo pairs has enabled gaze tracking from longer distances [11, 37]. 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. [23] 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 [23] may prove burdensome.

Fast tracking with galvanometer mirrors. Tracking with large galvo mirrors has been shown by [66]. 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 [66]; a single MEMS mirror can rotate in two dimensions due to the gimbal-swivel design. Finally, [66] 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.

2.2 Foveating Camera Theory and Design

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 2-2 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. 2-2. 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. 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,

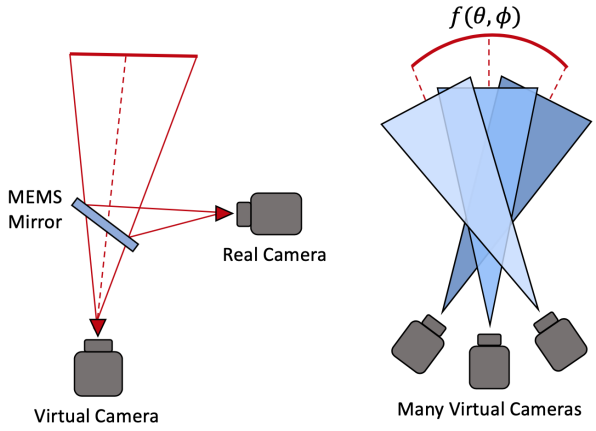


Figure 2-2. Moving mirror creates virtual views [115].

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

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.

The latest FoveCam device can be found in Figure 2-3. It includes a NVIDIA Jetson Nano for control, a Kurokesu 250mm L84 zoom lens, 6.4mm Mirrorcle MEMS Mirror, and FLIR Blackfly S 16S2C Board Level Camera. We show recent results from this prototype in Figure 2-1 at 1000 meters away.

2.2.1 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

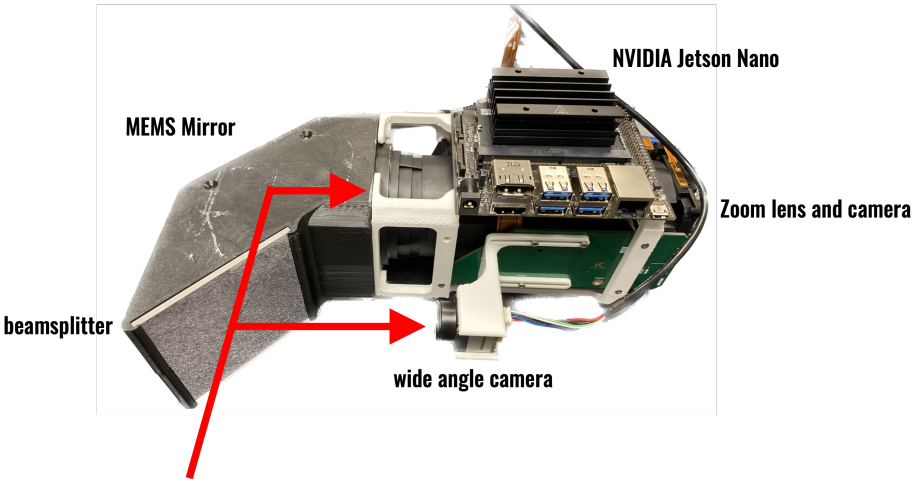


Figure 2-3. FoveaCam hardware prototype [115].

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 2-1 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.

2.2.2 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,

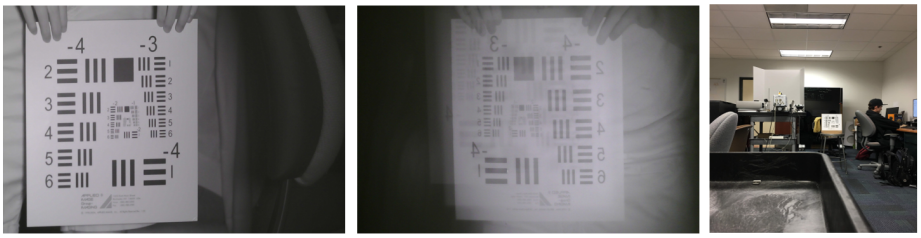
Table 2-1. Model Field of View Error (%).

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

we obtained a custom print of this standard 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 2-4 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.



without mirror with mirror iPhone 6s Plus

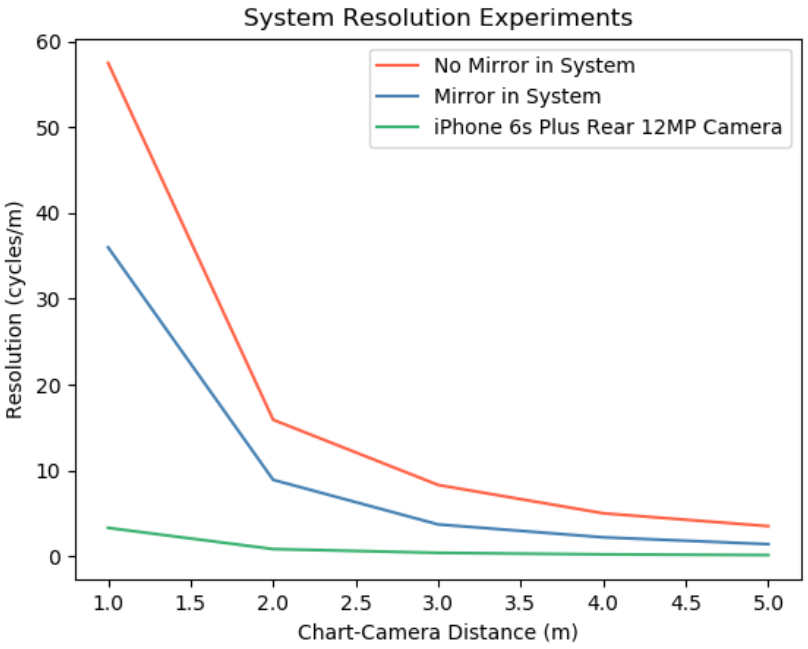


Figure 2-4. Resolution experiments [115].

2.2.3 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. 2-2, 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 FOV ω_{mirror}.

2.2.3.1 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 $\mathbf{U}$. *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

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

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 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 $5ms$ 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 Figure 2-5, 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 $80Hz$ with the camera imaging 640x480 at 8-bit RGB. We use 640x480 resolution since real time imaging is infeasible at the native 1920x1080. See Figure 2-5 for a visualization of our cameras timing.

Real-time demonstration. Figure 2-1 shows a real-time demonstration of our foveating camera using the point-to-point control algorithm for tracking four regions outside at 1000 meters at 20Hz for each region of interest.

Lissajous pattern. Our main control contribution is providing closed form differential

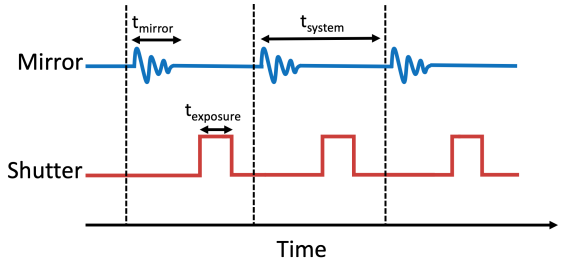


Figure 2-5. Timing description of our foveating camera [115].

updates to MEMS mirror 1D lissajous scanning motion. However, we only use lissajous scanning for the proof of concept control experiment in Section 2.3.5, 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 latency due to softer control on mirror coordinates. 1D lissajous scan patterns are realized by,

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

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 Figure 2-5 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.

2.2.3.2 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. Proability distributions can be obtained from a co-located sensor instead of the Kalman filter. Our contribution mainly lies in the subsequent sections Sect. 2.2.3.4 and Sect. 2.2.3.6, 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 [129].

2.2.3.3 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 2.3 we have used a vision-based face-tracker as a proxy for this filter. We define a control matrix $B_r(k)$ to update the state vector using the control vector $u_r(k)$:

$$q_r(k+1) = \mathbf{I_3} q_r(k) + B_r(k) u_r(k) + Q_r, \tag{2-4}$$

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

$$q(k+1) = F\ q(k) + B(k)\ u(k) + w, \tag{2-5}$$

where F is a target motion matrix, based on optical flow equations, 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

$$P(k+1) = [B_r(k)^T P_r B_r(k)\ 0;0\ P_t] + [Q_r(k)\ 0;0\ Q_t(k)], \tag{2-6}$$

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

$$K^{'} = P(k+1)(P(k+1)+R(k+1))^{-1}. \tag{2-7}$$

$$q^{'}(k+1) = q(k+1) + K^{'}(z(k+1) - q(k+1)). \tag{2-8}$$

$$P^{'}(k+1) = P(k+1) - K^{'}P(k+1). \tag{2-9}$$

2.2.3.4 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.

$$M_k = P_d \mathbf{E}[e^{'}(k+1)^T \; e^{'}(k+1)] + (1-P_d)\mathbf{E}[e(k+1)^T \; e(k+1)]. \tag{2-10}$$

where

$$e^{'}(k+1) = q(k) - q^{'}(k+1). \tag{2-11}$$

$$e(k+1) = q(k) - q(k+1). \tag{2-12}$$

$$\tag{2-13}$$

Using the trace trick, similar to [129], we can convert M_k into an expression using the covariance matrices,

$$M_k = tr[P(k+1)] - P_d(tr[P(k+1)] - tr[P^{'}(k+1)]). \tag{2-14}$$

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.

2.2.3.5 Updating the Control Vector

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

$$max_{\mathbf{u_r}} P_d \tag{2-15}$$

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,

$$P_d(q_r(k)) = \int_{s=0}^{L} f_t(x_r(k) + s\cos\alpha_r(k), y_r(k) + s\sin\alpha_r(k))ds \tag{2-16}$$

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, 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 $f_l(x,y)$

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

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

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

2.2.3.6 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 [131] 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 2-7I(c). 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 2-7II(c). In other words, the domain, within the Radan transform of bi-variate Gaussians, where we must search for the maximal slice, is bounded.

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. 2-6, 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.

2.2.3.7 Gradient Descent

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

Require: Kalman filter outputs, valid space **U**, epsilon error threshold ε, learning rate η, and initial control vector u_r
Ensure: Updated control vector u_r
  while True **do**
    $u_r^{tmp} \leftarrow u_r + \eta \frac{\delta P_d(q_r(k+1))}{\delta \mathbf{u_r}}$
    if $u_r^{tmp} \notin \mathbf{U}$ **then**
      return
    else if $\|u_r^{tmp} - u_r\| < \varepsilon$ **then**
      return
    else
      $u_r \leftarrow u_r^{tmp}$
    end if
  end while
  return u_r

Object 2-1. Gradient-based update of control vector u_r

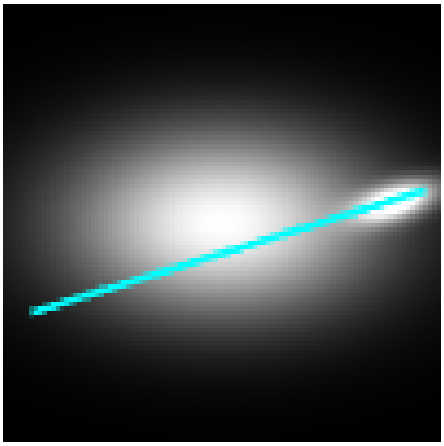


Figure 2-6. Heterogeneity [115].

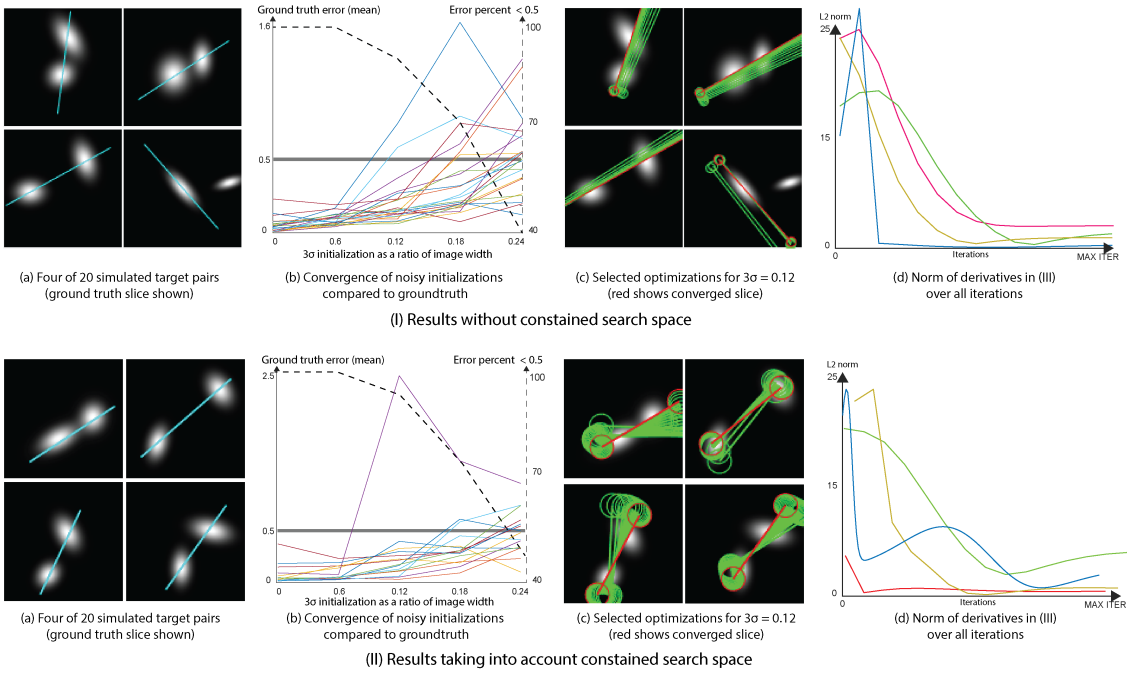


Figure 2-7. Simulations of 1D slice optimization [115].

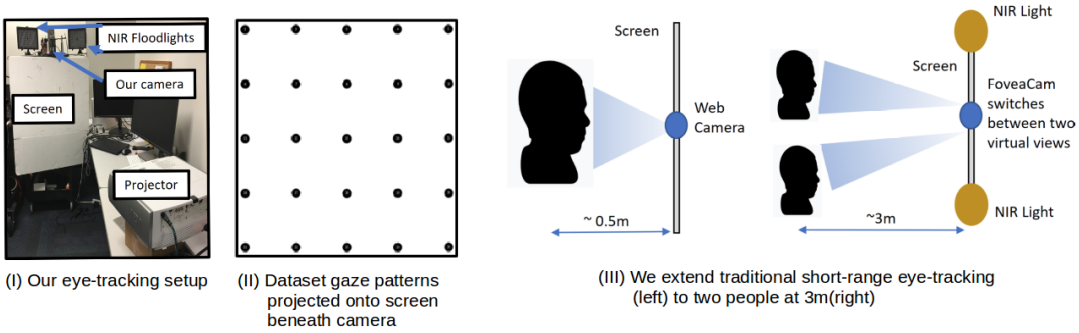


Figure 2-8. Our eye tracking setup and gaze pattern used for our finetuning dataset [115].

$$\frac{\delta P_d(q_r(k+1))}{\delta \mathbf{u_r}} = \frac{\delta P_d(q_r(k+1))}{\delta q_r(k+1)} \frac{\delta q_r(k+1)}{\delta \mathbf{u_r}} \tag{2-19}$$

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

$$\frac{\delta P_d(q_r(k+1))}{\delta q_r(k+1)} = \begin{bmatrix} \frac{\delta}{\delta x_r} P_d(q_r(k+1)) \\ \frac{\delta}{\delta y_r} P_d(q_r(k+1)) \\ \frac{\delta}{\delta \alpha_r} P_d(q_r(k+1)) \end{bmatrix} \tag{2-20}$$

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

$$\frac{\delta P_d(q_r(k+1))}{\delta q_r(k+1)} = \begin{bmatrix} \frac{\delta}{\delta x_r}\int_{s=0}^{L} f_l(x,y)ds \\ \frac{\delta}{\delta y_r}\int_{s=0}^{L} f_l(x,y)ds \\ \frac{\delta}{\delta \alpha_r}\int_{s=0}^{L} f_l(x,y)ds \end{bmatrix} + \begin{bmatrix} \frac{\delta}{\delta x_r}\int_{s=0}^{L} f_r(x,y)ds \\ \frac{\delta}{\delta y_r}\int_{s=0}^{L} f_r(x,y)ds \\ \frac{\delta}{\delta \alpha_r}\int_{s=0}^{L} f_r(x,y)ds \end{bmatrix} \tag{2-21}$$

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 Algorithm 2-1 we use these gradients to update the control vector.

