Compact and Low-Power Computational 3D Sensors for Gestural Input

by

Andrea B. Colaço

S.M., Massachusetts Institute of Technology (2010)
B.E., Birla Institute of Technology and Science, Pilani (2007)

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Media Arts and Sciences

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2014

Author_____
Program in Media Arts and Sciences
May 5, 2014

Certified by_____
Christopher M. Schmandt
Principal Research Scientist
Program in Media Arts and Sciences
Thesis Supervisor

Certified by_____
Vivek K Goyal
Assistant Professor of Electrical and Computer Engineering
Boston Univerisity
Thesis Co-Supervisor

Accepted by_____
Patricia Maes
Associate Academic Head
Program in Media Arts and Sciences

Compact and Low-Power Computational 3D Sensors for Gestural Input

by

Andrea B. Colaço

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
on May 5, 2014, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Media Arts and Sciences

Abstract

Mobile devices have evolved into powerful computing platforms. As computing capabilities grow and size shrinks, the most pronounced limitation with mobile devices is display size. With the adoption of touch as the de facto input, the mobile screen doubles as a display and an input device. Touchscreen interfaces have several limitations: the act of touching the screen occludes the display, interface elements like on-screen keyboards consume precious display real estate, and navigation through content often requires repeated actions like pinch-and-zoom. This thesis is motivated by these inherent limitations of using touch input to interact with mobile devices. Thus, the primary focus of this thesis is on using the space around the device for touchless gestural input to devices with small or no displays. Capturing gestural input in this volume requires localization of the human hand in 3D. We present a real-time system for doing so as a culmination of an exploration of novel methods for 3D capture. First, two related systems for 3D imaging are presented, both relying on modeling and algorithms from parametric sampling theory and compressed sensing. Then, a separate system for 3D localization, without full 3D imaging, is presented. This system, Mime, is built using standard, low-cost opto-electronic components — a single LED and three baseline separated photodiodes. We demonstrate fast and accurate 3D motion tracking at low power enabled by parametric scene response modeling. We combine this low-power 3D tracking with RGB image-based computer vision algorithms for finer gestural control. We demonstrate a variety of application scenarios developed using our sensor, including 3D spatial input using close-range gestures, gaming, on-the-move interaction, and operation in cluttered environments and in broad daylight conditions.

Thesis Supervisor: Christopher M. Schmandt
Title: Principal Research Scientist, Program in Media Arts and Sciences

Thesis Co-Supervisor: Vivek K Goyal
Title: Assistant Professor of Electrical and Computer Engineering, Boston Univerisity

Compact and Low-Power Computational 3D Sensors for Gestural Input

by

Andrea B. Colaço

The following people served as readers for this thesis:

Thesis Reader_____

Joseph A. Paradiso
Associate Professor of Media Arts and Sciences
Program in Media Arts and Sciences

Acknowledgements

I had an incredible experience during my doctoral studies at MIT. At times, close to six years in graduate school felt like a lifetime. But as I wrote this thesis, it all seems to have happened too quickly. Getting to this point has been made possible by many wonderful and talented people around me and I would like to take a few lines to express my sincere gratitude to them.

To my advisor Chris Schmandt for giving me an opportunity to attend graduate school at MIT by accepting me to his group twice – in 2008 for the Master program and then again in 2010 for doctoral studies. I am grateful to him for his support, encouragement and openness to new areas of research; but most importantly for the confidence he has in his students. To Vivek Goyal for taking a chance at collaborating with me when I was very early in the PhD, and for co-supervising my thesis. I would like to thank him for the time he spent providing feedback on my work and for imparting his rigor in technical writing and presenting. To Joseph Paradiso, for the inspiration his work and classes have been. I have been fortunate to have worked with him as part of the Design and Innovation workshops in India where I was inspired by his engagement with problems that are socially and economically unique to India.

To my colleagues at the Speech+Mobility group at the Media Lab for their support and for contributing to many good times. Special thanks to Jaewoo Chung, Drew Harry, Charlie DeTar, Misha Sra, Sujoy Chowdhury and Cindy Kao.

To my collaborators for their great spirit toward team work – Ahmed Kirmani, Hye Soo Yang, Dongeek Shin, Dr. Franco Wong and Dheera Venkatraman. I had a multiplicative learning factor due to the orthogonal expertise they brought to the team effort.

To Professors Jeffrey Shapiro and Alan Oppenheim whose classes and first-principle approach to problem formulation and solving have made a lasting impact on how I look at new problems.

To the larger MIT community for opportunities to explore entrepreneurship.

Finally, to my parents and brother for believing in my dreams, and to the memory of my grandmother, Lavita Braz, who was the bravest person I knew.

Contents

Chapter 1

Introduction

Anyone who has used mobile devices understands the broad set of functions they support. Their ubiquity makes them ideal for communication, scheduling, entertainment, data capture, and retrieval on the go. Early mobile devices were characterized by tangible keypads and small displays. However, their form factor has been in a state of constant flux in an attempt to support their expanding functionality. They are now available in the form of wrist-worn and head-mounted devices in an effort to make interaction more seamless. Simultaneously, interaction has also evolved to more natural forms like voice and touch. Additionally, mobile devices have extended our perception of the physical world through virtual and visible augmented reality.

We are embracing the digital future of ubiquitous computing envisioned by Mark Weiser [1], "The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it." Toward this vision we continue to explore natural input, output and feedback mechanisms for our mobile and wearable devices. Currently, smart phones and tablets have a distinctive flat touch-based interactive glass screen. These devices satisfy to a large extent our always-on computing requirements on the go. The natural next step is to think about truly weaving input, output and feedback subtly. This vision for ubiquitous human computer interaction has recently surfaced again, this time with higher fidelity head-mounted wearable displays and smart

watches with a smart phone or tablet providing the bulk of the computing capabilities. For a seamless experience between external computing and new form-factors of wearables, we are tasked with transitioning input from the domain of well understood keyboard, mouse and touch to meaningful synergetic forms for wearables.

Currently, touch-screen input is the primary interaction modality for smart devices which require a display - no matter how small. For wearables, such as head-mounted displays, voice is the input of choice; these upcoming devices do not have an touch-screen display which can double as an input device. Touch input on high fidelity displays elegantly merges input, output and visual feedback at the point of contact. However, flat screen touch interfaces do not fully take advantage of human dexterity and have their own set of limitations:

- the act of touching requires the user to be in constant contact with the device;
- touching the screen for input occludes the display;
- even simple tasks like menu navigation require tedious, repetitive actions; and
- accurate target acquisition is a challenge when the surface area of a finger is larger than supported (aptly described as the fat finger problem).

When touch surface shrinks, equipping users with better input tools for more complex and visually demanding tasks will be important in enabling new applications and making the interaction experience of the unit more intuitive and efficient.

In this thesis, we focus on capturing input around the mobile or wearable device. The space around the device is typically free, uncluttered and close enough to the device to provide line of sight if required by any sensing system. We focus on hand gesture input in the proximal space around the device as a natural way of moving beyond touch to free-space. By sensing off-screen input through gestures, this approach conserves display space that would ordinarily get consumed by touch input itself.