Simulations: In Fig 2-7 we show simulations of Algorithm 2-1 on 20 pairs of 2D Gaussians. In Fig 2-7I(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. 2-7I(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. 2-7II 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. 2-7II(b).

Fig. 2-7I-II(b) 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. 2-7I-II(c) we show the convergence path for these examples, and in Fig. 2-7I-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 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.

2.3 Experiments

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 [68] 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.

2.3.1 Our Eye Tracking Setup

Our setup, shown in Fig. 2-8, 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 5x5 grid of points spanning the width and height of the lambertian plane, 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. 2.3.5, we describe how to control the movement of the mirror due to subject motion, but in this section we will assume that

Figure 2-9. Sample images from our eye tracking dataset [115].

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.

2.3.2 Finetuning a Gaze Tracking Network

The iTracker convolutional neural network [68] 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% of the iTracker imagery is iPhone data trained on eye angles varying in y from 2cm (4.5^o) to 10cm (21.8^o) and x from -1cm (2.3^o) to 5cm (13.5^o). To maintain these angles at 3m for our data, we trained on patterns spanning x from -39cm to 39cm (7.4^o) and y from -21cm (4^o) to -82cm (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, $25cm$ in iTracker vs. $3m$ 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

Table 2-2. 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

our setup. All our training and testing was done at 3m from the camera.

2.3.3 Data Collection for Finetuning

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^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. 2-9(I)) enabled the collection of fine-tuning data in-situ with the device, in NIR, for the grid pattern in Fig. 2-8 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 [68], 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.

2.3.4 Experimental Results

In our experiments, the subjects were at $3m$ distance and six people were involved overall, four for training and testing, two for the proof of concept experiment in Section 2.3.5. To show that this relatively small fine-tuning dataset does not adversely affect our results, we show, in Table 2-2, 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 1400 users encapsulated in the pre-trained weights.

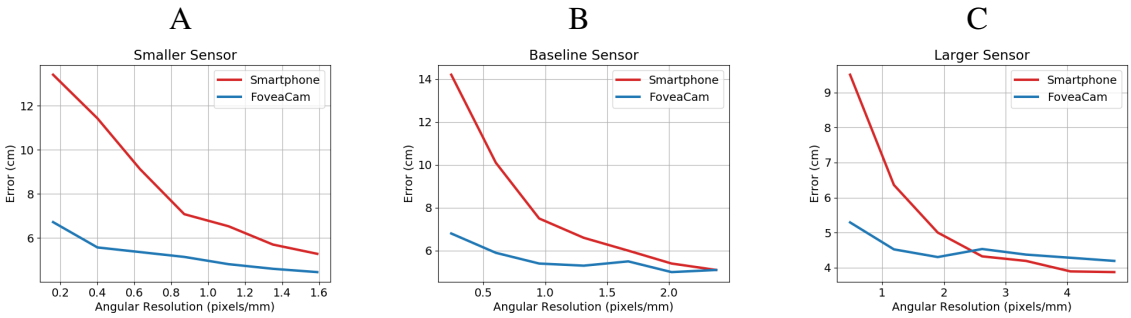


Figure 2-10. Examples of eye tracking error for various simulated sensors [114].

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 2.3.2.

Simulation Model. 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. Figure 2.3.4 demonstrates 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 Figures 2.3.4 A-C 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 Figure 2.3.4 C. 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 Figures 2-11 and 2-12 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 11.06cm. The camera parameters for this visualization include increasing sensor pixel sizes by 1.5, using 5mm and 2.5mm 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 Figure 2.3.4 A. Since our native FOV was 466mm x 262mm (W x H), the image size at these parameters was 332 x 104, this clearly shows the benefit of our camera: we are able to sacrifice significant resolution and maintain high performance.

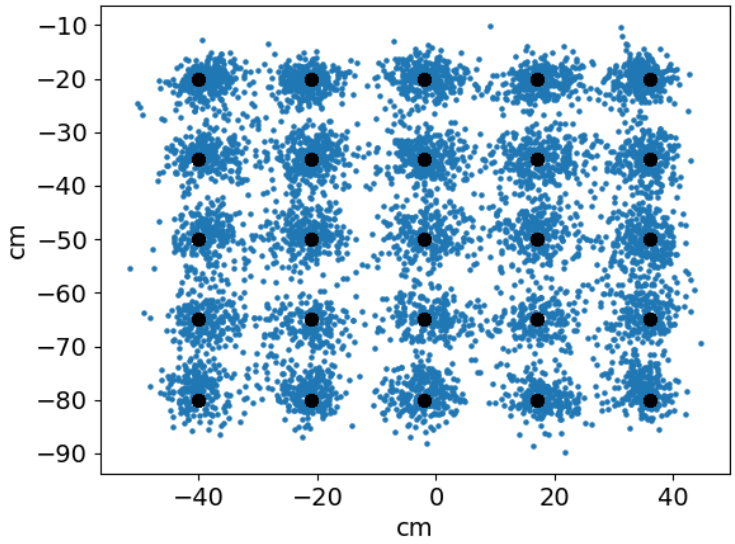


Figure 2-11. Raw network output for our foveating camera test data with 5.15cm error [114].

2.3.5 Proof-of-concept Control Experiment

Finally, we use the control from Section 2.2.3, 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,

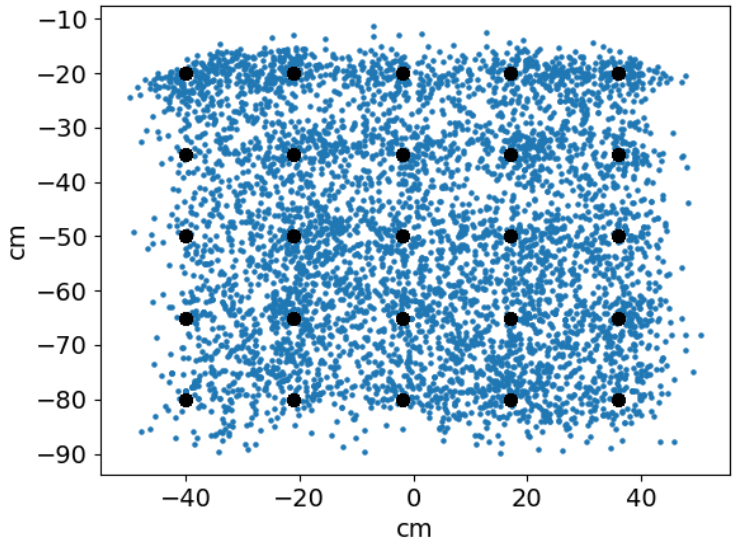


Figure 2-12. Raw network output for smartphone test data with 11.06cm error [114].

or testing the network are looking at a square pattern the network has not seen. This pattern has a smaller span than the 5 x 5 grid used in Section 2.3.

We use a the bounding box from a simple face-tracker [60] 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. 2-13, 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 (6.8cm and 6.03cm L2 error respectively).

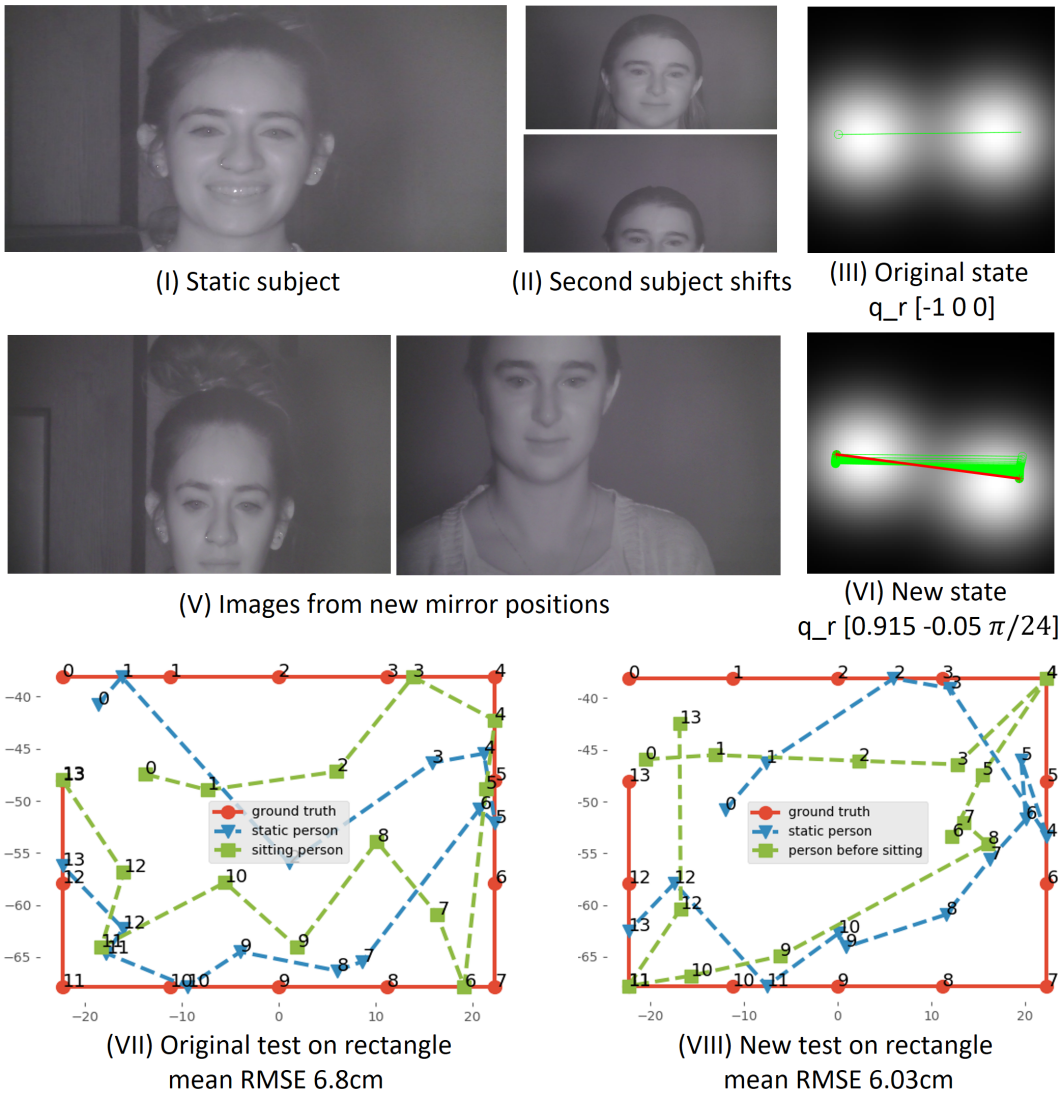


Figure 2-13. Proof-of-concept control experiments [115].

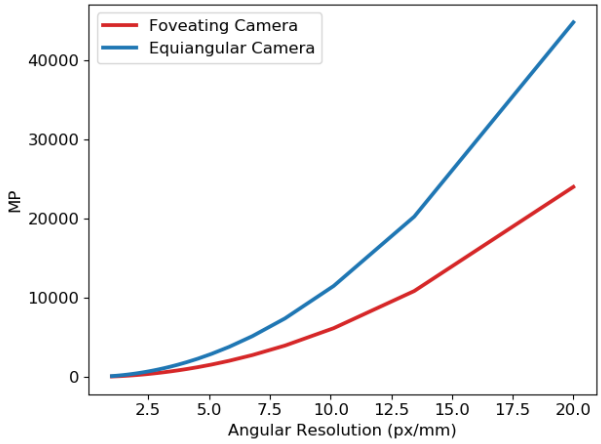


Figure 2-14. Foveating camera advantages versus gigapixel cameras [114].

2.4 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.

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 x 10cm x 10cm, we acknowledge this size will need to be reduced before foveating cameras could easily be integrated into robotic imaging systems.

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 Figure 2-14.

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.

CHAPTER 3
ACTIVE FOVEATED IMAGING

Active 3D depth sensors have diverse applications in augmented reality, navigation, and robotics. Recently, these sensor modules are widely used in consumer products, such as time-of-flight (Lidar[51]), structured light (Kinect V1 [56]) and others. In addition, many computer vision algorithms have been proposed to process the acquired data for downstream tasks such as 3D semantic understanding [112], object tracking [53], guided upsampling in SLAM [95], among other applications.

Unlike stereo cameras that only sense reflected ambient light passively, active depth sensors illuminate the scene with modulated light patterns, either spatially, temporally, or both. The illumination encodings allow robust estimation of scene depths. However, this also leads to three shortcomings: First, active depth sensors consume optical power, burdening wearable devices that are on a tight power budget. Second, the number of received photons reflected back from the scene drops with inverse-square relationship to scene depth. The maximum sensing distance is thereby limited by the received signal-to-noise ratio (SNR). Third, strong, active light sources on the device may unintentionally hurt the user or other people around. For consumer devices, this constraint can be as strict as ensuring safety when a baby accidentally stares at the light source directly. Interestingly, these three factors are often entangled with each other. For example, naively increasing range by raising optical power makes the device less eye-safe. An active 3d sensor would benefit from the joint optimization of these three goals, as illustrated in Fig. 3-1(a).

In this paper, we present an adaptive depth-sensing strategy. Our key idea is that the coded scene illumination need not be sent to the entire scene (Fig. 3-1(b)). Intuitively, by limiting the illumination samples, the optical power per sample is increased, therefore extending the maximum sensing distance. This idea of *adaptive sensing* is supported by three observations: First, illumination samples only need to be sent to parts of a scene where passive depth estimation fails (due to lack of texture). Second, depth estimation is often application-driven, accurate depths are only needed around AR objects to be inserted into the scene. Finally, for video applications, sending light to regions where depths are already available from previous frames is redundant.

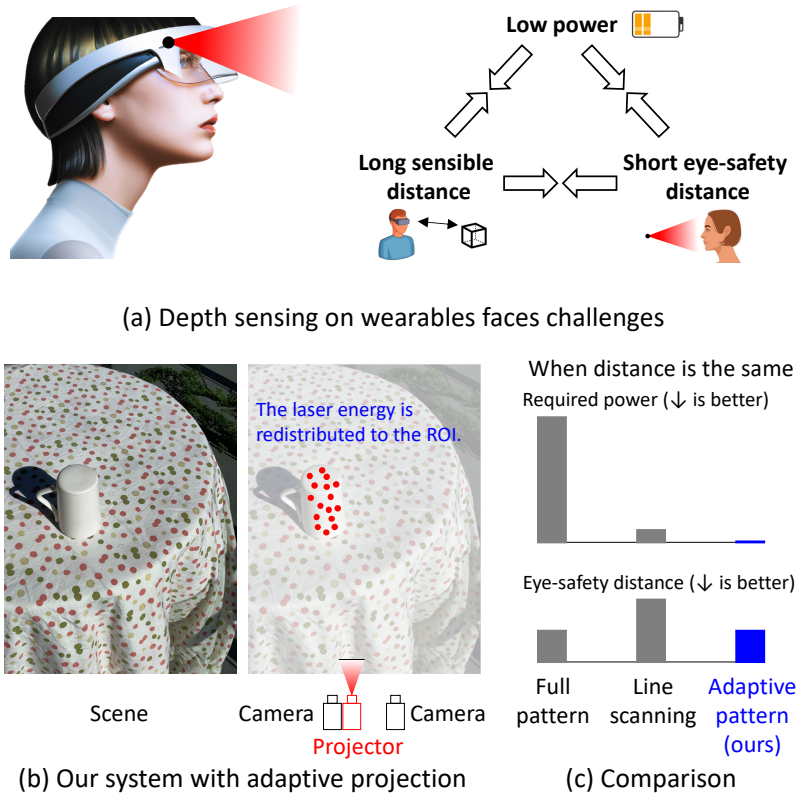


Figure 3-1. Active foveated imaging for energy-efficient adaptive 3D sensing [117].

Based on these observations, we demonstrate this adaptive idea with a stereo-projector setup (*i.e.*, active stereo [134, 27, 5]), where an attention map is computed from the camera images for efficient light redistribution.

To quantitatively understand the benefits of our approach, we propose a sensor model that analytically characterizes various sensing strategies, including full-frame (RealSense [64]), line-scanning (Episcan3D [90]), point-scanning (Lidar [96]) and proposed adaptive sensing. We establish, for the first time, a framework that jointly analyzes the power, range, and eye-safety of different strategies and demonstrates that, for the same maximum sensing distance, adaptive sensing consumes the least power while achieving the shortest (best) eye-safety distance.