1.1 Around-device input

The use of gestures for human-computer interaction is an active research area. From a user experience viewpoint, gestural control using 3D cameras has been demonstrated to be an intuitive, robust, and widely-popular input mechanism in gaming applications.[1] New input technologies, like the Leap Motion Controller [2] and compact time-of-flight (TOF) cameras [3, 4], are still being explored for gesture-controlled interfaces in the context of personal computing spaces. Recent user studies have demonstrated that 3D gesture input is at least as effective as touch input for mobile devices [5]. In addition to input implications for smartphones, this finding raises interesting possibilities for smart wearables, like head-mounted displays (HMD), which lack a dominant input interface like touch.

Around-device interaction can be captured in different physical spaces around the mobile device. This could be the space in the immediate proximity of the device or input locations that are not within line-of-sight but create opportunities for around device-like input and interaction. Broadly, we categorize these interactions spaces as three different regions. This categorization of previously explored around-device interaction methods is presented in Fig. 1-1.

1.1.1 On-body and close-to-body interaction

An opportunistic space in the immediate proximity of personal devices is the body surface itself, which can often be available for input. The gesture pendant [6] is a very early example of gesture recognition in the space around a wearable device. More recently, researchers have examined the design of on-body and close-to-body gestural interfaces using optical imaging sensors 2D/3D cameras mounted in different regions close to the user; Omnitouch [7] mounts the 3D camera on the shoulder of the user, and Shoesense [8] mounts a 3D camera on a user's shoe facing up. Often these interactions necessitate appropriating the body surface for displaying interface elements. In the above examples as well as in examples like Wear-ur-world (WUW) [9] and LightGuide [10], body worn projectors and projectors mounted in the

[1] Microsoft Kinect Sensor. www.xbox.com/en-US/kinect

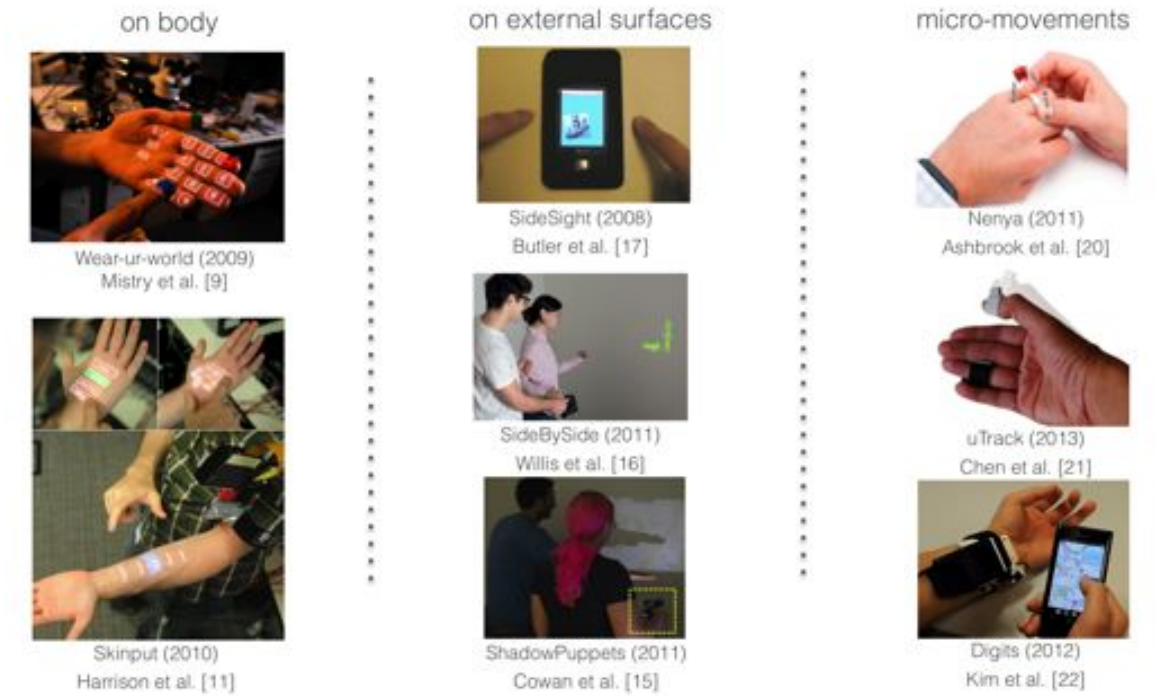


Figure 1-1: Categories of around-device interaction. On and close-to body interactions that use the body surface for a combination of input and visual feedback (left). On external surface interaction which uses surfaces like table-tops and walls around the device (middle). Micro-movements that off-load interaction without requiring much visual distraction (right).

environment respectively provide visual cues to the user. The user primarily manipulates these projected elements through the use of hand movements and gestures. Capturing hand movements or touch-like finger input on the body surface extend to non-optical sensing systems. For example, Skinput [11] senses these signals directly through bio-acoustic sensing of muscle movements. On-body interfaces have also been implemented through distributed sensors on clothing – as seen in Second Skin [12] – which capture body motion with actuated feedback for motor learning. Another class of examples [13,14] instrument the environment with motion capture systems such as the Vicon to create applications that track free-form hand movement, position, and orientation. Imaginary interfaces [13] uses close-to-body hands movements without the use of visual or haptic feedback with the premise that part of the interface resides in the user's imagination. This interaction technique offloads visual feedback requirements to the user's spatial memory. In a related piece, Chen et al. [14]

explored the use of spatial memory by using locations around the body to trigger different input actions. Their work shows proximal interactions through body shortcuts.

1.1.2 Proximal surfaces for around-device interaction

The space around the user often has surfaces that may be available for capturing free-form hand gestures as input without using traditional interaction tools like touch, styli, keyboards and mice. Typically, these include surfaces like walls, tabletops and sometimes physical objects surrounding the user. Often such surfaces are uncluttered or ideal for feedback using projection mechanisms to present visual information. Input is then captured through hand movements in front of such projections. An illustrative example in this case is ShadowPuppets [15], in which the mobile screen interface is projected on to a surface around the user; the user then interacts through hand gestures. Placing the hand in front of the projector casts a shadow of the gesture on the surface, which is then captured by an optical imager. The detected gesture manipulates the projected interface. Another similar interaction scenario for multiple users is demonstrated in SideBySide [16]. Opportunistically projecting on objects and surfaces has the advantage of being able to provide a shared visual space for multiple users. Projecting on everyday real-world objects was also explored in [9]. Horizontal surfaces are also ideal for ADI while using the mobile device screen as output. Hand movements, like tapping on the surface, are captured as input, and relevant output is displayed on the mobile screen. SideSight [17], for example, captures input on both sides of the mobile device using infrared (IR) proximity sensors and tracks multiple fingers. The system comprises pairs of IR emitters and photodiodes that detect the presence of fingers. The use of a 1-D array of these emitter-receiver pairs offers coarse spatial (1-D) granularity of input as well as depth (distance to the array or proximity). Feedback to the user is provided through the display of the mobile device and is application dependent. A related example using a similar sensing approach is seen in HoverFlow [18], where ADI is explored above the device.

1.1.3 Micro-movements

Remote input through subtle finger and wrist movements to a device is another way of off-loading interactions from small touch screen displays. This style of interaction requires wrist worn sensors instead of sensors mounted on the mobile device itself. Finger-worn rings demonstrated by Bainbridge and Paradiso in [19], the system Nenya [20] and uTrack [21] provide input mechanisms without requiring visual attention in the direction of input. Both these interaction techniques are enabled by magnetic field sensing to determine finger movements. uTrack provides mouse pointer-like movement just by minor finger movements, while Nenya maps rotation of the ring itself to navigate interface elements. In another hand-worn implementation, the authors of Digits [22] propose using all fingers in a single-handed interaction to control interface elements without distraction of visual attention; a similar system has been recently implemented and evaluated by Way and Paradiso in [23]. While the above interactions are subtle and require minimal visual attention, they require instrumenting the user with sensors, which could possibly encumber the experience.

1.2 Technical challenges and sensing constraints

The review of around-device interaction in the previous section reveals important benefits of extending input space for devices constrained by touch display size. The space around the device is unused and often unoccluded. Close-range 3D gesture sensing introduces a new interaction paradigm that goes beyond touch and alleviates its current limitations. Examples in our review reveal new interaction techniques that may be adapted to emerging wearable devices. However, the review also points to existing challenges with sensing such input. The implementation of 3D gestural input requires integration of depth sensors in mobile and wearable devices. However, existing state-of-the-art 3D sensors cannot be embedded in mobile platforms because of their prohibitive power requirements, bulky form factor, and hardware footprint. Wearing additional sensors on the body possibly creates user encumberance while sensing input. These challenges demonstrate the need for unencumbered 3D gesture sensing in scenarios where the user is mobile. The limitations of conventional input

technologies warrant development of new sensors for intuitive and effective free-form gestural input, corroborating our vision for building sensors that can eventually be embedded in mobile and wearable devices.

Design considerations: An input technology intended for around-device input and interaction with mobile or wearable devices should ideally possess the following characteristics:

- **Technical**: High accuracy, low power, low latency, small size, daylight insensitivity, and robust performance in cluttered, noisy and fast-changing environments.
- **User experience**: Interacting with the mobile device should be intuitive and should not induce fatigue upon prolonged use. The input device must be able to support both motion- and position-controlled gestures in 2D and 3D.
- **User convenience**: The sensor should be embedded within the mobile device to enable unencumbered user interaction. The user should not be required to wear markers [24] or external sensors like wrist worn trackers [22, 25] or carry additional touch pads.

1.3 Thesis theme and outline

This thesis presents research on new 3D capture techniques that enable around-device input for mobile and wearable devices, satisfying the design considerations outlined in the previous section. The goal of this work is to capture the richness of the three-dimensional world through novel computational 3D sensing frameworks that could lead to new input and interaction opportunities. The process involved exploring, designing and building several different measurement and processing frameworks in an effort to resolve technical constraints of existing 3D sensors – power, form-factor (size), computation requirements, and performance in diverse environmental conditions – by exploiting the physics of light transport. The latter part of the thesis focused on integrating these sensing approaches with mobile devices like smartphones and head-mounted displays. Fig. 1-2 shows a snapshot of the overarching research goal and different approaches covered.

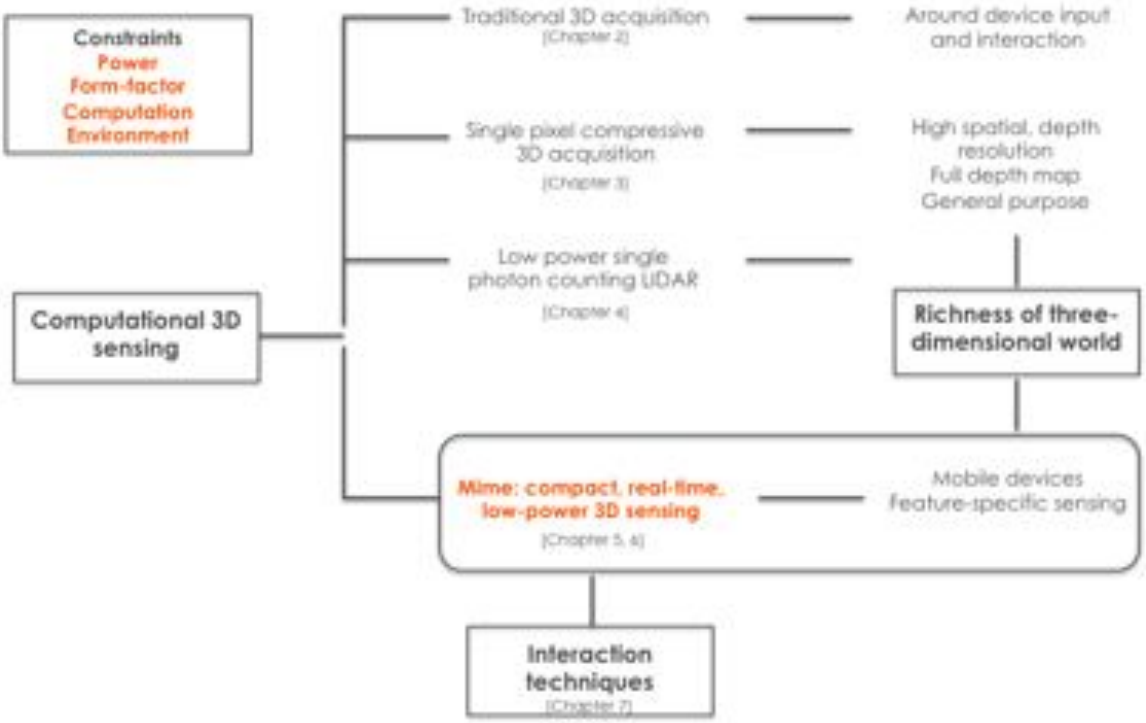


Figure 1-2: Thesis theme and outline. This thesis has focused on capturing the richness of our 3D world through the use of computational 3D sensing to create new interaction techniques with mobile devices. We have explored new 3D acquisition techniques to meet the constraints mobile devices present. The main focus of this dissertation is Mime, a compact, low-power 3D sensor and its real-time use in applications for mobile devices.

Chapter 2 reviews existing sensing techniques. The rest of this thesis then describes each of the novel imaging techniques in detail. Our first exploration involved reducing the size of 3D sensors to a single pixel. To achieve this, we introduce the framework for compressively acquiring 3D information using spatio-temporal modulation of active illumination discussed in Chapter 3 and [26]. Next, we investigated lowering power requirements for active 3D imagers, resulting in the methods in Chapter 4 and [27]. Here, we combined savings in size from the compressive depth acquisition framework with a low-power implementation enabled by a very sensitive single-photon avalanche photodiode (SPAD). However, these techniques do not address mobile-friendly computation requirements. Additionally, for tractable computation the overall system bandwidth of the imaging components should be within mobile processor capabilities. This makes the acquisition problem even more challenging because

the choice of sensing elements and illumination sources is limited to low-bandwidth components. The advantage of low bandwidth components is their low cost and ease of fabrication. For the purpose of the thesis, we focus only on addressing mobile technical constraints. The culmination of our 3D acquisition frameworks is a low-power, compact, precise and real-time operating sensor, Mime, which is the main contribution of the thesis, presented in Chapter 5 and [28]. This sensor was designed to resolve constraints that previous systems did not address while benefitting from the trade-off insights observed through the design and experiments with previous systems. Chapter 6 and [29] present theoretical extensions of the sensing framework for acquiring multiple hands as well as generic planar surfaces. Chapter 7 focuses on designing and building new interaction and input experiences with the real-time operating Mime sensor. We present two specific styles of user scenarios, one with the device mounted on a head-mounted display (in our case Google Glass[2]) and the other with a device mounted on a smartphone which makes the surface next to the display an immediate input canvas. We discuss the trade-offs and implications of the acquisition techniques presented in this thesis in Chapter 8 and point to future extensions.