Note that the realization of scene-adaptive illumination is not trivial: Common off-the-shelf projectors simply block part of the incident light, which wastes optical power. We propose two hardware implementations for adaptive illumination: One is inspired by digital holography, which uses Liquid Crystal on Silicon (LCoS) Spatial Light Modulators (SLM) to achieve free-form light

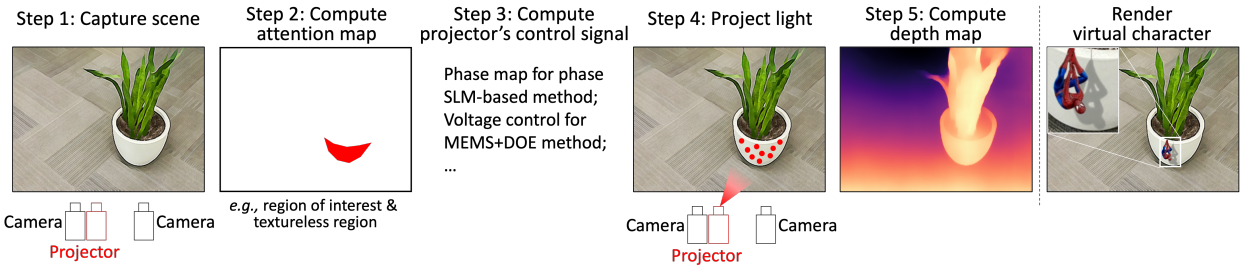


Figure 3-2. Our method overview [117].

projection. The other implementation uses diffractive optical elements (DOE) to generate dot patterns in a local region of interest (ROI), which is directed to different portions of the scene by a micro-electro-mechanical (MEMS) mirror.

Our contributions are summarized as follows:

1. We propose adaptive 3D sensing and demonstrate its advantage in a theoretical framework that jointly considers range, power and eye-safety.

2. We implement the proposed adaptive active stereo approach with two hardware prototypes based on SLM and MEMS + DOE.

3. Real-world experimental results validate that our sensor can adapt to the scene and outperform existing sensing strategies.

3.1 Related Work

Active 3D sensing with ambient light noise. Various techniques have been proposed to address photon noise due to strong ambient light (sunlight), such as choosing a wavelength where sunlight is weak [92, 127], using a polarizing filter [92]. Gupta [46] uses a theoretical model to show that instead of illuminating the full scene, concentrating light on different parts of a scene sequentially improves SNR for structured light, which is demonstrated with a rotating polygonal mirror. Based on similar principles, MC3D [79] uses a MEMS/galvo-driven laser and an event camera to achieve bandwidth-efficient scanning. Episcan3D [90] and EpiToF [1] use a line-scanning laser and a synchronized rolling-shutter camera to achieve fast and efficient depth sensing. This paper further extends this line of work by showing that, with the freedom to

adaptively illuminate part of the scene, a lower power budget is needed to achieve the same sensing range while being safer to the eyes.

Adaptive 3D sensing. Ideas from visual attention [33, 12] have influenced vision and robotics. Efficient estimation algorithms have been shown for adaptive sensing and point-zoom-tilt (PZT) cameras [13, 130]. In the 3D sensor space, 3D light-curtains [126, 7, 15] represent a flexible system where curtains can be adaptively placed in the scene for robotics, and other applications [97, 4, 3]. Full control of the MEMS mirror enables adaptive sampling for LIDAR [111, 96, 109] and adaptive passive camera resolution for monocular depth sensing [116]. While previous adaptive systems focus on different aspects such as flexibility, frame rate, among others, this work studies the interplay between range, power, and eye-safety.

3.2 Signal to Noise Ratio Analysis

The workflow of the proposed adaptive 3D sensing is shown in Fig. 3-2. We use an active stereo design with two cameras and a projector. The device first captures an image of the scene and computes an attention map to determine the region of interest (ROI). Hardware-specific control signals are computed from the attention map such that the projector redistributes the light to the ROI. Finally, a high-quality depth map can be calculated from captured stereo images.

Before getting into details on how to implement this adaptive illumination in practice, let us assume an ideal *flexible* projector for a moment: If we can redistribute the optical power to an arbitrarily-shaped ROI, how well can it perform? We adopt a model to quantify its depth estimation performance and compare it to other existing or naively conceived active depth sensing strategies.

3.2.1 Sensor Model and SNR Analysis

The accuracy of various active depth sensors, including structured light, active stereo, continuous-wave time-of-flight (CW-ToF), among others, can be quantified by a single metric: the SNR of the measured light signal (projected by the active illumination source and reflected by the scene). The noise consists of the photon noise from both the signal itself and the ambient light,

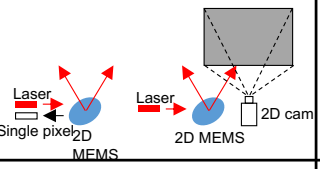
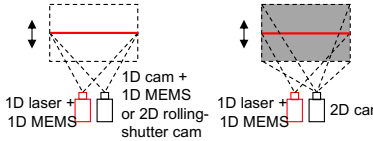
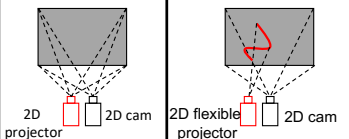
	Point scanning		Line scanning		(e) Full-frame pattern	(f) Adaptive **(Proposed)**
	(a) V1: Synced	(b) V2: Unsynced	(c) V1: Synced	(d) V2: Unsynced		
*Typically, $N = 10^2 \sim 10^3$						
Illuminated Area Divisor (R_a)	N^2	N^2	N	N	1	N
Laser Exposure Divisor (R_{t1})	N^2	N^2	N	N	1	1
Camera Exposure Divisor (R_{t2})	N^2	1	N	1	1	1
Same power, sensing at same distance:						
SNR $= c\frac{R_a\sqrt{R_{t2}}}{R_{t1}}$	$\boldsymbol{cN}$	c	$c\sqrt{N}$	c	c	$\boldsymbol{cN}$
Same maximum sensing distance d_{max}:						
Power P	$\boldsymbol{k_p d_{max}^2 N^{-1}}$	$k_p d_{max}^2$	$k_p d_{max}^2 N^{-0.5}$	$k_p d_{max}^2$	$k_p d_{max}^2$	$\boldsymbol{k_p d_{max}^2 N^{-1}}$
Eye-safety distance l_{min}	$k_{ld} d_{max} N^{0.25}$	$k_{ld} d_{max} N^{0.75}$	$k_{ld} d_{max} N^{0.125}$	$k_{ld} d_{max} N^{0.375}$	$\boldsymbol{k_{ld} d_{max}}$	$\boldsymbol{k_{ld} d_{max}}$

Figure 3-3. Schematic diagrams and analysis of various 3D sensing strategies [117].

and the sensor readout noise, mathematically defined as follows [46, 47]:

$$\begin{aligned} SNR &= \frac{\text{Signal}_{\text{projector}}}{\sqrt{N^2_{\text{photon\_ambient}} + N^2_{\text{photon\_projector}} + N^2_{\text{read}}}} \\ &= \frac{\frac{P}{d^2 a} t_1}{\sqrt{P_{\text{sun}} t_2 + \frac{P}{d^2 a} t_1 + N^2_{\text{read}}}} \end{aligned} \tag{3-1}$$

where P is the optical power of the projector (assuming an albedo of 1 in the scene), a is the illuminated area at unit distance, d is the distance of the scene (thus the inverse-square fall-off), P_{sun} is the optical power of the ambient light, t_1 and t_2 are the duration when the laser is on and the camera is active, respectively, and N_{read} is the standard deviation of the read noise. Ambient light-induced photon noise dominates in outdoor scenarios and also indoors with power-limited devices, which is the major focus of the following analysis. In these situations, SNR can be simplified as:

$$SNR \approx \frac{\frac{P}{d^2 a} t_1}{\sqrt{P_{\text{sun}} t_2}}. \tag{3-2}$$

When readout noise dominates, which happens in a dark room or at night, SNR can be simplified as this

$$SNR \approx \frac{\frac{P}{d^2 a} t_1}{N_{\text{read}}}. \tag{3-3}$$

Analyzing different sensing strategies. We use this SNR model to compare the performance of different depth sensors. For a fair comparison, we assume all depth sensors have equal total optical power P, sensing at same depth d. Their performance is then uniquely determined by t_1, t_2 and a. For off-the-shelf *full-frame* projectors (Fig. 3-3(e)), we denote $a = A$ which corresponds to the entire FOV, and $t_1 = t_2 = T$ as both the sensor and the projector are active during the entire camera exposure T.

Previous work [46, 79, 90, 1] has shown that, instead of flood-illuminating the entire scene, focusing optical power on different parts of the scene sequentially can lead to higher SNR. To quantitatively analyze this effect, we represent the illuminated area, laser exposure and camera exposure as a division of the full-frame case:

$$a = A/R_a, \qquad t_1 = T/R_{t1}, \qquad t_2 = T/R_{t2}, \tag{3-4}$$

where R_a, R_{t1}, R_{t2} are defined as illuminated area divisor, laser exposure divisor, camera exposure divisor, respectively. SNR is then a function of these divisors:

$$SNR = \frac{\frac{P}{d^2 A/R_a} T/R_{t1}}{\sqrt{P_{\text{sun}} T/R_{t2}}} = \underbrace{\frac{P\sqrt{T}}{d^2 A \sqrt{P_{sun}}}}_{c} \underbrace{\frac{R_a \sqrt{R_{t2}}}{R_{t1}}}_{X}, \tag{3-5}$$

where c is the SNR of full-frame projection. X is a factor that describes how each method compares with the baseline full-frame projection. It is difficult to optimize X directly since not every combination of (R_a, R_{t1}, R_{t2}) is feasible in hardware. Nevertheless, it provides a useful tool to characterize different sensing strategies.

State-of-the-art systems such as Episcan3D [90] implement this idea as a *line scanning* scheme, as shown in Fig. 3-3(c). If we assume the total illuminated region of line scanning is the

same as full-pattern, then $R_a = R_{t1} = R_{t2} = N$, where N is the number of scanlines (typically $10^2 - 10^3$). By plugging these terms into Eq. 3-5, the SNR of line scanning is $X = \sqrt{N}$ times higher than the full-pattern.

One interesting question is: Can we push this idea further and scan a dot at a time? This *point scanning* idea can be implemented with a co-located laser and single-pixel sensor deflected by a 2D MEMS mirror. Using the same assumption, $R_a = R_{t1} = R_{t2} = N_p$, where N_p is the number of dots (typically $N_p = N^2 = 10^4 - 10^6$). Fig. 3-3(a) shows that dot scanning does offer a higher SNR and is $X = N$ times higher than the full-pattern. Notice that this SNR benefit comes from the fact that the laser and the sensor are *synchronized*: The sensor only receives light from the area illuminated by the laser at any instant. For their unsynchronized counterparts where the sensor is a 2D camera that captures the entire 2D FOV during the whole imaging time (easier to implement in hardware), their SNR is exactly the same as the full-pattern approach (Fig. 3-3(b,d)).

textbfAdaptive sensing. Our *adaptive* sensor projects a static pattern that does not change during the entire exposure, $R_{t1} = R_{t2} = 1$. However, the optical power is concentrated to a small ROI, which we assume can be as small as one line in the line-scanning approach $R_a = N$. As shown in Fig.3-3(f), our adaptive sensor has a N_l times higher SNR than the full-pattern approach. In summary, we observed that $SNR_{\text{adaptive}} = SNR_{\text{point}} \gg SNR_{\text{line}} \gg SNR_{\text{full}}$.

3.2.2 Comparison of Power, Range, and Eye-Safety

Sec. 3.2.1 analyzes the SNR for different sensors at the same depth. However, this analysis is insufficient, since increasing SNR and the maximum range implies a higher risk of eye injury. In this section, we discuss how this model can be extended to analyze the trade-off between power, range and eye-safety. We consider two key constraints: maximum sensing distance and minimum eye-safety distance.

Maximum sensing distance. We assume that for reliable estimation of the depth, the SNR must be greater than a minimum detection threshold $\text{SNR}_{\text{thres}}$. The equality holds when the maximum sensing distance $d = d_{max}$ is reached,

$$SNR_{\text{thres}} = \frac{P\sqrt{T}}{d_{\max}^2 A\sqrt{P_{sun}}} X. \tag{3-6}$$

Rearranging this equation gives

$$\boxed{P = k_p \cdot d_{\max}^2 X^{-1} = k_p \cdot d_{\max}^2 R_a^{-1} R_{t1} R_{t2}^{-0.5},} \tag{3-7}$$

where k_p is a method-independent constant.

Minimum eye-safety distance. A minimum eye-safety distance can be defined when the maximum permissible exposure (MPE, defined in ANSI Z136) is reached:

$$\frac{P}{l_{\min}^2 a} = \frac{MPE(t_1)}{t_1}, \tag{3-8}$$

It is considered dangerous for eyes to be exposed at a distance shorter than l_{min}. Intuitively, the shorter the minimal eye-safety distance is, the more eye-safe the device is. We expand MPE based on definitions from ANSI Z136:

$$\frac{P}{l_{\min}^2 a} = \frac{MPE(t_1)}{t_1} = \frac{C_\lambda t_1^{0.75} 10^{-3}\ (\text{J}\cdot\text{cm}^{-2})}{t_1} = k_e t_1^{-0.25}, \tag{3-9}$$

where k_e is a method-independent constant. Plug in Eq. 3-4 and rearrange,

$$l_{\min} = k_l \cdot P^{0.5} R_a^{0.5} R_{t1}^{-0.125}, \tag{3-10}$$

where k_l is a method-independent constant.

Comparing different sensors. From Eq. 3-7 and Eq. 3-10, it is clear that for a depth sensing method, specifying one quantity among P, $d_{\max}$ and l_{min} will also determine the other two. We thus focus on the following question: To reach the same maximum sensing distance $d_{\max}$, what is the power consumption P and eye-safety distance l_{min} of each method? This is a key problem for consumer devices with limited power budget. Plug Eq. 3-7 into Eq. 3-10 and

rearrange:

$$\boxed{l_{\min} = k_{ld} \cdot d_{\max} R_{t1}^{0.375} R_{t2}^{-0.25},} \tag{3-11}$$

where k_{ld} is a method-independent constant.

Fig. 3-3 summarizes the results derived from Eq. 3-7 and Eq. 3-11 for different sensing methods. Full-frame pattern method is the most eye-safe but consumes the most power. Conversely, point scanning (synced) consumes the least power but is also the least eye-safe, which highly limits its application in consumer devices (laser projectors). Line scanning (synced) strikes the sweet middle ground, which extends the distance by a large margin while maintaining eye safety. Finally, by concentrating to a small ROI, the proposed adaptive method consumes the least power and achieves the best eye-safety.

To intuitively showcase this advantage, we assume $N \sim 100$ to 500, which is consistent with the spatial resolution of most concurrent 3D sensors. For high-resolution depth sensors with $N > 1000$, the gain is even greater. At the same maximum sensing distance, adaptive sensing:

1. has the same eye-safety distance as full-frame sensors, while consuming N^{-1} (0.01 to 0.002)$\times$ lower power.

2. has $N^{-0.125}$ (0.56 to 0.46)$\times$ shorter (better) eye-safety distance as line-scanning, while consuming $N^{-0.5}$ (0.1 to 0.04)$\times$ lower power.

It is important to mention that these calculations are based on the assumption that the illuminated area for adaptive sensing is the same as line scanning: $R_a = N$. In practice, this area may be larger depending on the scene and application. The adaptive projector (SLM) may also have a limited light efficiency, which gives an effectively smaller R_a and thus lower SNR. Nonetheless, at the same maximum distance, adaptive sensing still has a power benefit as long as $R_a > \sqrt{N}$, and it always has a eye-safety benefit since l_{min} is independent of R_a. The theoretical analysis forms the foundation for the proposed adaptive 3D sensing. We validate the analysis in a real-world prototype in Sec. 3.4.

Disjoint ROIs. So far, we assume an ideal flexible projector which can project light to

Table 3-1. Variations of adaptive sensing.

	SNR	P	l_{min}
V1	cN	$k_p d_{max}^2 N^{-1}$	$k_{ld} d_{max}$
V1-a	cNK^{-1}	$k_p d_{max}^2 N^{-1} K$	$k_{ld} d_{max} K^{0.375}$
V1-b	$cNK^{-0.5}$	$k_p d_{max}^2 N^{-1} K^{0.5}$	$k_{ld} d_{max} K^{0.125}$

arbitrarily-shaped, even disjoint ROIs simultaneously. In practice, certain hardware implementations do not have this capability (an example is discussed in Sec. 3.3.2). To this end, we propose a more flexible scanning strategy: During the camera exposure, the system scans K disjoint ROIs sequentially (typically $2 \leq K \leq 5$). This adaptive V1-a method consumes slightly more power and has a slightly longer eye-safety distance (Tab. 3-1). Another option is to divide the camera exposure into K shorter exposures, and the system scans a single ROI during each exposure. This adaptive V1-b method performs comparably as V1, but requires a K times higher camera frame rate.