[2]Google Glass. www.google.com/glass

Chapter 2

Background

This chapter reviews prior work on optical 3D acquisition techniques that this thesis builds upon. We discuss application-specific sensing, in our case human hand gesture sensing and introduce alternate methods that capture hand gestures without traditional imaging.

Three main technical areas are central to understanding the computational sensors presented in this thesis:

- Time-of-flight (TOF) based active 3D sensing
- Compressed sensing
- Parametric optical signal processing

These areas will be discussed in Sections 2.1, 2.2 and 2.3. Through the use of the computational sensors presented in this thesis, we are interested in enabling gesture-based around device input for mobile devices – handheld smart phones and wearables; related systems for gesture capture are discussed in Section 2.4.

2.1 Time of flight principles for active 3D sensing

Sensing 3D scene structure is an integral part of applications ranging from 3D microscopy [30] to geographical surveying [31]. While 2D imaging is a mature technology, 3D acquisition techniques have room for significant improvements in spatial resolution, range accuracy, and cost effectiveness. Humans use both monocular cues – such as motion parallax – and binocular cues – such as stereo disparity – to perceive depth, but camera-based stereo vision techniques [32] suffer from poor range resolution and high sensitivity to noise [33, 34]. Computer vision techniques – including structured-light scanning, depth-from-focus, depth-from-shape, and depth-from-motion [32, 35, 36] – are computation intensive, and the range output from these methods is highly prone to errors from miscalibration, absence of sufficient scene texture, and low signal-to-noise ratio (SNR) [33, 34, 36].

In comparison, active range acquisition systems such as LIDAR systems [37] and TOF cameras [38, 39] are more robust against noise [34], work in real-time at video frame rates, and acquire range information from a single viewpoint with little dependence on scene reflectance or texture. Both LIDAR and TOF cameras operate by measuring the time difference of arrival between a transmitted pulse and the scene reflection. LIDAR systems consist of a pulsed illumination source such as a laser, a mechanical 2D laser scanning unit, and a single time-resolved photodetector or avalanche photodiode [37, 40, 41]. The TOF camera illumination unit is composed of an array of omnidirectional, modulated, infrared light emitting diodes (LEDs) [38,39,42]. The reflected light from the scene – with time delay proportional to distance – is focused at a 2D array of TOF range sensing pixels. Localization in 3D by the TOF principle could also be performed without optics by exploiting ultra-wide band (UWB) impulse radios which detect TOF of reflected RF impulses [43]. The resolution provided by such systems often tends to be the limiting factor in their adoption. A major shortcoming of LIDAR systems and TOF cameras is low spatial resolution, or the inability to resolve sharp spatial features in the scene. For real-time operability LIDAR devices have low 2D scanning resolution. Similarly, due to limitations in the 2D TOF sensor array fabrication process and readout rates, the number of pixels in TOF camera sensors is typically lower than most 2D RGB cameras [42]. Consequently, it is desirable to develop

novel, real-time range sensors that possess high spatial resolution without increasing the device cost and complexity.

2.2 Sparsity and compressed sensing

Many natural signals can be represented or approximated well using a small number of nonzero parameters. This property is known as sparsity and has been widely exploited for signal estimation and compression [44]. Making changes in signal acquisition architectures – often including some form of randomization – inspired by the ability to effectively exploit sparsity in estimation has been termed compressed sensing (CS). CS provides techniques to estimate a signal vector x from linear measurements of the form $y = Ax + w$, where w is additive noise and vector y has *fewer* entries than x. The estimation methods exploit that there is a linear transformation T such that Tx is approximately sparse. An early instantiation of CS in an imaging context was the "single-pixel camera" [45, 46] which demonstrated the use of pseudorandom binary spatial light modulator (SLM) configurations for acquiring spatial information and exploitated transform-domain sparsity.

The depth map of a scene is generally more compressible or sparse than the reflectance or texture (see Fig. 2-1). Thus, we expect a smaller number of measurements to suffice; this is indeed the case, as our number of measurements is 1 to 5% of the number of pixels as compared to 10 to 40% for reflectance imaging [45, 46].

2.2.1 Challenges in exploiting sparsity in range acquisition

In TOF systems, depths are revealed through phase offsets between the illumination signal and the reflected light rather than by direct measurement of time delays. These measurements are made either by raster scanning every point of interest in the field of view or establishing a correspondence between each spatial point and an array of sensors. Compressively acquiring range information using only a single detector poses two major challenges. First, the quantity of interest – depth – is embedded in the reflected signal as a time shift.

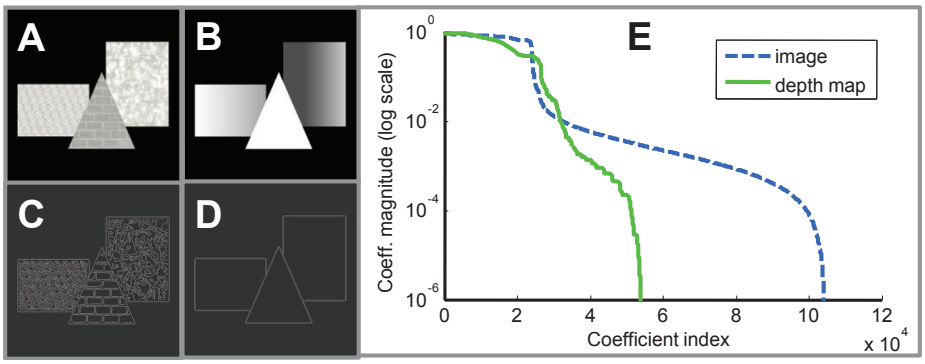


Figure 2-1: *Sparsity* of a signal (having a basis expansion or similar representation with a small number of coefficients significantly different from zero) is widely exploited for signal estimation and compression [44]. An $N \times N$-pixel digital photograph (A) or depth map (B) of a scene requires N^2 pixel values for representation in the spatial domain. As illustrated with the output of an edge-detection method, the Laplacian of a depth map (D) typically has fewer significant coefficients than the Laplacian of a photograph (C). This structure of natural scenes is also reflected in discrete wavelet transform (DWT) coefficients sorted by magnitude: a photograph has slower decay of DWT coefficients and more nonzero coefficients (E: blue, dashed) than the corresponding depth map (E: green, solid). We exploit this simplicity of depth maps in our range acquisition framework.