3.3 Implementation of Adaptive Illumination

Now that we have theoretically analyzed the benefit of the proposed adaptive illumination, how can the proposed adaptive illumination be realized? Notice that this is not a trivial problem. The hardware implementation must satisfy two criteria: (1) The system can redistribute the optical power to a small ROI (guided by an attention map), and (2) This ROI can be projected to different parts of the scene flexibly and in real-time (30Hz). A common LCD or DLP projector satisfies (2) but does not satisfy (1). In this section, we propose two hardware configurations that satisfy both conditions.

3.3.1 Implementation 1: Phase SLM

Fig. 3-4(a) shows our SLM-based implementation. Our holographic projection approach is inspired by recent work on holographic near-eye 3D displays [76]. Specifically, a hologram to be reproduced by the SLM is decomposed as a sum of *sub-holograms*, where each sub-hologram diffracts light to a single object point in the scene. In [76], each sub-hologram is created using a *lens phase function*:

$$f_n^{\text{lens}}(\mathbf{X}) = e^{j2\pi\sqrt{(X-x_n)^2+(Y-y_n)^2+(Z-z_n)^2)}/\lambda}, \tag{3-12}$$

where (X,Y,Z) is the 3D position of each pixel in the sub-hologram, (x_n,y_n,z_n) is the 3D position of the n-th object point, and λ is the light wavelength. The full hologram is,

$$H(\mathbf{X}) = \sum_{n=1}^{N} f_n^{\text{lens}}. \tag{3-13}$$

This lens phase function mimics a lens that focuses light to the object point at the correct depth, which works well for near-eye displays. One limitation of this lens phase function approach is that light is only redistributed *locally*. This is because the SLM can only reproduce a smooth hologram due to the Nyquist frequency determined by the finite pixel pitch. However, f^{lens} varies rapidly for off-center pixels (X,Y far away from x_n,y_n), causing aliasing artifacts. Therefore, sub-holograms of much smaller sizes must be used, which greatly limits the light efficiency.

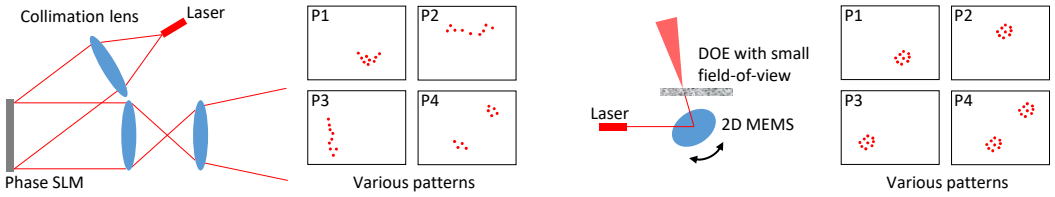


Figure 3-4. Hardware implementations ray diagrams [117].

To alleviate this limitation, we propose the use of *mirror phase function*:

$$f_n^{\text{mirror}}(\mathbf{X}) = e^{j(X \cdot x_n + Y \cdot y_n)}, \; H(\mathbf{X}) = \sum_{n=1}^{N} f_n^{\text{mirror}}. \tag{3-14}$$

The mirror phase function corresponds to a smooth phase map linear in terms of X, Y, and can be implemented on the SLM without aliasing. It allows us to use each sub-hologram as a mirror that reflects light to the right direction. By taking the sum of sub-holograms that reflect to different directions, desired projection patterns can be achieved.

Conversion to phase-only holograms. Notice that Eq. 3-14 creates a hologram with both amplitudes and phases being spatially-variant, which cannot be implemented on a phase-only SLM. Several approaches [52, 38] have been proposed to convert such a full hologram to a phase-only hologram. Fortunately, our goal is not to project a high-quality image, and simple amplitude discarding is sufficient to project unique texture to the scene:

$$H_{phase} = \text{Arg}[H]. \tag{3-15}$$

Efficient implementation. The mirror phase function consists of simple arithmetic operations on large matrices, which can be implemented efficiently on a GPU. We implement our hologram generator in CUDA and render the resulting hologram phase from the framebuffer to the SLM using OpenGL-CUDA interoperability. On a NVIDIA Jetson Nano, an embedded system-on-module with a Tegra X1 Maxwell 256-core GPU and limited computing resources, we are able to generate 1080p holograms with around 100 points or less at 30 fps. Our

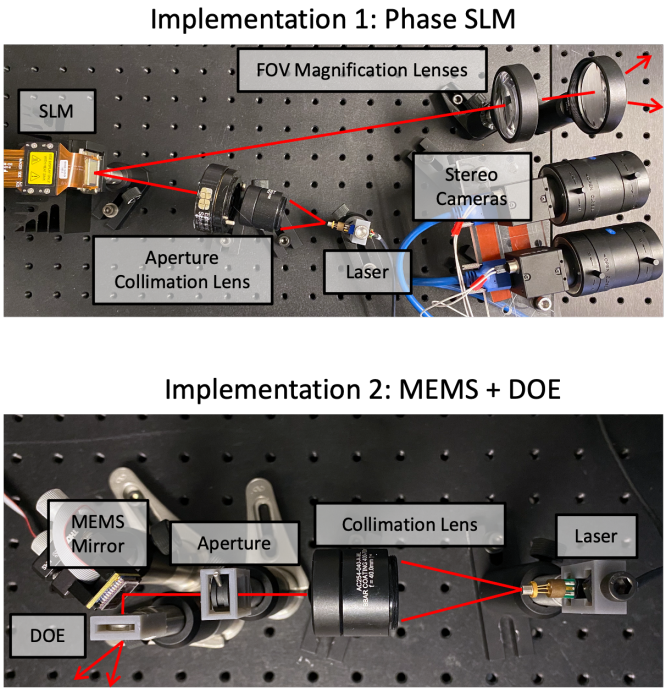


Figure 3-5. Hardware prototypes [117].

implementation and simulator can be found at https://github.com/btilmon/holoCuhttps://github.com/btilmon/holoCu.

3.3.2 Implementation 2: MEMS + DOE

Our second implementation is to adjust the beam incident angle of a diffractive optical element (DOE) with a MEMS mirror, as shown in Fig. 3-4(b). DOE offers a cheap, energy-efficient solution for random dot projection in single-shot structured light (Kinect V1) or active stereo (RealSense). While those systems use a DOE that covers the entire scene, we use a small FOV ($\approx 5^\circ$) that only corresponds to a small ROI. By rotating the MEMS mirror, the deflected laser beam hits the DOE at different angles, thus generating a dot pattern at different ROIs of the scene.

Comparison with phase SLM. The MEMS + DOE implementation is less flexible than the SLM implementation since the hologram shape is fixed (determined by the DOE phase pattern). This is schematically shown in Fig. 3-4: While SLM can illuminate ROIs of various shapes (P1-P3), MEMS + DOE can only create the same shape shifted across the scene. Moreover, while

the SLM can redistribute the optical power over two disjoint ROIs during the same camera exposure (P4), different ROIs are scanned and imaged sequentially by the MEMS mirror, which slightly decreases the SNR (see Sec. 3.2.2 for detailed analysis). Nevertheless, the MEMS + DOE approach benefits from low cost, simple optics and small form factor, which are important factors for mobile and wearable devices.

3.4 Experiments

Hardware prototypes. Fig. 3-5 shows both hardware prototypes of our proposed method. We use two FLIR BFS-U3-16S2C-CS cameras equipped with 20mm lenses as a stereo pair. Our SLM implementation uses a Holoeye GAEA LCoS (phase-only) SLM, which can display 4K phase maps at 30 frames per second. Our MEMS + DOE implementation uses a 0.8*mm* diameter bonded Mirrorcle MEMS Mirror. A random dot DOE with a small FOV is preferred. Here, we used a Holoeye DE-R 339 DOE that produces a periodic 6x6 dot pattern with 5° FOV instead and we tilt the DOE such that the pattern is still unique locally on the epipolar line.

3.4.1 Attention Map and Depth Estimation.

We adopt classical semi global block matching for depth estimation [30]. The attention map is determined by randomly choosing pixels that do not have a valid depth value from the depth map computed from passive images. In practice, the attention map can be conditioned by the application such that illumination is only needed within the regions where AR objects are inserted. Our goal is to present a general sensor that can fit into many different perception systems and improve active depth sensing.

3.4.2 Comparison Between 3D Sensing Strategies.

We emulate full-frame and line-scanning strategies on our SLM implementation and compare them with our adaptive sensing strategy. An example of each emulated sensor can be

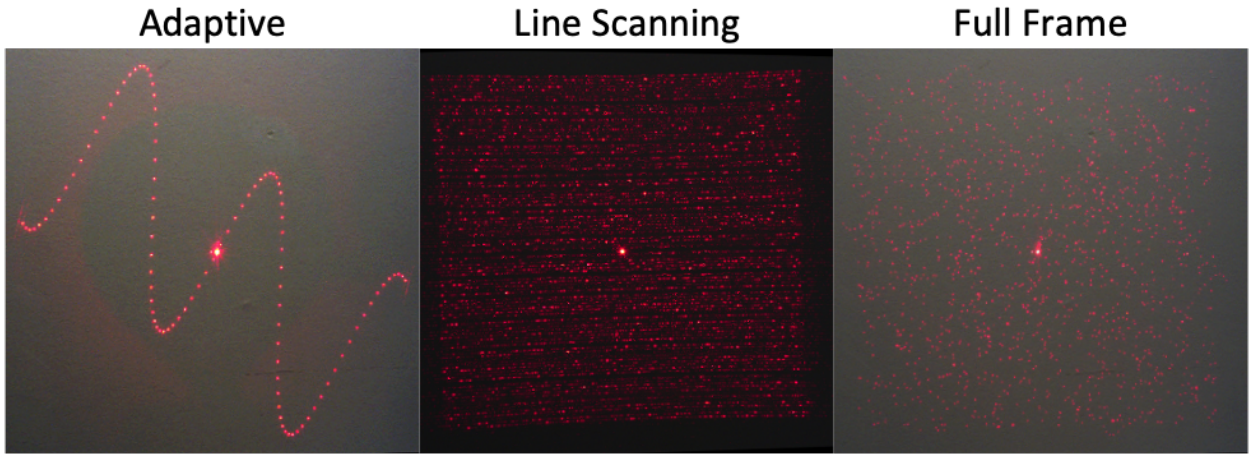


Figure 3-6. Emulating 3D sensors on a phase only spatial light modulator [117].

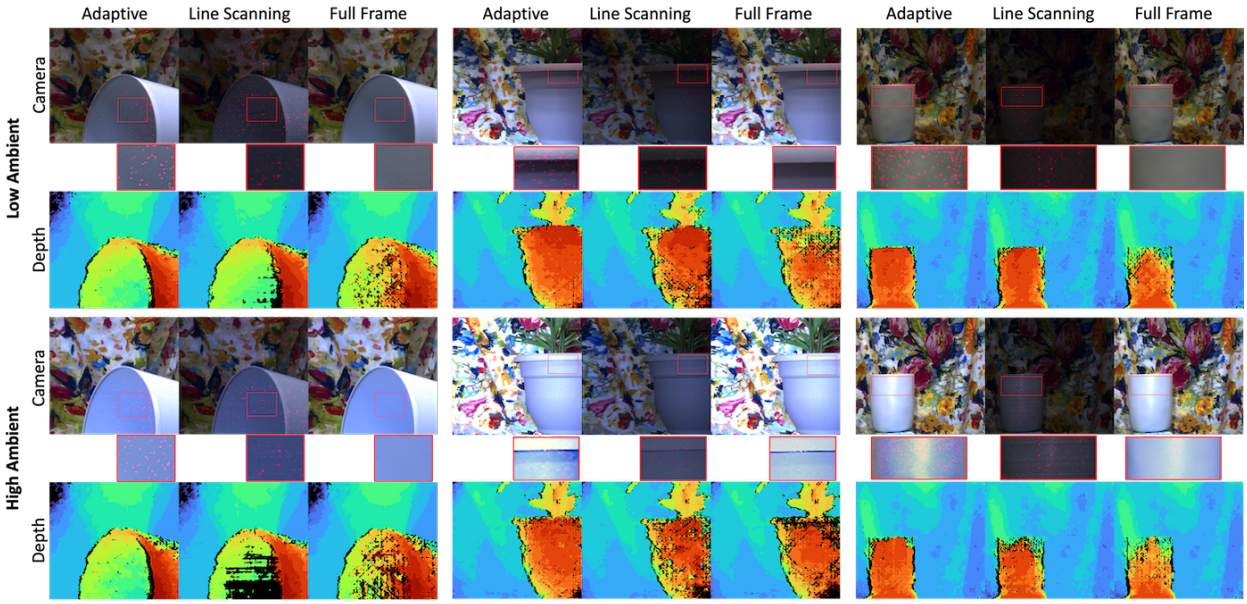


Figure 3-7. Comparison between 3D sensing strategies [117].

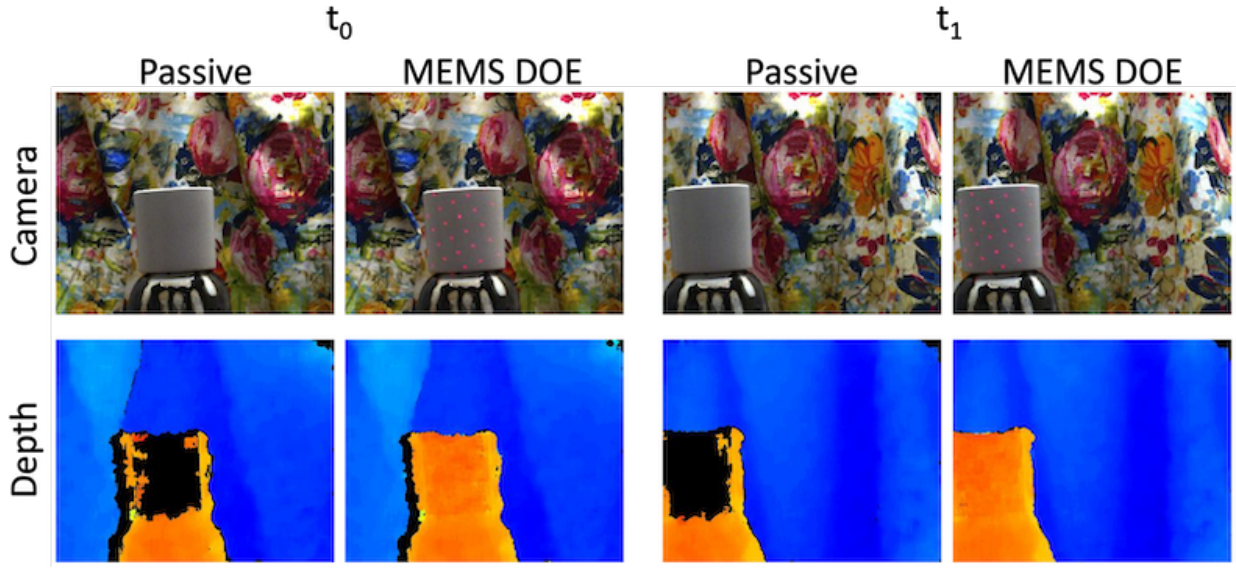


Figure 3-8. MEMS + DOE implementation [117].

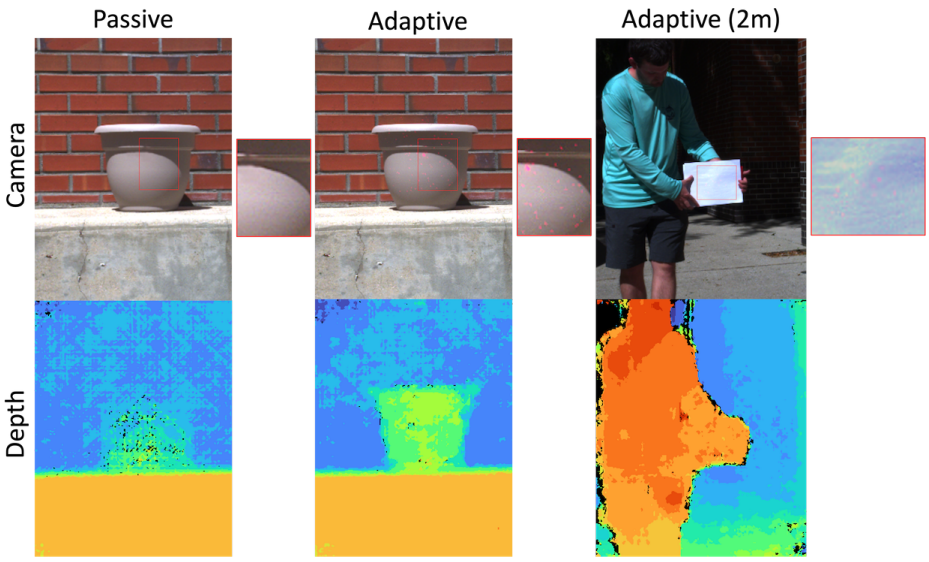


Figure 3-9. Outdoor 3D sensing with phase only spatial light modulator [117].

found in Fig. 3-6. For line scanning, we compute and project the hologram of the dot pattern line-by-line. We capture the image for each line individually and stitch the corresponding camera rows together into a single image.

Fig. 3-7 shows the results for three different scenes. All scenes are illuminated with the same ambient lighting and laser power. Laser power, exposure time and illuminated area are chosen to ensure fair comparison. Due to the dominating photon noise from the ambient light, full-frame and line scanning methods have a low SNR. As a result, depth estimation fails in the textureless regions. Since the proposed adaptive sensing technique concentrates light to the textureless regions, it achieves much higher SNR and obtains higher-quality depth maps, which validates our theoretical analysis.

3.4.3 Outdoor Depth Sensing Under Direct Sunlight.

Fig. 3-9 demonstrates our Phase SLM prototype working outdoors under 50 kilolux direct sunlight. We can rely on passive stereo to compute depth for the majority of the scene and only project light where necessary, such as the white textureless pot. We also show a distance test of the Phase SLM prototype at 2 meters. We believe this distance could be increased with further SLM optical engineering in future work.

3.4.4 MEMS + DOE Implementation.

Fig. 3-8 demonstrates our MEMS + DOE prototype. The dot pattern is projected to a textureless object which improves the disparity compared to passive stereo. When the system moves to another location at t_1, it analyzes the new captured images and directs the ROI to the new position of the textureless object.

3.5 Conclusion

Optical power vs computation power. Although we do not explicitly compare the optical power savings from adaptive sensing with the additional computation power needed for computing the attention map and projector control signal (phase map for SLM), we show that such computations consist of basic arithmetic operations and can be implemented on embedded systems like NVIDIA Jetson Nano, suggesting that our approach can be deployed on increasingly available mobile GPUs. Our system will have even higher benefits for outdoor applications where optical power dominates.

Learning-based attention map and depth estimation. In this work, we use simple, low-complexity texture analysis and semi global matching for attention map and depth estimation. It is possible to design neural networks to achieve better depth estimation, at the cost of higher computation. Our focus in on validating the proposed adaptive sensing as a promising novel sensing strategy, and we expect more practical algorithms to be developed in future work.

Other active depth sensing mechanisms. Although this paper only shows hardware implementations for active stereo, the adaptive sensing strategy can be applied to other depth sensing mechanisms such as single/multi-shot structured light, direct/indirect ToF, FMCW Lidar, among others, which can be a promising future direction.

CHAPTER 4
END-TO-END FOVEATED IMAGING

Deep depth estimation from a single view has been effective at demonstrating the rich geometric cues available in an image [106, 99, 73, 107, 19]. Additionally, these results are improved by using other cues, such as sparse LIDAR or stereo measurements [118, 136, 72, 10]. Our key idea is to notice that most previous monocular approaches assume a nearly equal distribution of sensor pixels across the camera's field-of-view (FOV). In contrast, animal eyes distribute resolution unevenly using fast, mechanical motion, or *saccades*, that change where the eye's fovea views the scene with high acuity. In this paper, we present *SaccadeCam*, a new algorithmic and hardware framework for visual attention control that automatically distributes resolution onto a scene to improve monocular depth estimation.

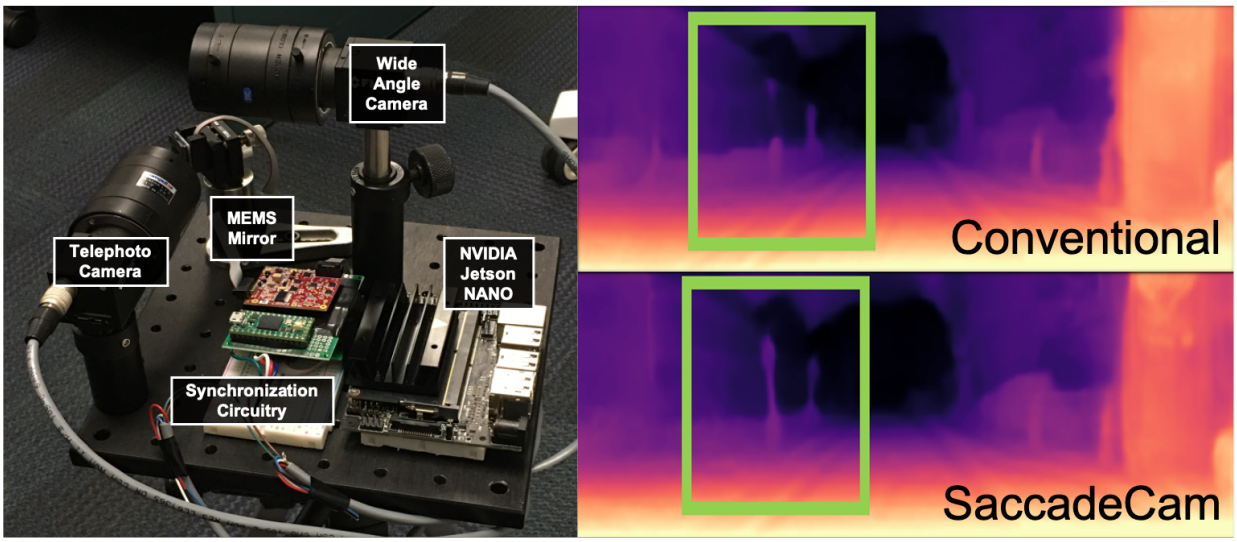


Figure 4-1. Our method learns to foveate resolution end-to-end for depth sensing [116].

Why Leverage Attention for Depth Sensing? Many methods seek to replicate the biological advantages of attention, such as computational efficiency. However, most efforts apply attention within network training and testing, *after* images have been captured [98, 122, 71, 132, 58]. Our framework complements existing attention-based learning, since SaccadeCam leverages visual attention to distribute resolution *during* image capture, and deep attention mechanisms can still be applied after the capture of a SaccadeCam image. Since

SaccadeCam can leverage attention during image capture, it can extract novel efficiencies, particularly for bandwidth of image data. The potential for bandwidth reduction is important — Marr observed that to have foveal resolution everywhere *"...would be wasteful, unnecessary and in violation of our own experience as perceivers..."* [78]. SaccadeCam extracts the biological bandwidth advantages of attention, which impacts platforms that need perception within strict budgetary constraints, such as small robots and long-range drones. We show SaccadeCam results for distributing visual attention (using the proxy of image resolution) to improve depth estimation. In summary, our contributions are:

1. We define a new problem of distributing image resolution under a fixed camera bandwidth around the scene with the goal of succeeding at depth estimation (Sect. 4.2 and Table 4-2).

2. We design an end-to-end network that controls resolution distribution, showing that SaccadeCam images outperform conventional distribution of resolution and can detect important objects for robot navigation, such as poles, signs and distant vehicles (Sect. 4.3, Table 4-3 and Fig. 4-3, Sect. 4.5 Fig. 4-4).

3. We validate our method on a real hardware prototype that images multiple fovea per frame. We also present a generalized selection algorithm to extract discrete fovea from the attention mask. (Sect. 4.5).

4.1 Related Work

Table 4-1. SaccadeCam framework vs. other alternatives.

Method (with few examples)	Adaptive	Test Input	Depth Recovery	Attention *during* image capture	Self/Semi/Guided
Deep Attention Mechanisms [122, 128, 58]	Yes	Mono/Mono+X	Yes	No	All
Compressive Imaging [29]	No	Mono/Mono+X	Yes	No	All
Monocular Depth Estimation [106, 44]	No	Mono	Yes	No	All
Monocular Guided Upsampling [20, 34]	No	Mono+X	Yes	No	Semi/Guided
Adaptive Guided Upsampling [7, 10]	Yes	Mono+X	Yes	No	Guided
End-to-end Optics [16]	No	Mono	Yes	No	Guided
Learned Zoom [135]	No	Mono	No	No	Guided
Adaptive Zoom [119]	Yes	Mono	No	No	Self
SaccadeCam (**Ours**)	Yes	Mono	Yes	Yes	Self

Saccades, attention and related ideas have been studied in robotics and active vision for many years [2, 6, 32, 88, 24, 33, 12]. In addition, foveal designs to enable high-quality imaging

are also common [89, 54, 83, 25]. Our SaccadeCam framework is different in three important ways. First, we explore rich distribution of resolution with *multiple fovea*, which has never been demonstrated before for depth estimation. Second, we apply end-to-end learning to find where to place fovea in a scene to estimate monocular depth with non-uniform spatial resolution. Finally, we demonstrate a working SaccadeCam with a microelectromechanical (MEMS) mirror that is *directly controlled by our trained networks*. We now discuss specific groups of related work, summarized in Table 4-1.

Attention in Deep Learning. Attention in deep learning typically involves learning the parameters of transformations of internal weights, so that the network can differentiably focus on specific regions. Recurrent attention networks, spatial transformer networks and Gaussian attention networks all learn such transformations [67, 61, 41, 57]. [91] show how to optimally select viewing tiles within a FOV for efficient video streaming in VR headsets. There are also approaches that use reinforcement learning for attention when a differentiable attention model is not available [125, 119, 120]. For example, in [119], the goal is to select from a small, fixed number of high-resolution patches to obtain better classification accuracy. In contrast, in our method, patches can be placed anywhere in the FOV, and SaccadeCam controls where patches are placed for depth estimation. In this sense, we take the goals of deep attention mechanisms *inside the camera*, changing how image resolution is distributed under a fixed camera bandwidth.

Monocular and Guided Depth Completion. Monocular depth methods have been very successful [106, 99, 73, 107, 19]. A variety of improvements on these methods by applying a "mono+X" strategy have been proposed [8, 20, 77, 75, 118, 102, 55] with an available benchmark on the KITTI dataset [118]. Upsampling has been shown with sparse depth [121], single-photon imagers [10] and flash lidar [42]. SaccadeCam can be seen a first step towards physical instantiations of recent depth estimation methods that seek to self-improve imperfect measurements [118, 136, 72, 10, 96]. In contrast to these other approaches, our method is a fully passive approach that adaptively distributes resolution to enable successful monocular estimation, see Table 4-1.

Foveated Rendering in VR/AR. Foveation based on eye tracking is used to bypass rendering entire resolution frames in VR/AR headsets [43, 62]. [62] proposed a GAN reconstruction network that is able to take roughly 10% of an image as input and reconstruct a plausible foveated video. Rather than generating compelling viewing, we are interested in foveated imagery for depth estimation.

Compressive Sensing for Vision. Compressive signal processing uses coded optics during capture for applications such as classification [124, 26, 29]. Compressive sensing optimizes bandwidth at the cost of computing (such as L1-optimization), after image capture, to decode the measurements. Our approach is about emphasizing scene areas with new measurements during image capture, reducing bandwidth without extra computing.

Adaptive Imaging for Vision. End-to-end learning inside the camera has impacted many applications in computational cameras and computer vision. These include learning optimal structured light patterns [5], learning optimal lens parameters for monocular depth estimation [16] and HDR imaging [80], and learning optimal sensor designs [14]. SaccadeCam is different in that the optics are not fixed but foveate, enabling active, adaptive changes in imaging inside the camera. This is also what separates us from previous work that does not use learning to decide where to distribute resolution [115]. In this sense, our work is similar to adaptive LIDAR work [97, 72, 10, 96], but instead we seek to control monocular resolution for depth sensing.

4.2 Can Adaptive Attention Improve Depth?

Our hypothesis is that distributing pixels within a camera field-of-view can positively impact monocular depth estimation. This is only possible if models of differing bandwidths perform similarly on smooth consistent regions and perform differently on critical regions. We want to test this hypothesis and build learning mechanisms to distribute these pixels in a self-supervised manner, with no requirement for ground truth labels as recent work has shown [40]. Given a fixed bandwidth, the reduction of resolution in some areas frees up resolution to place onto critical regions such as pedestrians, signs, cars and foliage. In the next section, we discuss how to decide where to place the resolution and demonstrate the validity of our

hypothesis. Now, we discuss the implications of our approach in Table 4-2.

4.2.1 Bandwidth

Table 4-2 has three baselines at different bandwidths. **We define bandwidth** as the number of angular samples across the FOV, i.e. our notion of bandwidth is identical to angular resolution. Therefore, while for practical reasons we may show images of the same spatial resolution (i.e. pixels in computer memory), they are of very different angular resolution. For all our experiments we use images with camera intrinsics from the KITTI dataset [35], from which we simulate different camera resolutions.

We simulate bandwidth by downsampling based on the scaled intrinsic matrix and then upsampling back to original resolution. This simulates a camera that, in practice, would have less resolution bandwidth over the same field of view. The three baselines in Table 4-2 are full resolution (70 px/mm bandwidth), target resolution (31.30 px/mm bandwidth) and three low-resolution images that we term as wide-angle camera (WAC) bandwidth in the context of the SaccadeCam hardware in Sect. 4.5.

4.2.2 Depth from SaccadeCam Images

In our experiments we use the ground truth color images as the full resolution. The high resolution attention regions in our SaccadeCam images are also at the full resolution. We compare equiangular sampling of the target resolution with SaccadeCam images that have to be at the same bandwidth as the target resolution. *SaccadeCam images are created by fusing high resolution images into attention regions within the low-resolution WAC images.* The WAC resolution and the number of attention regions are constrained by the fact that their sum must

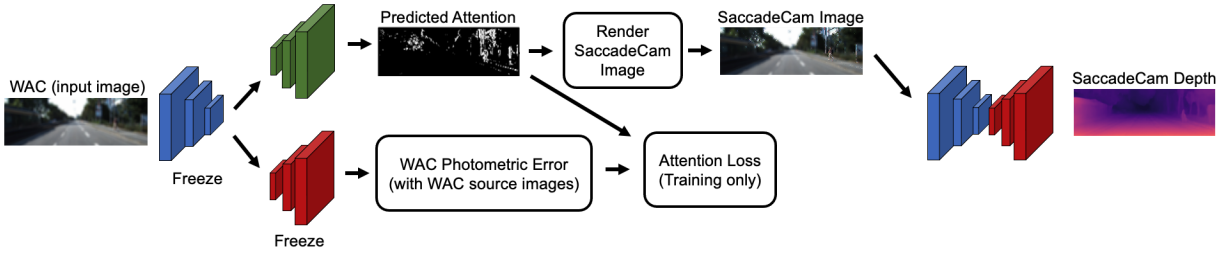


Figure 4-2. Self-supervised foveation network [116].

Table 4-2. Oracle motivation from the KITTI dataset [35].

		Abs Rel	Sq Rel	RMSE	RMSE log	δ<1.25	δ<1.25^2	δ<1.25^3
	Full Resolution (70 pixels/mm)	0.109	0.883	4.960	0.208	0.865	0.949	0.975
	Target resolution (31 pixels/mm)	0.118	0.988	5.188	0.214	0.851	0.944	0.974
	Wide Angle Camera (27 pixels/mm)	0.119	0.991	5.238	0.216	0.846	0.943	0.974
	Photometric Oracle	0.116	0.941	5.134	0.213	0.851	0.945	0.975
(a)	**True Oracle**	**0.114**	**0.853**	**4.850**	**0.208**	**0.857**	**0.950**	**0.976**
	Wide Angle Camera (22 pixels/mm)	0.121	1.005	5.275	0.219	0.840	0.939	0.973
	Photometric Oracle	0.116	0.931	5.114	0.214	0.848	0.943	0.974
(b)	**True Oracle**	**0.111**	**0.850**	**4.846**	**0.206**	**0.863**	**0.950**	**0.976**
	Wide Angle Camera (15 pixels/mm)	0.128	1.067	5.507	0.228	0.824	0.934	0.971
	Photometric Oracle	0.120	0.960	5.238	0.219	0.840	0.941	0.973
(c)	**True Oracle**	**0.112**	**0.847**	**4.848**	**0.206**	**0.866**	**0.951**	**0.976**

equal the target angular resolution. While monocular images with equiangular resolutions have a variety of methods for depth estimation, these cannot be used directly on SaccadeCam images without training or fine tuning. This is because SaccadeCam images have spatially varying resolution, and in Sect. 4.3 we discuss how to extract depth from such monocular imagery. Now we discuss the implications of what is possible if such SaccadeCam depth estimation is solved.

4.2.3 Oracles

Our approach is to compare monocular depth estimation of equiangular images with SaccadeCam images, created by unevenly distributed resolution. We design oracle experiments that determine ideal locations to distribute resolution to, and then place focused depth predictions as a perfect color-to-depth mapping in the attention regions.