The measured signal at the detector is a sum of all reflected returns and hence does not directly measure this time shift. This nonlinearity worsens when there are multiple time shifts in the returned signal corresponding to the presence of many depths. Varying the SLM configuration would produce different nonlinear mixtures of depths and thus could make the solution unique, but the complexity stemming from nonlinearity of mixing remains. The second challenge comes from the fact that a single detector loses all directionality information about the reflected signals; this is present in reflectance imaging as well.

2.2.2 Compressive LIDAR

In a preliminary application of the CS framework to range acquisition in LIDAR systems [47], 2 ns square pulses from a function generator drive a 780 nm laser diode to illuminate a scene. Reflected light is focused onto a digital micromirror device (DMD) that implements pseudorandomly-chosen binary patterns. Light from the sensing patterns is received at a photon-counting detector and gated to collect photons arriving from an *a priori*

chosen range interval, and then conventional CS reconstruction is applied to recover an image of the objects within the selected depth interval. The use of impulsive illumination and range gating make this a conventional CS problem in that the quantities of interest (reflectances as a function of spatial position, within a depth range) are combined linearly in the measurements. Hence, while this approach unmixes spatial correspondences it does not directly solve the aforementioned challenge of resolving nonlinearly-embedded depth information. The need for accurate range intervals of interest prior to reconstruction is one of the major disadvantages of this system. It also follows that there is no method to distinguish between objects at different depths within a chosen range interval. Moreover, acquiring a complete scene depth map requires a full range sweep. The proof-of-concept system [47] has 60 cm range resolution and 64×64 pixel resolution.

2.3 Parametric optical signal processing

Analog signals in the real world typically contain rich information and have variable bandwidths. Utilization of these signals routinely requires access to digital versions of these signals without losing fidelity during the conversion. Classical sampling theory necessitates sampling above the Nyquist rate. For a signal whose maximum frequency is f_{max}, this rate would be $2f_{max}$. Sampling above the Nyquist rate ensures that the original analog signal is preserved and usable. For higher bandwidth signals, sampling at the Nyquist rate would require very fast sampling.

However, some classes of high bandwidth signals that appear in several practical applications, can be completely described by a small number of parameters. Examples include short pulse trains in medical imaging, radar and ultra-wideband systems. In the pulse train example, these parameters would be the time-delay between pulses and their amplitudes. These two parameters completely describe this example signal of interest. Even though the Fourier bandwidth of an impulse function (the closest representation of a very short pulse) is infinite, we can see how it may be wasteful to sample this pulse train at its Nyquist rate if it can be uniquely described by only two parameters. Of course, we would need to

recover these two parameters for each pulse in the train. This brings us to an important condition – within any finite time segment of length τ, the signal is completely described by no more than K parameters. This is the local rate of innovation of the signal which is no more than K degrees of freedom every τ seconds. A signal is said to have a finite rate of innovation (FRI) if it can be described by a finite or small number of parameters per unit time [48]. The recently proposed finite rate innovation (FRI) framework solves the problem of recovering these signal parameters from discrete samples of a filtered version of the signal of interest, as long as the number of samples is at least twice the number of parameters. This proves to be a powerful tool to recover a relatively high bandwidth signal at a much lower sampling rate.

The problem of recovering time-delays corresponding to depth of scene objects in the time-of-flight setup is analogous to the pulse train example we just discussed. Further, we also know that depth is a naturally sparse signal which supplements the requirement of a small number of parameters in the reflected signal of interest. This problem was not previously explored in the context of optical signals. Most time-of-flight sensors are inherently bandlimited. The sensor thus provides access to a lowpass filtered version of the time-of-flight pulses.

2.4 Optical imaging systems for capturing hand gestures

Sensing human hand gestures requires some form of sensing the human hand. For the purpose of this thesis, we define hand gestures as coordinated spatial and temporal movements that are either captured at a single point in time or over a period of time. Therefore, hand gesture capture could require capturing hand movement independent of shape or hand motion along with additional information such as pose. This thesis considers hand motion with and without shape dependence. In this chapter, we will discuss techniques, sensors and systems designed to recover the continuum from rigid hand motion to hand shape and motion coordinated over time. From a design perspective, only motion-based gesture activation has a broader set solutions that could reside outside of line of sight of target hand; in this

category we discuss sensing systems either present in the environment or mounted on the target, such as magnetic field based sensing, electric field based sensing, ultrasound based sensing as well as bio-acoustic signal sensing of the target hand. Shape recovery however requires direct line of sight with the target hand. Typically, optical imaging is a popular choice for recovering hand shape together with motion.

In this section, we review techniques sensing hand motion and shape. We will highlight key considerations which determine suitability of these techniques to mobile device constraints. Our goal through this exposition is to identify how the theoretical framework of signal modeling and the practical considerations in the implementation of Mime overcome challenges in previously developed systems.

The three main techniques for depth sensing are stereo disparity-based methods from computer vision [36], and active illumination techniques which are further categorized as near infrared (NIR) intensity-based methods, and sensors that operate on the TOF principle. Stereo disparity-based techniques are passive in the sense that they use natural light or ambient illumination. On the other hand, active optical methods use specialized NIR light sources for scene illumination. Compared with passive stereo vision, active optical methods have proven to be more robust and reliable for a variety of industrial, consumer and scientific applications [34], [49], [50]. Here we present a short survey of of these techniques and compare them with Mime on the aforementioned performance metrics and resource constraints.

2.4.1 Gestural control with 2D cameras

Computer vision techniques allow the use of embedded 2D cameras to recognize hand gestures [51]. These gestures have been widely used for unencumbered line-of-sight interaction with mobile devices. Standard RGB image-based gesture recognition suffers from several technical problems: it has poor resolution and is not robust in cluttered environments, it is computationally complex for mobile processors, and it supports only a small dictionary of simple motion-cued gestures like waving. RGB image-based gesture recognition can be

made more precise, robust and generic at the expense of using of additional elements like color markers or infrared trackers [6].

2.4.2 NIR Intensity Based Techniques

The three main NIR image intensity based techniques are active stereo vision, speckle pattern decorrelation, and infrared proximity array sensing. All of these methods require illuminating the scene with an always *on* light pattern which is the major source of power consumption in these sensors.

Active stereo vision involves illuminating the scene with a structured light pattern and imaging with two baseline-separated cameras [52]. The illumination patterns provides rich scene texture which is necessary to create a dense stereo-correspondence and thereby depth maps that do not contain artifacts such as holes in scene regions with insufficient texture – a common problem with passive stereo disparity imaging.

Speckle pattern decorrelation uses a laser source to project a speckle or dot pattern on the scene. The dot pattern on the scene is imaged using a single NIR camera. The capture images is processed using a local cross-correlation method called region-growing random dot matching algorithm to detect and assign depth labels to the laser speckles [53].

Infrared Proximity Array (IPA) sensors operate on the fact that closer objects appear brighter under active illumination. The scene is illuminated using two light sources with different characteristics (intensity and half-angle) and two separate NIR intensity images are recorded. Scene depth at every pixel is computed using a pre-computed polynomial regression curve that relates the intensity ratio between the two NIR images with scene depth point [54].

Despite the advantages that active NIR intensity based method have over conventional stereo vision in imaging quality and depth resolution, they require high optical output because the light source is always *on* and it needs to be stronger than ambient illumination in order for the technique to be effective. As a result active NIR intensity based depth

sensors are also sensitive to ambient light. Also their depth accuracy and range resolution performance are baseline limited. Moreover, these sensors are found to perform poorly at close working ranges of under 1 meter.

2.4.3 TOF Based Techniques

TOF depth sensors operate by flood illuminating the scene using an intensity modulated light source and focusing the reflected light on to a 2D array of special sensors which computes the distance to a scene point by measuring the time delay or phase shift caused due to the back and forth propagation of light. There are three major types of TOF depth sensors, amplitude modulated cosine wave (AMCW), light detection and ranging (LIDAR) systems and short pulsed TOF systems with time-gated sensors.

AMCW TOF cameras flood illuminate the scene with an omnidirectional NIR light pulse. Typically, the transmitted pulse is a modulated square wave with a pulse repetition frequency (PRF) of 10 MHz or higher and a 50% duty cycle. This implies that the light source is on half the time. The reflected light is converted to an electrical signal by the CMOS sensor pixels which is then correlated with a cosine wave of the same frequency as the modulation frequency of the light source [38]. Four samples of the cross correlation function are used to compute the amplitude, background intensity and the phase shift of the received waveform relative to the transmitted pulse. This phase shift measured at each pixel is directly proportional to scene depth.

Time-gated TOF systems also operate using flood NIR illumination but using a much shorter pulse (50% duty cycle at a high PRF of 50 MHz) [55], [56]. The image sensor has a fast gating mechanism based on an electro-optic Gallium-Arsenide shutter. The amount of pulse signal collected at pixel corresponds to where within the depth range the pulse was reflected from, and can thus be used to calculate the distance to a corresponding point on the captured subject.

LIDAR systems measure scene distance by raster scanning the scene with a short pulsed laser source [37]. The pulse widths are typically in the sub-nanosecond range with a PRF

of $1-2$ MHz. The sensor is single photon counting detector which time-stamps the arrival of every detected photon with picosecond accuracy. For every raster position, the photons are collected over a small time-interval. The peak position of the histogram computed using the photon arrival times is the depth estimate.

TOF depth sensors offer several advantages over active NIR intensity based techniques for depth measurement. They have higher depth accuracy and range resolution, and operate from a single viewpoint with no baseline restrictions. AMCW TOF cameras have ambient light rejection mechanisms [4] making them less sensitive to ambient light. Despite the improvement in imaging quality by TOF cameras, they come with their own set of limitations. Due to the fast timing requirements, TOF sensors use custom hardware for illumination and sensing which is often difficult to manufacture and leads to expensive cost. The spatial or lateral resolution of TOF cameras is lower than that of NIR intensity based depth sensors [26]. In contrast, NIR intensity based depth cameras use standard CMOS arrays and NIR light sources for scene illumination and also are priced at a significantly lower cost. TOF depth cameras also have to account for additional imaging artifacts like distance aliasing or phase wrapping, and multipath distortion which are absent in NIR intensity based cameras.

2.5 Non-camera based techniques for capturing hand gestures

In this section we review motion-based gestural control enabled by techniques that do not use any image formation. Non-camera based techniques offer advantages in power consumption and distance of operation. However, these techniques typically do not disambiguate hand shapes and often provide lower accuracy tracking compared with their optical imaging counterparts.

2.5.1 Sound based 3D Source Localization

In addition to optical methods for distance sensing, ultrasound based transducers are also frequently used in robotics for 3D object localization [57]. More recently, they have been incorporated into human computer interfaces for hand and gesture tracking [58], [59]. A typical setup involves transmitting an omni-directional ultrasound beam towards the scene and using a linear array of microphones to accomplish 3D source localization using beam-forming and time-delay of arrival estimation [57]. Another implementation Soundwave [60] uses the speaker and microphone that are already present in most devices to sense in-air gestures around such devices. The system generates an inaudible tone and uses the doppler (frequency) shift that is produced when the tone gets reflected by moving objects such as hands.

State-of-the-art ultrasound systems are less precise at a centimeter or worse range resolution, compared with their optical counterparts which offer sub-centimeter depth resolution. Also ultrasound transducers do not produce full scene depth maps and perform poorly in localizing multiple objects. Their performance also degrades significantly in cluttered natural scenes and reverberant environments due to strong multiple scattering of sound waves. In contrast, light undergoes diffuse scattering in most natural scenes which does not mar the imaging quality of active optical depth sensors as much.

2.5.2 Magnetic field based sensing

Sensing motion based on movement of a magnet-marker has been implemented in several different forms. Positioning systems like the Polhemus tracker, track object movement very accurately (mm accuracy) within the region covered by the magnetic field generated by the system. The disadvantage of such a system is the requirement of bulky hardware installation. For mobile devices, this problem is resolved with the use of the 3-axis compass which most smart phones carry. This compass is relatively weaker but has been used in several implementations of magnetic field based position and motion tracking. Magnetic field based positioning systems require an external magnet mounted on the target of interest.

The movement of the magnet result in measurable magnetic field changes along each axis of the magnetometer or compass (in the case of smart phones). By recording the relative strengths of the magnetic field lines of force along 3 axes, the system can resolve the 3D position of the target (which is retrofitted with a magnet). An advantage of measuring the magnetic field for computing 3D position is that the technique does not require the target to be in direct line of sight of the sensor because the magnetic field can pass through several different materials unlike optical signals. However, the volume of operation depends on the strength of the magnets and the sensitivity of the sensors – strength of the field required also determines the size of the magnets used. Consequently, accuracy depends on the sensitivity of the system and the size of the sensors. Moreover, tracking is possible only with the presence of magnet on the target of interest which could be encumbering in some cases.

Abracadabra [61] couples a wrist-mounted magnetometer with a passive magnet worn on a finger of the opposite hand – this allows 3-dimensional input through the finger. Ashbrook et al. [20] built Nenya, a finger-ring input device that uses magnetic tracking performed by a wrist-worn sensor. More recently, implementations that use the compass on the smart phone have demonstrated finger and hand motion tracking for interaction with the device that uses the space around the device. The positioning system described in [62] uses a permanent magnet (rod, pen or ring) held or worn by the user. In the uTrack [21] system, the user wears a ring with a small magnet. The region of interaction is confined to very fine movements at the user's fingertips.

2.5.3 Electric field sensing

This technique measures the disturbance caused by a target hand of an electric field applied between transmitter and receiver electrodes. In its most simple form, the transmitter is driven by a low-voltage radio source, typically at low-frequency. The target hand enters the electric field between the transmitter and receiver, the capacitive coupling from the transmitter into the hand, the target hand into the receiver, and body into ground changes the signal detected at the receiver. Improving resolution and complexity of sensing

is achieved by increasing the number of transmitters and receivers and additionally multiplexing transmit and receive functions. Implementations of multiple sensing configurations and exhaustive performance details are outlined in [63].

This type of sensing is limited by the strength of the field and number of transmitter/receiver pairs required for finer resolution. The volume of interaction improves with size of electrodes and separation between transmit/receive electrodes.