For the Photometric Oracle in Table 4-2, the attention regions are computed based on the top N locations of the difference between the WAC depth prediction errors from a fully trained WAC network and full resolution depth prediction errors from a fully trained full resolution network using the method of [40]. We then replace the WAC depth with focused depth in the attention regions. N is the limit of available pixels left after the target resolution and WAC resolutions are determined from our camera model. We hypothesize that the focused depth errors should be lower than WAC depth errors in high resolution attention regions and similar to WAC depth errors in smooth geometrically consistent regions.

For the True Oracle in Table 4-2, the attention regions are computed based on the top N

locations of the difference between WAC depth and ground truth LIDAR, where N is scaled according to the number of LIDAR samples versus full resolution for fair comparison. We then replace the WAC depth with focused depth in the attention regions. Therefore, if the worst depth estimates of WAC images are replaced by the corresponding depths in the same regions of full resolution images, then, as can be seen by the Table 4-2, depth from SaccadeCam *has the potential* to outperform state-of-the-art. Our oracle experiments support our idea that better resolution can help with depth estimation as suggested in [78, 40].

4.3 End-to-end Learning for Adaptive Attention

In Figure 4-2 we depict the complete flow for our self-supervised method. Our system consists of one encoder (blue) and two decoders (red and green). Each of these are designed for self-supervised stereo, following the method of [40]. Our method could easily be integrated with self-supervised monocular training as well, since the pose can be estimated from multiple views of a single camera using a pose network. At test time the flow in Fig. 4-2 is monocular (single image), but at training time, each network takes a stereo pair.

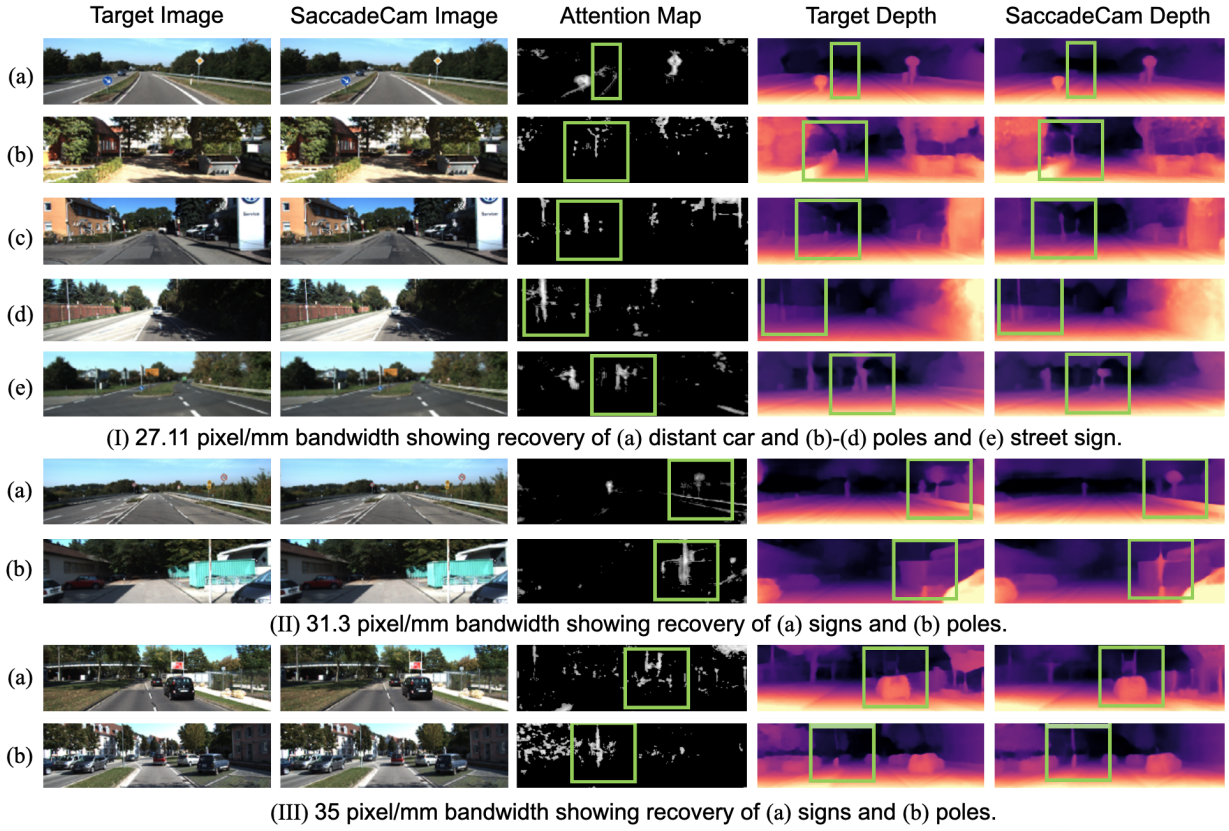


Figure 4-3. Overview of our KITTI results [116].

Adaptive Attention. The attention decoder (green in Figure 4-2) is trained with a stereo pair of low-resolution, wide angle camera (WAC) images. The attention decoder input is the latent vector of the training depth encoder. The attention decoder then predicts per pixel attention and calculates binary cross entropy loss against the "true" binary attention mask given by the top photometric error regions calculated from the training depth network. This trains the attention

mask towards 1. Our insight is that these error regions should be where additional resolution might make a difference. However, we are not strictly tied to the photometric error, as we will soon see. We then differentiably render a SaccadeCam image using the predicted attention mask, focused image, and WAC image. Here the bandwidth is given by the maximum number of samples that are possible at the highest resolution of the system. The bandwidth is a function of the target resolution and the amount of bandwidth that has already been used up by the WAC image.

SaccadeCam Rendering. Our SaccadeCam rendering module consists of alpha blending a focused image onto the WAC image using an attention mask as the blend weight. We use this to create SaccadeCam images from either a learned or oracle attention mask $\mathbf{M}$. This allows us to differentiably train our attention network end to end with a downstream monocular network,

$$I_{SaccadeCam} = \mathbf{M} \odot (I_{focused}) + (1 - \mathbf{M}) \odot (I_{WAC}). \tag{4-1}$$

Depth Network and Attention Regularization. The last module is the encoder-decoder pair (blue and red) that converts the SaccadeCam image into a depth. When calculating the view synthesis photometric loss [40], we compute the loss between the target SaccadeCam image and the synthesized target image that is also foveated with the same attention mask, but with the synthesized focused target image in the attention regions. The encoder and decoder used in SaccadeCam depth estimation are the same used in obtaining the WAC depth during attention estimates. During attention estimation, the gradients of the depth encoder and decoder pair are frozen. In other words, the encoder and decoder drifts towards monocular SaccadeCam image depth reconstruction, while also regularizing attention estimates. Practically, such a system is more efficient since it shares SaccadeCam features with the attention module and allows for flexible attention beyond WAC photometric errors.

Loss Terms. Our final loss is $L = \mu L_p + \lambda L_s + \alpha L_a$. L_p and L_s follow the view synthesis photometric loss and depth smoothness loss common in monocular depth estimation. We set

Table 4-3. SaccadeCam compared against equiangular (conventional) images.

		Fovea weighting	Abs Rel	Sq Rel	RMSE	RMSE log	δ<1.25	δ<1.25^2	δ<1.25^3
	Full Resolution (70 pixels/mm)	No	0.109	0.883	4.960	0.208	0.865	0.949	0.975
	Target Resolution (35 pixels/mm)	No	0.117	1.001	5.144	0.213	0.855	0.946	0.974
	Wide Angle Camera (30 pixels/mm)	No	0.119	1.026	5.202	0.216	0.850	0.943	0.974
	Ours no weighting	No	0.115	0.942	5.087	0.209	0.853	0.948	0.976
	Ours fovea weighting	Yes	0.116	0.950	5.038	0.206	0.852	0.948	0.977
	Color edges no weighting	No	0.122	0.974	5.278	0.220	0.836	0.940	0.973
(a)	Color edges fovea weighting	Yes	0.123	0.958	5.267	0.220	0.831	0.940	0.974
	Target Resolution (27 pixels/mm)	No	0.118	1.013	5.209	0.215	0.848	0.943	0.974
	Wide Angle Camera (23 pixels/mm)	No	0.121	0.996	5.264	0.219	0.839	0.940	0.973
	Ours no weighting	No	0.121	1.003	5.192	0.211	0.844	0.945	0.976
	Ours fovea weighting	Yes	0.119	0.938	5.161	0.211	0.842	0.944	0.976
	Color edges no weighting	No	0.137	1.124	5.721	0.247	0.797	0.920	0.964
(b)	Color edges fovea weighting	Yes	0.134	1.056	5.660	0.240	0.801	0.924	0.967
	Target Resolution (8 pixels/mm)	No	0.194	2.705	7.378	0.296	0.730	0.889	0.949
	Wide Angle Camera (7 pixels/mm)	No	0.234	4.144	8.317	0.330	0.686	0.867	0.937
	Ours no weighting	No	0.167	1.516	6.815	0.270	0.743	0.900	0.958
	Ours fovea weighting	Yes	0.164	1.463	6.555	0.256	0.754	0.909	0.964
	Color edges no weighting	No	0.167	1.514	6.836	0.273	0.741	0.898	0.957
(c)	Color edges fovea weighting	Yes	0.167	1.472	6.589	0.260	0.747	0.907	0.963

$\mu = 1$ to avoid masking out fovea regions and $\lambda = 0.001$. L_a is the binary cross entropy loss between the predicted attention and WAC photometric error given by the SaccadeCam depth network. We freeze our depth network and set $\mu = \lambda = 0$, $\alpha = 1$ when training the attention network. We found that the attention decoder learned much quicker than the depth network (roughly 5 epochs for attention compared to roughly 20 epochs for high bandwidth depth). We also found that an attention network trained on a single bandwidth generalizes well across different bandwidths. In an online setting, we hypothesize that infrequently updating or significantly lowering the learning rate of the attention network relative to the depth network would be beneficial.

4.4 Experiments

We implement our network in PyTorch on a single NVIDIA GTX 1080 Ti. Our encoder architecture is a ResNet18 and our decoder architecture is similar to [40]. All our training was initialized with ImageNet parameters. In Table 4-3, we show our results over a few different bandwidths. We found our SaccadeCam depth networks finished training earlier than networks trained on equiangular images based on the validation error. We train the depth networks of (a), (b), (c) for 17, 11, 2 epochs respectively and the attention networks of (a), (b), (c) for 5 epochs each. We train all equiangular resolution models for 20 epochs. *Note that not all bandwidths are*

appropriate for SaccadeCam. For example, extremely high-resolution images may not benefit from bandwidth optimization, and very low resolution images may result in extreme WAC depth errors.

We also explored weighting our loss with a weighted binary version of the predicted attention mask based on the observation that high resolution models train longer than low resolution models, this supports giving the high resolution attention region more weighting during training since the periphery is lower resolution. We train the weighted variants of (a), (b), (c) for 7, 14, 1 epochs respectively. Overall the region weighting boosts performance and speeds up training. We found at higher bandwidth SaccadeCam data the region weighting delta must be smaller because, while the periphery is lower resolution than the high resolution attention region, it is still high enough resolution that it needs a stronger weighting to train. We weight the foveal/WAC regions of the photometric error 1.15/0.85, 1.25/0.75, 1.5/0.5 for (a), (b), and (c) respectively in Table 4-3.

We compare our results to monocular self-supervised depth reconstruction at the target resolution. We also compare to a color edge detector as an attention proxy. We found that edges performed well at very low resolutions, but performed poorly at higher resolutions where the fovea must be more intelligently placed to meaningfully impact performance. For our SaccadeCam networks, we first train our depth networks using the WAC photometric error as an attention proxy. We then train the attention network with the same frozen depth network using the WAC photometric error as psuedo ground truth as described in Section 4.3. At test time, we use the learned attention mask. We found $\geq 95\%$ overlap between the predicted attention masks and error regions on average for the test set across bandwidths, which shows the attention masks sufficiently learned to represent the error regions.

Fig. 4-3 shows visual results from our SaccadeCam models. Our hypothesis holds true in that we perform similar to equiangular models on smooth and geometrically consistent scene regions while outperforming equiangular models on irregular edge-case regions. Notice the SaccadeCam framework allows us to detect road signs, poles, and other distant objects such as

cars that the equiangular models cannot detect.

4.5 SaccadeCam Hardware Prototype

Here we discuss a physical instantiation of SaccadeCam that can adaptively distribute resolution onto regions of interest based on our trained models. SaccadeCam consists of a low-resolution wide angle camera (WAC) whose field-of-view (FOV) covers the scene, and a narrow FOV telephoto camera that views reflections off a small, fast moving microelectromechanical (MEMS) mirror. These components are collectively the SaccadeCam device seen in Fig. 4-1.

Unlike many other MEMS mirror enabled devices (such as LIDARs [31, 108, 69]), we do not run our MEMS mirror at resonance. Instead we use a specific scan pattern, and we are able to control 5 points (i.e. 5 fovea) in the FOV at 5 Hz. This speed is reasonably fast for most objects in common scenes for depth inference. Our telephoto and WAC cameras consist of a 1.6 MP FLIR Blackfly S-U3-16S2C-CS, where the telephoto camera has a 30mm lens and the WAC camera has a 6mm lens. The telephoto camera views reflections off a 3.6mm Mirrorcle Technologies MEMS mirror with custom modifications to prevent ghosting artifacts induced from MEMS electronic packaging. Our main computer is a NVIDIA Jetson NANO, a popular embedded board with GPU and CUDA capabilities. We trace our PyTorch models to TorchScript so we can run our models on-device in C++. The Jetson NANO communicates with custom synchronization circuitry containing a Teensy 4.0 microcontroller that triggers the cameras and MEMS mirror in lockstep. The MEMS mirror is physically controlled from the Teensy through a Mirrorcle Technologies PicoAmp 5.4 X200 Digital to Analog Converter. Our hardware prototype is capable of on-device training although we leave this for future work.

4.5.1 Feasible Fovea from the Attention Mask

In Sect. 4.3 we discussed how to process the input, low-resolution WAC image to produce an attention mask across the WAC FOV, with the goal of increasing resolution in this region up to the bandwidth limit. Such an attention mask is deformable and non-convex, in the sense that there are no restrictions on optical feasibility of sensing the attention region in higher resolution, quickly.

In this section we discuss how to extract a discrete number of optically feasible saccades from the attention mask for a practical MEMS-mirror-based SaccadeCam. We also contend that it will apply to any camera that is not capable of producing programmable spatially varying deformable point spread functions (PSFs). While phase masks [133] can achieve these types of deformable attention masks, they are both slow and work best with coherent light, rather than incoherent light from a scene.

Our goal is to maximize attention mask coverage with n saccades, or mirror viewpoints. These correspond to n pairs of voltages that specify the MEMS mirror viewpoints, $\{(\theta(V(t_1)),\phi(V(t_1))),...(\theta(V(t_n)),\phi(V(t_n)))\}$. We first tackle the problem of fixed foveal size or fovea FOV, and then we generalize such that each viewing direction i could have its own unique FOV (perhaps using a liquid lens [137]).

Greedy Attention Algorithm. The greedy algorithm requires an attention mask and a *fixed* angular fovea size ω_{fovea}. Given an attention mask defined on the FOV, $\mathbf{A}(\omega)$ where $\omega \in \omega_{fov}$, we can find the location of the maximum attention value, ω_{max} in this mask. We then follow an iterative procedure, where we capture a fovea by selecting t_1 such that the first mirror direction $(\theta(V(t_1)),\phi(V(t_1)))$ points along the central axis of the solid angle defined by ω_{max}. We then destroy attention mask information around the first maximum such that $\mathbf{A}(\omega) = 0$, where $\omega \in \omega_{fov}$ *and* $\|\omega_{max} - \omega\| \leq \omega_{fovea}$. We then repeat the procedure n times for n fovea, until a set of mirror voltages are obtained $\{(\theta(V(t_1)),\phi(V(t_1))),...(\theta(V(t_n)),\phi(V(t_n)))\}$. The proof of this method follows from the greedy selection of subsequently maximum attention values, all of which are monotonically decreasing (i.e. ω_{max} for t_1 is less than ω_{max} at t_2 and so on). Therefore, there is no way that there exists an attention value at location ω_{missed} that is greater than the n selected values at different locations of ω_{max}, because otherwise it would have been selected for measurement at some point between t_1 and t_n. We present derivations for an advanced attention coverage algorithm based on the optical knapsack algorithm from [94] in the supplementary, although we do not implement this algorithm in hardware.