2.5.4 Capacitive sensing

This method exploits the simple charge holding capacity of the plates of capacitors as a function of the distance and dielectric medium between them. To measure displacement and hence position and motion of a target human hand, the hand itself acts as one of plates, the other is typically a metal plate. The distance of the hand to the metal plate results in a variable resonance frequency in the resonant circuit attached to the metal plate. The human hand is a good candidate for such sensing because of its conducting properties and relatively high dielectric constant.

One implementation of the system described above is seen in the system Thracker [64]. The system was designed to capture hand motion and basic gestures in front of displays. The sensing system comprised four metal plates – one at each corner of the display – to improve precision and capture multiple moving targets, for example, two hands.

This technique is relatively low-cost and small in size. However, it is not ideal for tracking beyond a few centimeters. For longer-range operation, the system requires the use of larger metal plates. Additionally, tracking resolution and sensitivity quickly drop with distance from the sensor. Sensing is restricted to objects that are conductive. While it is still ideal for tracking human hands at very close proximity, this requirement precludes tracking of any other non-conductive target. Further, the presence of conductive material other than the target of interest interferes with the sensing.

2.5.5 Wi-fi based sensing

Similar to ultrasound localization systems, there have been several examples of using wireless signals to localize human targets and gestures. The more popular approaches use the received signal strength indicator (RSSI), from multiple antennas to localize the target. Newer approaches like [65], [66] use the Doppler shift measured in the received signal induced by target motion. These implementations use existing wireless infrastructure (routers with multiple antennas) to perform both localization and motion recognition.

The use of wireless signals makes this approach useful in non line-of-sight configurations and for sensing motion based gestures through walls. However, the granularity of resolution obtained is on the scale of the moving target, that is, such a system is ideal for detecting larger hand movements.

Chapter 3

Compressive Depth Acquisition Camera

Three dimensional sensors and sensing frameworks provide opportunities for optimization at different steps in the acquisition and processing pipeline. These are broadly represented as trade-offs between active optical imaging power, sensor size, and spatial and temporal resolution capabilities. Applying computational techniques often has the effect of scaling down sensor size and optical power while maintaining or improving resolution. In time-of-flight based 3D acquisition, the limiting factor is predominantly sensor array fabrication. However, TOF based systems tend to provide higher accuracy. Consequently, maintaining accuracy while reducing sensor array size is an advantage to such systems. This chapter introduces a new computational technique for recovery of depth by the time-of-flight principle using only a single time-resolved sensor [26]. This is enabled by accurate impulse response modeling of natural scenes.

Natural scenes can often be approximated by planar facets. Here, we introduce a framework for acquiring the depth map of a piecewise-planar scene at high range and spatial resolution using only a single photodetector as the sensing element and a spatiotemporally-modulated light source as the illumination unit. In our framework (see Fig. 3-1), an omnidirectional, temporally-modulated periodic light source illuminates a spatial light modulator (SLM)

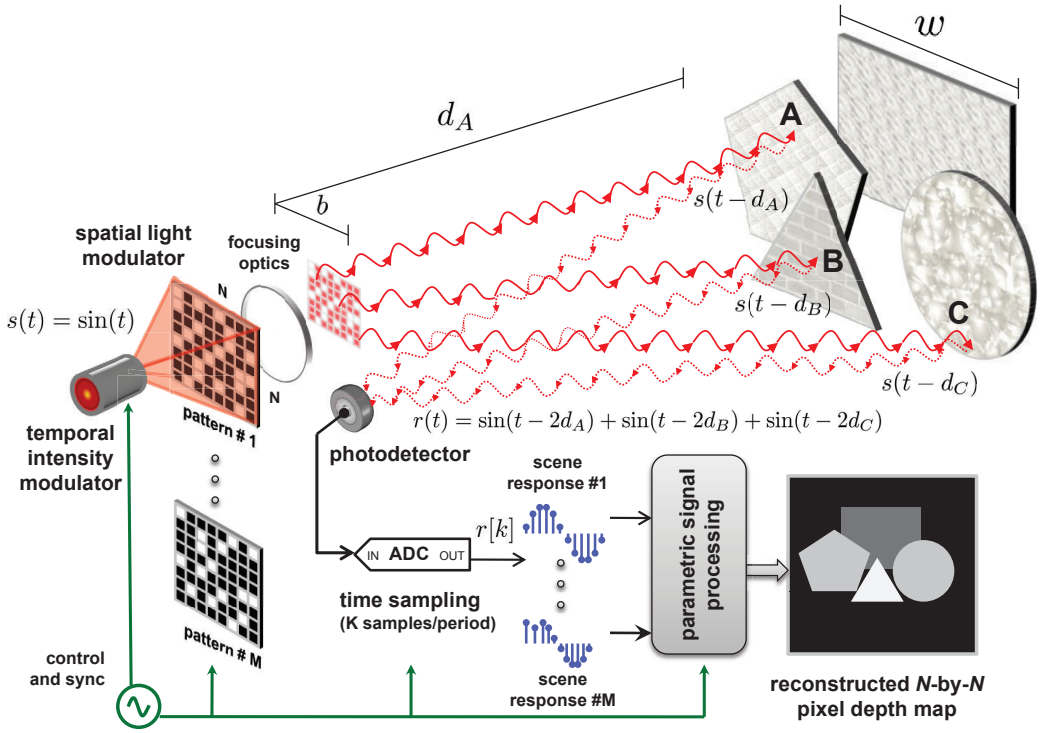


Figure 3-1: The proposed architecture for acquiring depth maps of scenes constituted of piecewise-planar facets. The scene is in far field, i.e., the baseline b and the dimensions of each planar facet w are much smaller than the distance between the imaging device and the scene. A light source with periodically-varying intensity $s(t)$ illuminates an $N \times N$-pixel SLM. The scene is serially illuminated with M chosen spatial patterns. For each patterned illumination the reflected light is focused at the photodetector and K digital time samples are recorded. The total $M \times K$ time samples are computationally processed to reconstruct an $N \times N$-pixel depth map of the scene.

with an $N \times N$ pixel resolution, which then projects a chosen 2D spatial pattern on the piecewise-planar scene. The light reflected from the illuminated portions of the scene is then focused at a time-resolving photodetector and digitized into K digital samples by an analog-to-digital converter (ADC) that is synchronized with the light source. This measurement process is repeated M times; depending on the desired spatial resolution, M typically ranges from 1 to 5% of the total number of pixels in the SLM. The recorded time samples are computationally processed to obtain a 2D scene depth map at the same pixel resolution as the SLM.

The sequence of SLM configurations and the computational processing each proceed in two

steps. Both steps exploit implicit or explicit modeling of the scene as piecewise planar.

Step 1 uses no spatial patterning from the SLM, i.e., a fully-transparent configuration. Under the assumption that the scene is approximately piecewise planar, the continuous-time light intensity signal at the single photodetector is approximated well in a certain parametric class. Estimation of the parameters of the signal implies recovery of the range of depth values present in the scene. Note that the use of a parametric signal modeling and recovery framework [67, 68] enables us to achieve high depth resolution relative to the speed of the time sampling at the photodetector. After discretizing the depths identified in this step, the remaining problem is to find correspondences between spatial locations and depths to form the depth map.