4.5.2 Hardware Prototype Results

We show qualitative results on real data captured with our SaccadeCam hardware prototype in Fig. 4-4. Our results are obtained on-device at video rate as follows. The NVIDIA Jetson NANO triggers the WAC camera and passes the WAC image through our trained attention network. Next, given a calibrated MEMS mirror, telephoto and WAC cameras, we determine the top ten pixel locations (and therefore MEMS voltages) that optimally cover the attention prediction with our greedy algorithm. The mirror is triggered and moves to a location whereby the telephoto camera is subsequently triggered to capture an image of the MEMS mirror reflection. We choose ten fovea for our hardware prototype so the previous step is repeated until ten MEMS mirror images are captured; the Hardware Attention column in Fig. 4-4 shows examples of the ten captured MEMS mirror images taken by the telephoto camera. We then gamma correct and blend the telephoto camera images onto the WAC image to form the SaccadeCam image. Finally, the SaccadeCam image is passed through our depth network to obtain our result.

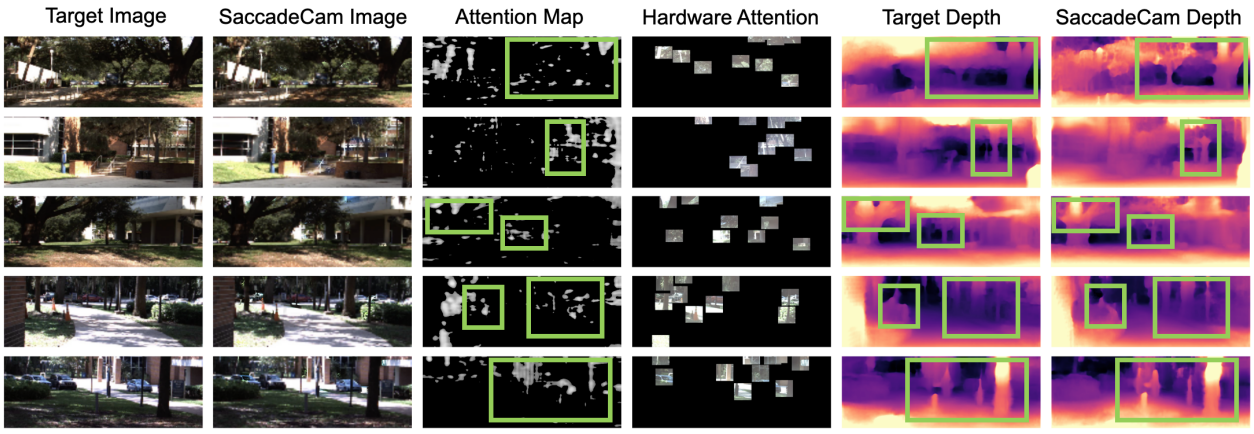


Figure 4-4. Results for real data captured with our SaccadeCam hardware prototype [116].

For results with our SaccadeCam hardware prototype, we keep the target resolution bandwidth at 35 px/mm and SaccadeCam WAC bandwidth at 31 px/mm with ten fovea. This lets us use models trained on the much larger KITTI dataset. For target depth we use the 20 epoch weights of 35 px/mm target bandwidth. We finetune SaccadeCam weights for 5 epochs at 1e-7 learning rate on KITTI with patch fovea to smooth out rough square boundary edges occurring

when overlaying fovea images onto the WAC image since the fovea images are square and do not perfectly approximate the learned attention.

Fig. 4-4 shows that our hardware prototype can qualitatively match the results seen on the KITTI test set in Fig. 4-3 in that SaccadeCam depth outperforms target depth in the learned attention regions thanks to the natively-high angular resolution of the telephoto camera viewing the MEMS mirror.

4.6 Conclusion

We introduced a new framework, SaccadeCam, for leveraging visual attention during image formation. Our key idea is to adaptively distribute resolution onto the scene, to improve depth sensing, demonstrating that our framework can perform better than equiangular distribution of pixels. We now discuss some limitations that we would like to improve in future work:

Real-time demonstrations. Our current hardware prototype allows for on-device end-to-end learning at nearly 5 Hz. We want to demonstrate dynamic scenes results soon with faster hardware.

Deformable attention masks. Our setup and theory already allow deformable attention masks, and we wish to use a liquid lens to demonstrate this.

Beyond depth estimation. The differentiable and modular nature of the SaccadeCam framework encourages integrating SaccadeCam into other existing vision applications such as semantic segmentation or pedestrian detection.

CHAPTER 5
SUMMARY AND CONCLUSIONS

Throughout this dissertation, the idea of optimizing visual sensing through principles derived from biological foveation has been rigorously explored. Each chapter provided insights and advancements in unique aspects of this overarching theme. Here's a summary and forward-looking synthesis of the topics discussed:

Our passive foveated imaging system, FoveaCam, revealed the inherent advantages of quickly modulating a pixel-dense camera viewpoint [115, 114]. This has direct implications for fields such as robotics, augmented reality, and autonomous vehicles, especially in tasks like 3D reconstruction, long-range navigation, and enhancing safety in critical regions. However, as noted, there are still areas like integrating an auto-focusing element and reducing the camera's size that would further its practical applications. It's also essential to compare the camera's performance with other competing sensors and datasets, which would be a significant next step in solidifying its position in the realm of advanced imaging.

Next, the work presented on active foveated imaging improved energy-efficient adaptive 3D sensing [117]. While our approach provides a promising blueprint for energy-efficient adaptive sensing, especially valuable for outdoor 3D sensing applications, the balance and comparison between optical and computational resources require further exploration. The introduction of neural networks for depth estimation and the potential application of adaptive sensing across different depth sensing mechanisms, such as ToF or FMCW Lidar, hints at a vast and rich field of future study.

Finally, the end-to-end foveated imaging system, SaccadeCam, offers an exciting paradigm shift in how we approach visual attention during image formation [116]. The preliminary results show that adaptively distributing resolution can improve depth sensing beyond traditional methods. However, pushing its limits in real-time scenarios, using deformable attention masks, and integrating its capabilities into other vision tasks, such as semantic segmentation or detection, represents the future direction of this work.

In conclusion, the adaptive, foveation-inspired imaging systems explored in this dissertation

hold the promise of revolutionizing many fields where visual sensing plays a pivotal role. They offer a blend of the tried-and-tested principles from nature and cutting-edge technology. While we have made substantial strides in understanding and building these systems, the journey ahead promises further advancements and applications, ensuring that the next wave of imaging systems will be more intelligent, adaptive, and efficient.

LIST OF REFERENCES

[1] Supreeth Achar, Joseph R Bartels, William L Whittaker, Kiriakos N Kutulakos, and Srinivasa G Narasimhan, *Epipolar time-of-flight imaging*, ACM Transactions on Graphics (TOG) **36** (2017), no. 4, 37.

[2] John Aloimonos, Isaac Weiss, and Amit Bandyopadhyay, *Active vision*, International journal of computer vision **1** (1988), no. 4, 333–356.

[3] Siddharth Ancha, Gaurav Pathak, Srinivasa G Narasimhan, and David Held, *Active safety envelopes using light curtains with probabilistic guarantees*, arXiv preprint arXiv:2107.04000 (2021).

[4] Siddharth Ancha, Yaadhav Raaj, Peiyun Hu, Srinivasa G Narasimhan, and David Held, *Active perception using light curtains for autonomous driving*, European Conference on Computer Vision, Springer, 2020, pp. 751–766.

[5] Seung-Hwan Baek and Felix Heide, *Polka lines: Learning structured illumination and reconstruction for active stereo*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021.

[6] Ruzena Bajcsy, *Active perception*, (1988).

[7] Joseph R Bartels, Jian Wang, William Whittaker, Srinivasa G Narasimhan, et al., *Agile depth sensing using triangulation light curtains*, Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 7900–7908.

[8] Ramy Battrawy, René Schuster, Oliver Wasenmüller, Qing Rao, and Didier Stricker, *Lidar-flow: Dense scene flow estimation from sparse lidar and stereo images*, arXiv preprint arXiv:1910.14453 (2019).

[9] Jacques M Beckers, *Adaptive optics for astronomy: principles, performance, and applications*, Annual review of astronomy and astrophysics **31** (1993), no. 1, 13–62.

[10] A. Bergman, D. Lindell, and G. Wetzstein, *Deep adaptive lidar: End-to-end optimization of sampling and depth completion at low sampling rates*, ICCP (2020).

[11] David Beymer and Myron Flickner, *Eye gaze tracking using an active stereo head*, 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings., vol. 2, IEEE, 2003, pp. II–451.

[12] Neil Bruce and John Tsotsos, *Attention based on information maximization*, Journal of Vision **7** (2007), no. 9, 950–950.

[13] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard, *Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age*, IEEE Transactions on robotics **32** (2016), no. 6, 1309–1332.

[14] Ayan Chakrabarti, *Learning sensor multiplexing design through back-propagation*, Advances in Neural Information Processing Systems, 2016, pp. 3081–3089.

[15] Dorian Chan, Srinivasa G Narasimhan, and Matthew O'Toole, *Holocurtains: Programming light curtains via binary holography*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022.

[16] Julie Chang and Gordon Wetzstein, *Deep optics for monocular depth estimation and 3d object detection*, 2019.

[17] Benjamin Charrow, Gregory Kahn, Sachin Patil, Sikang Liu, Ken Goldberg, Pieter Abbeel, Nathan Michael, and Vijay Kumar, *Information-theoretic planning with trajectory optimization for dense 3d mapping.*, Robotics: Science and Systems, vol. 11, Rome, 2015.

[18] Nanhu Chen, Benjamin Potsaid, John T Wen, Scott Barry, and Alex Cable, *Modeling and control of a fast steering mirror in imaging applications*, 2010 IEEE International Conference on Automation Science and Engineering, IEEE, 2010, pp. 27–32.

[19] Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng, *Single-image depth perception in the wild*, Advances in neural information processing systems, 2016, pp. 730–738.

[20] Zhao Chen, Vijay Badrinarayanan, Gilad Drozdov, and Andrew Rabinovich, *Estimating depth from rgb and sparse sensing*, Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 167–182.

[21] Dong-Chan Cho, Wah-Seng Yap, HeeKyung Lee, Injae Lee, and Whoi-Yul Kim, *Long range eye gaze tracking system for a large screen*, IEEE Transactions on Consumer Electronics **58** (2012), no. 4, 1119–1128.

[22] Iulian B Ciocoiu, *Foveated compressed sensing*, Circuits, Systems, and Signal Processing **34** (2015), no. 3, 1001–1015.

[23] O. Cossairt, D. Miau, and S.K. Nayar, *Gigapixel Computational Imaging*, IEEE International Conference on Computational Photography (ICCP), Mar 2011.

[24] Donald G Dansereau, Ian Mahon, Oscar Pizarro, and Stefan B Williams, *Plenoptic flow: Closed-form visual odometry for light field cameras*, 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2011, pp. 4455–4462.

[25] Trevor Darrell, Baback Moghaddam, and Alex P Pentland, *Active face tracking and pose estimation in an interactive room*, Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, 1996, pp. 67–72.

[26] Mark A Davenport, Petros Boufounos, Michael B Wakin, Richard G Baraniuk, et al., *Signal processing with compressive measurements.*, J. Sel. Topics Signal Processing **4** (2010), no. 2, 445–460.

[27] James Davis, Ravi Ramamoorthi, and Szymon Rusinkiewicz, *Spacetime stereo: A unifying framework for depth from triangulation*, 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings., vol. 2, IEEE, 2003, pp. II–359.

[28] Chong Ding, Bi Song, Akshay Morye, Jay A Farrell, and Amit K Roy-Chowdhury, *Collaborative sensing in a distributed ptz camera network*, IEEE Transactions on Image Processing **21** (2012), no. 7, 3282–3295.

[29] Marco F Duarte, Mark A Davenport, Dharmpal Takhar, Jason N Laska, Ting Sun, Kevin F Kelly, and Richard G Baraniuk, *Single-pixel imaging via compressive sampling*, IEEE signal processing magazine **25** (2008), no. 2, 83–91.

[30] Fixstars, *libsgm*, https://github.com/fixstars/libSGM.

[31] Thomas P Flatley, *Spacecube: A family of reconfigurable hybrid on-board science data processors*, (2015).

[32] Simone Frintrop and Patric Jensfelt, *Attentional landmarks and active gaze control for visual slam*, IEEE Transactions on Robotics **24** (2008), no. 5, 1054–1065.

[33] Simone Frintrop, Erich Rome, and Henrik I Christensen, *Computational visual attention systems and their cognitive foundations: A survey*, ACM Transactions on Applied Perception (TAP) **7** (2010), no. 1, 6.

[34] Rahul Garg, Neal Wadhwa, Sameer Ansari, and Jonathan T Barron, *Learning single camera depth estimation using dual-pixels*, Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 7628–7637.

[35] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun, *Vision meets robotics: The kitti dataset*, The International Journal of Robotics Research **32** (2013), no. 11, 1231–1237.

[36] Andreas Geiger, Philip Lenz, and Raquel Urtasun, *Are we ready for autonomous driving? the kitti vision benchmark suite*, 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 3354–3361.

[37] David Geisler, Dieter Fox, and Enkelejda Kasneci, *Real-time 3d glint detection in remote eye tracking based on bayesian inference*, 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2018, pp. 7119–7126.

[38] R W Gerchberg and W O Saxton, *A Practical Algorithm for the Determination of Phase from Image and Diffraction Plane Pictures*, Optik **35** (1972), 237–246.

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

[40] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J. Brostow, *Digging into self-supervised monocular depth prediction*, (2019).

[41] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra, *Draw: A recurrent neural network for image generation*, 2015.

[42] Tobias Gruber, Frank Julca-Aguilar, Mario Bijelic, Werner Ritter, Klaus Dietmayer, and Felix Heide, *Gated2depth: Real-time dense lidar from gated images*, arXiv preprint arXiv:1902.04997 (2019).

[43] Brian Guenter, Mark Finch, Steven Drucker, Desney Tan, and John Snyder, *Foveated 3d graphics*, ACM Trans. Graph. (2012).

[44] Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon, *3d packing for self-supervised monocular depth estimation*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 2485–2494.

[45] Mohit Gupta, Shree K Nayar, Matthias B Hullin, and Jaime Martin, *Phasor imaging: A generalization of correlation-based time-of-flight imaging*, ACM Transactions on Graphics (ToG) **34** (2015), no. 5, 156.

[46] Mohit Gupta, Qi Yin, and Shree K. Nayar, *Structured light in sunlight*, 2013 IEEE International Conference on Computer Vision, 2013, pp. 545–552.

[47] Samuel W Hasinoff, Frédo Durand, and William T Freeman, *Noise-optimal capture for high dynamic range photography*, 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 553–560.

[48] Tim Hawkins, Per Einarsson, and Paul E Debevec, *A dual light stage.*, Rendering Techniques **5** (2005), 91–98.

[49] Felix Heide, Matthias B Hullin, James Gregson, and Wolfgang Heidrich, *Low-budget transient imaging using photonic mixer devices*, ACM Transactions on Graphics (ToG) **32** (2013), no. 4, 45.

[50] Craig Hennessey and Jacob Fiset, *Long range eye tracking: bringing eye tracking into the living room*, Proceedings of the Symposium on Eye Tracking Research and Applications, ACM, 2012, pp. 249–252.

[51] Radu Horaud, Miles Hansard, Georgios Evangelidis, and Clément Ménier, *An overview of depth cameras and range scanners based on time-of-flight technologies*, Machine Vision and Applications **27** (2016), 1005–1020.

[52] C. K. Hsueh and A. A. Sawchuk, *Computer-generated double-phase holograms*, Applied Optics **17** (1978), no. 24, 3874–3883.

[53] Hou-Ning Hu, Qi-Zhi Cai, Dequan Wang, Ji Lin, Min Sun, Philipp Krahenbuhl, Trevor Darrell, and Fisher Yu, *Joint monocular 3d vehicle detection and tracking*, Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 5390–5399.

[54] Hong Hua and Sheng Liu, *Dual-sensor foveated imaging system*, Applied optics **47** (2008), no. 3, 317–327.

[55] Tak-Wai Hui, Chen Change Loy, and Xiaoou Tang, *Depth map super-resolution by deep multi-scale guidance*, Proceedings of European Conference on Computer Vision (ECCV), 2016, pp. 353–369.

[56] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al., *Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera*, Proceedings of the 24th annual ACM symposium on User interface software and technology, ACM, 2011, pp. 559–568.

[57] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu, *Spatial transformer networks*, 2016.

[58] Adrian Johnston and Gustavo Carneiro, *Self-supervised monocular trained depth estimation using self-attention and discrete disparity volume*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 4756–4765.

[59] Andrew Jones, Ian McDowall, Hideshi Yamada, Mark Bolas, and Paul Debevec, *Rendering for an interactive 360 light field display*, ACM Transactions on Graphics (TOG) **26** (2007), no. 3, 40.

[60] Michael Jones and Paul Viola, *Fast multi-view face detection*, Mitsubishi Electric Research Lab TR-20003-96 **3** (2003), no. 14, 2.

[61] Samira Ebrahimi Kahou, Vincent Michalski, and Roland Memisevic, *Ratm: Recurrent attentive tracking model*, 2016.

[62] Anton S. Kaplanyan, Anton Sochenov, Thomas Leimkühler, Mikhail Okunev, Todd Goodall, and Gizem Rufo, *Deepfovea: Neural reconstruction for foveated rendering and video compression using learned statistics of natural videos*, ACM Trans. Graph. (2019).

[63] Abhishek Kasturi, Veljko Milanovic, Bryan H Atwood, and James Yang, *Uav-borne lidar with mems mirror-based scanning capability*, Proc. SPIE, vol. 9832, 2016, p. 98320M.

[64] Leonid Keselman, John Iselin Woodfill, Anders Grunnet-Jepsen, and Achintya Bhowmik, *Intel realsense stereoscopic depth cameras*, 2017.

[65] Mohamed Khamis, Axel Hoesl, Alexander Klimczak, Martin Reiss, Florian Alt, and Andreas Bulling, *Eyescout: Active eye tracking for position and movement independent gaze interaction with large public displays*, Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology, ACM, 2017, pp. 155–166.

[66] Hiromasa Oku Kohei Okumura and Masatoshi Ishikawa, *High-speed gaze controller for millisecond-order pan/tilt camera*, ICRA (2011).

[67] Adam R. Kosiorek, Alex Bewley, and Ingmar Posner, *Hierarchical attentive recurrent tracking*, 2017.

[68] Kyle Krafka, Aditya Khosla, Petr Kellnhofer, Harini Kannan, Suchendra Bhandarkar, Wojciech Matusik, and Antonio Torralba, *Eye tracking for everyone*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[69] Krassimir T Krastev, Hendrikus WLAM Van Lierop, Herman MJ Soemers, Renatus Hendricus Maria Sanders, and Antonius Johannes Maria Nellissen, *Mems scanning micromirror*, September 3 2013, US Patent 8,526,089.

[70] Anat Levin, Rob Fergus, Frédo Durand, and William T Freeman, *Image and depth from a conventional camera with a coded aperture*, ACM transactions on graphics (TOG) **26** (2007), no. 3, 70.

[71] Zheng Li, Ying Wei, Yu Zhang, and Qiang Yang, *Hierarchical attention transfer network for cross-domain sentiment classification*, Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, 2018.

[72] Chao Liu, Jinwei Gu, Kihwan Kim, Srinivasa Narasimhan, and Jan Kautz, *Neural rgb-to-d sensing: Depth and uncertainty from a video camera*, arXiv preprint arXiv:1901.02571 (2019).

[73] Miaomiao Liu, Mathieu Salzmann, and Xuming He, *Discrete-continuous depth estimation from a single image*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 716–723.

[74] Sheng Liu, Craig Pansing, and Hong Hua, *Design of a foveated imaging system using a two-axis mems mirror*, International Optical Design Conference 2006, vol. 6342, International Society for Optics and Photonics, 2006, p. 63422W.

[75] Jiajun Lu and David Forsyth, *Sparse depth super resolution*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 2245–2253.

[76] Andrew Maimone, Andreas Georgiou, and Joel S. Kollin, *Holographic near-eye displays for virtual and augmented reality*, ACM Trans. Graph. **36** (2017), no. 4.

[77] Fangchang Mal and Sertac Karaman, *Sparse-to-dense: Depth prediction from sparse depth samples and a single image*, 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2018, pp. 1–8.

[78] David Marr, *Vision: A computational investigation into the human representation and processing of visual information*, (1982).

[79] Nathan Matsuda, Oliver Cossairt, and Mohit Gupta, *MC3D: Motion Contrast 3D Scanning*, IEEE International Conference on Computational Photography (ICCP) (Houston, TX, USA), IEEE, April 2015, pp. 1–10.

[80] Christopher A. Metzler, Hayato Ikoma, Yifan Peng, and Gordon Wetzstein, *Deep optics for single-shot high-dynamic-range imaging*, 2019.

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

[82] Veljko Milanović, Abhishek Kasturi, James Yang, and Frank Hu, *A fast single-pixel laser imager for vr/ar headset tracking*, Proc. of SPIE Vol, vol. 10116, 2017, pp. 101160E–1.

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

[84] Shree K Nayar, Vlad Branzoi, and Terry E Boult, *Programmable imaging: Towards a flexible camera*, International Journal of Computer Vision **70** (2006), no. 1, 7–22.

[85] Ren Ng, *Fourier slice photography*, ACM transactions on graphics (TOG), vol. 24, ACM, 2005, pp. 735–744.

[86] Matthew O ´ Toole, Felix Heide, Lei Xiao, Matthias B Hullin, Wolfgang Heidrich, and Kiriakos N Kutulakos, *Temporal frequency probing for 5d transient analysis of global light transport*, ACM Transactions on Graphics (ToG) **33** (2014), no. 4, 87.

[87] Matthew O ´ Toole, David B Lindell, and Gordon Wetzstein, *Confocal non-line-of-sight imaging based on the light-cone transform*, Nature **555** (2018), no. 7696, 338.

[88] John Oberlin and Stefanie Tellex, *Time-lapse light field photography for perceiving non-lambertian scenes.*, Robotics: Science and Systems, 2017.

[89] Kohei Okumura, Hiromasa Oku, and Masatoshi Ishikawa, *High-speed gaze controller for millisecond-order pan/tilt camera*, 2011 IEEE International Conference on Robotics and Automation, IEEE, 2011, pp. 6186–6191.

[90] Matthew O'Toole, Supreeth Achar, Srinivasa G. Narasimhan, and Kiriakos N. Kutulakos, *Homogeneous codes for energy-efficient illumination and imaging*, ACM Trans. Graph. **34** (2015), no. 4.

[91] Cagri Ozcinar, Julián Cabrera, and Aljosa Smolic, *Visual attention-aware omnidirectional video streaming using optimal tiles for virtual reality*, IEEE Journal on Emerging and Selected Topics in Circuits and Systems **9** (2019), no. 1, 217–230.

[92] Denise D. Padilla, Patrick A. Davidson Jr, Jeffrey J. Carlson, and David N. Novick, *Advancements in sensing and perception using structured lighting techniques :an LDRD final report.*, Tech. Report SAND2005-5935, 875617, September 2005.

[93] Oskar Palinko, Andrew L Kun, Alexander Shyrokov, and Peter Heeman, *Estimating cognitive load using remote eye tracking in a driving simulator*, Proceedings of the 2010 symposium on eye-tracking research & applications, ACM, 2010, pp. 141–144.

[94] Francesco Pittaluga and Sanjeev J Koppal, *Privacy preserving optics for miniature vision sensors*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 314–324.

[95] Francesco Pittaluga, Sanjeev J Koppal, Sing Bing Kang, and Sudipta N Sinha, *Revealing scenes by inverting structure from motion reconstructions*, arXiv preprint arXiv:1904.03303 (2019).

[96] Francesco Pittaluga, Zaid Tasneem, Justin Folden, Brevin Tilmon, Ayan Chakrabarti, and Sanjeev J Koppal, *Towards a mems-based adaptive lidar*, 2020 International Conference on 3D Vision (3DV), IEEE, 2020, pp. 1216–1226.

[97] Yaadhav Raaj, Siddharth Ancha, Robert Tamburo, David Held, and Srinivasa G. Narasimhan, *Exploiting and refining depth distributions with triangulation light curtains*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2021, pp. 7434–7442.

[98] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jonathon Shlens, *Stand-alone self-attention in vision models*, arXiv preprint arXiv:1906.05909 (2019).

[99] Rene Ranftl, Vibhav Vineet, Qifeng Chen, and Vladlen Koltun, *Dense monocular depth estimation in complex dynamic scenes*, Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 4058–4066.

[100] Ramesh Raskar, Amit Agrawal, and Jack Tumblin, *Coded exposure photography: motion deblurring using fluttered shutter*, ACM transactions on graphics (TOG), vol. 25, ACM, 2006, pp. 795–804.

[101] Ramesh Raskar, Greg Welch, Matt Cutts, Adam Lake, Lev Stesin, and Henry Fuchs, *The office of the future: A unified approach to image-based modeling and spatially immersive displays*, Proceedings of the 25th annual conference on Computer graphics and interactive techniques, ACM, 1998, pp. 179–188.

[102] Gernot Riegler, Matthias Rüther, and Horst Bischof, *Atgv-net: Accurate depth super-resolution*, European Conference on Computer Vision, Springer, 2016, pp. 268–284.

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

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

[105] Thilo Sandner, Claudia Baulig, Thomas Grasshoff, Michael Wildenhain, Markus Schwarzenberg, Hans-Georg Dahlmann, and Stefan Schwarzer, *Hybrid assembled micro scanner array with large aperture and their system integration for a 3d tof laser camera*, MOEMS and Miniaturized Systems XIV, vol. 9375, International Society for Optics and Photonics, 2015, p. 937505.

[106] Ashutosh Saxena, Min Sun, and Andrew Y Ng, *Make3d: Learning 3d scene structure from a single still image*, IEEE transactions on pattern analysis and machine intelligence **31** (2008), no. 5, 824–840.

[107] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus, *Indoor segmentation and support inference from rgbd images*, European conference on computer vision, Springer, 2012, pp. 746–760.

[108] Barry L Stann, Jeff F Dammann, Mark Del Giorno, Charles DiBerardino, Mark M Giza, Michael A Powers, and Nenad Uzunovic, *Integration and demonstration of mems–scanned ladar for robotic navigation*, Proc. SPIE, vol. 9084, 2014, p. 90840J.

[109] Zhanghao Sun, Ronald Quan, and Olav Solgaard, *Resonant scanning design and control for fast spatial sampling*, Scientific Reports **11** (2021), no. 1, 20011.

[110] Tao Tang, Yongmei Huang, Chengyu Fu, and Shunfa Liu, *Acceleration feedback of a ccd-based tracking loop for fast steering mirror*, Optical Engineering **48** (2009), no. 1, 013001.

[111] Zaid Tasneem, Charuvahan Adhivarahan, Dingkang Wang, Huikai Xie, Karthik Dantu, and Sanjeev J Koppal, *Adaptive fovea for scanning depth sensors*, The International Journal of Robotics Research **39** (2020), no. 7, 837–855.

[112] Lyne Tchapmi, Christopher Choy, Iro Armeni, JunYoung Gwak, and Silvio Savarese, *Segcloud: Semantic segmentation of 3d point clouds*, 2017 international conference on 3D vision (3DV), IEEE, 2017, pp. 537–547.

[113] Sebastian Thrun, Wolfram Burgard, and Dieter Fox, *Probabilistic robotics*, MIT press, 2005.

[114] B. Tilmon, E. Jain, S. Ferrari, and S. Koppal, *Fast foveating cameras for dense adaptive resolution*, IEEE Transactions on Pattern Analysis amp; Machine Intelligence **44** (2022), no. 09, 4867–4878.

[115] Brevin Tilmon, Eakta Jain, Silvia Ferrari, and Sanjeev Koppal, *Foveacam: A mems mirror-enabled foveating camera*, 2020 IEEE International Conference on Computational Photography (ICCP), IEEE, 2020, pp. 1–11.

[116] Brevin Tilmon and Sanjeev J. Koppal, *Saccadecam: Adaptive visual attention for monocular depth sensing*, Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), October 2021, pp. 6009–6018.

[117] Brevin Tilmon, Zhanghao Sun, Sanjeev J. Koppal, Yicheng Wu, Georgios Evangelidis, Ramzi Zahreddine, Gurunandan Krishnan, Sizhuo Ma, and Jian Wang, *Energy-efficient adaptive 3d sensing*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2023, pp. 5054–5063.

[118] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger, *Sparsity invariant cnns*, 2017 International Conference on 3D Vision (3DV), IEEE, 2017, pp. 11–20.

[119] Burak Uzkent and Stefano Ermon, *Learning when and where to zoom with deep reinforcement learning*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020.

[120] Burak Uzkent, Christopher Yeh, and Stefano Ermon, *Efficient object detection in large images using deep reinforcement learning*, Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), March 2020.

[121] Wouter Van Gansbeke, Davy Neven, Bert De Brabandere, and Luc Van Gool, *Sparse and noisy lidar completion with rgb guidance and uncertainty*, arXiv preprint arXiv:1902.05356 (2019).

[122] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, *Attention is all you need*, arXiv preprint arXiv:1706.03762 (2017).

[123] Andreas Velten, Di Wu, Belen Masia, Adrian Jarabo, Christopher Barsi, Chinmaya Joshi, Everett Lawson, Moungi Bawendi, Diego Gutierrez, and Ramesh Raskar, *Imaging the propagation of light through scenes at picosecond resolution*, Communications of the ACM **59** (2016), no. 9, 79–86.

[124] Michael B Wakin, Jason N Laska, Marco F Duarte, Dror Baron, Shriram Sarvotham, Dharmpal Takhar, Kevin F Kelly, and Richard G Baraniuk, *An architecture for compressive imaging*, Image Processing, 2006 IEEE International Conference on, IEEE, 2006, pp. 1273–1276.

[125] Hanxiao Wang, Venkatesh Saligrama, Stan Sclaroff, and Vitaly Ablavsky, *Cost-aware fine-grained recognition for iots based on sequential fixations*, 2018.

[126] Jian Wang, Joseph Bartels, William Whittaker, Aswin C. Sankaranarayanan, and Srinivasa G. Narasimhan, *Programmable triangulation light curtains*, Proceedings of the European Conference on Computer Vision (ECCV), September 2018.

[127] Jian Wang, Aswin C. Sankaranarayanan, Mohit Gupta, and Srinivasa G. Narasimhan, *Dual Structured Light 3D Using a 1D Sensor*, European Conference on Computer Vision (ECCV) (Cham), vol. 9910, Springer International Publishing, 2016, pp. 383–398.

[128] Yi Wang, Youlong Yang, and Xi Zhao, *Object detection using clustering algorithm adaptive searching regions in aerial images*, European Conference on Computer Vision, Springer, 2020, pp. 651–664.

[129] Hongchuan Wei, Wenjie Lu, Pingping Zhu, Guoquan Huang, John Leonard, and Silvia Ferrari, *Optimized visibility motion planning for target tracking and localization*, 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2014, pp. 76–82.

[130] Hongchuan Wei, Pingping Zhu, Miao Liu, Jonathan P How, and Silvia Ferrari, *Automatic pan–tilt camera control for learning dirichlet process gaussian process (dpgp) mixture models of multiple moving targets*, IEEE Transactions on Automatic Control **64** (2018), no. 1, 159–173.

[131] Eric W. Weisstein, *Radon transform–gaussian. From MathWorld—A Wolfram Web Resource*, Last visited on 10/8/2019.

[132] Bohan Wu, Iretiayo Akinola, Abhi Gupta, Feng Xu, Jacob Varley, David Watkins-Valls, and Peter K Allen, *Generative attention learning: a "general" framework for high-performance multi-fingered grasping in clutter*, Autonomous Robots (2020), 1–20.

[133] Yicheng Wu, Vivek Boominathan, Huaijin Chen, Aswin Sankaranarayanan, and Ashok Veeraraghavan, *Phasecam3d—learning phase masks for passive single view depth estimation*, 2019 IEEE International Conference on Computational Photography (ICCP), IEEE, 2019, pp. 1–12.

[134] Li Zhang, Brian Curless, and Steven M Seitz, *Spacetime stereo: Shape recovery for dynamic scenes*, 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings., vol. 2, IEEE, 2003, pp. II–367.

[135] Xuaner Zhang, Qifeng Chen, Ren Ng, and Vladlen Koltun, *Zoom to learn, learn to zoom*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 3762–3770.

[136] Yinda Zhang and Thomas Funkhouser, *Deep depth completion of a single rgb-d image*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 175–185.

[137] Mo Zohrabi, Robert H Cormack, and Juliet T Gopinath, *Wide-angle nonmechanical beam steering using liquid lenses*, Optics express **24** (2016), no. 21, 23798–23809.

[138] Assaf Zomet and Shree K Nayar, *Lensless imaging with a controllable aperture*, Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, vol. 1, IEEE, 2006, pp. 339–346.

BIOGRAPHICAL SKETCH

Brevin Tilmon received his Bachelor of Science in Engineering Physics with an emphasis in Electrical Engineering at Murray State University and his PhD in Electrical and Computer Engineering from the University of Florida. During his PhD he spent time with NASA, Meta, and Snap working on various computational imaging problems.