Step 2 uses many pseudorandom binary patterns on the SLM. The assumption that the scene is approximately piecewise planar translates to the Laplacian of the depth map being approximately sparse. We introduce a novel convex optimization problem that finds the depth map consistent with the measurements that approximately minimizes the number of nonzero entries in the Laplacian of the depth map. Solving this optimization problem with a general-purpose software package yields the desired depth map.

Outline

The remainder of this chapter is organized as follows: Section 3.1 establishes notation for our imaging setup. Sections 3.2 and 3.3 discuss the modeling and computational recovery associated with Steps 1 and 2, respectively, with the scene restricted to a single planar, rectangular facet for clarity of exposition. Section 3.4 describes the extensions of the framework that handle scenes with multiple planar facets that are not necessarily rectangular. The experiment is described in Section 3.5, and further extensions to textured scenes and non-impulsive illumination are discussed in Section 3.6.

3.1 Notation and assumptions for analysis of a single rectangular facet

Consider the setup shown in Fig. 3-2. A chosen SLM pattern is focused on the scene using a focusing system as shown in Fig. 3-2A. The center of the focusing system is denoted by O and is also the origin for a 3D coordinate system (X, Y, Z). All angles and distances are measured with respect to this global coordinate system. The focusing optics for the SLM illumination unit are chosen such that it has a depth-of-field (DOF) between distances d_1 and d_2 $(d_1 < d_2)$ along the Z dimension and a square field-of-view (FOV) along the X-Y axes. Thus, the dimensions of a square SLM pixel projected onto the scene remains constant within the DOF and across the FOV. We denote the dimensions of an SLM pixel within the DOF by $\Delta \times \Delta$. An SLM with higher spatial resolution corresponds to a smaller value of Δ. We also assume that the scene lies within the DOF so that all planar facets in the scene are illuminated by projection pixels of the same size. In our mathematical modeling and experiments, we only consider binary patterns, i.e., each SLM pixel is chosen to be either completely opaque or fully transparent. In Section 3.6, we discuss the possibility of using continuous-valued or gray-scale SLM patterns to compensate for rapidly-varying scene texture and reflectance.

The light reflected from the scene is focused at the photodetector. Note that we assume that the baseline separation b between the focusing optics of the detector and the SLM illumination optics is very small compared to the distance between the imaging device and the scene; i.e., if Q is a scene point as shown in Fig. 3-2, the total path length $O \rightarrow Q \rightarrow$ photodetector is approximately equal to the path length $O \rightarrow Q \rightarrow O$. Thus, we may conveniently model O as the effective optical center of the entire imaging setup (illumination and detector).

Sections 3.2 and 3.3 provide analyses of the time-varying light intensity at the detector in response to impulse illumination of a scene containing a single rectangular planar facet. The dimensions of the facet are $W \times L$. Let OC be the line that lies in the Y-Z plane and is also perpendicular to the rectangular facet. The plane is tilted from the zero-azimuth

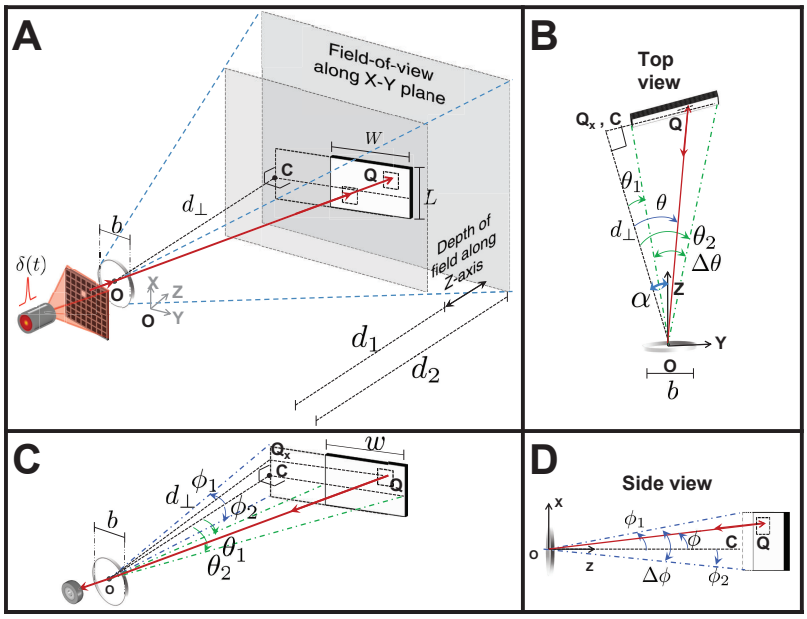


Figure 3-2: (A) Scene setup for parametric signal modeling of TOF light transport; (B) Top view; (C) Notation for various angles; (D) Side view.

axis (marked Z in Fig. 3-2), but the developments of Section 3.2 will show that this tilt is immaterial in our approach to depth map construction. For simplicity, we assume no tilt from the zenith axis (marked X in Fig. 3-2); a nonzero tilt would be immaterial in our approach.

The following parameters completely specify the rectangular facet (see Fig. 3-2C):

- $d_\perp$ denotes the length of the line OC.
- ϕ_1 and ϕ_2 are angles between line OC and the extreme rays connecting the vertical edges of the rectangular facet to O, and $\Delta\phi = |\phi_1 - \phi_2|$ is their difference; clearly, $\Delta\phi$ is related to L.
- θ_1 and θ_2 are angles between line OC and the extreme rays connecting the horizontal edges of the rectangular facet to O, and $\Delta\theta = |\theta_1 - \theta_2|$ is their difference; clearly, $\delta\theta$ is related to W.
- α is the angle between OC and the Z axis in the Y-Z plane.

For our light transport model, we assume that the scene is in the far field, i.e., the dimensions of the rectangular facet are small compared to the distance between the scene and the imaging device, or $W \ll d_1$ and $L \ll d_1$. This implies that $\Delta\phi$ and $\Delta\theta$ are small angles and that the radial fall-off attenuation of light arriving from different points on the rectangular facet is approximately the same for all the points. For developing the basic light transport model we also assume that the rectangular facet is devoid of texture and reflectance patterns. When a 2D scene photograph or image is available prior to data acquisition, then this assumption can be relaxed without loss of generality as discussed in Section 3.6. Finally, we set the speed of light to unity so that the numerical value of the time taken by light to traverse a given distance is equal to the numerical value of the distance.

3.2 Response of a single rectangular facet to fully-transparent SLM pattern

3.2.1 Scene response.

Let Q be a point on the rectangular planar facet at an angle of θ ($\theta_1 < \theta < \theta_2$) and ϕ ($\phi_1 < \phi < \phi_2$) with respect to the line OC as shown in Fig. 3-2. A unit-intensity illumination pulse, $s(t) = \delta(t)$, that originates at the source at time $t = 0$ will be reflected from Q, attenuated due to scattering, and arrive back at the detector delayed in time by an amount proportional to the distance $2\,|OQ|$. Since the speed of light is set to be unity, the delay is exactly equal to the distance $2\,|OQ|$. Thus the signal incident on the photodetector in response to impulse illumination of Q is mathematically given by

$$q(t) \;=\; a\,\delta(t - 2\,|OQ|),$$

where a is the total attenuation (transmissivity) of the unit-intensity pulse. Since the photodetector has an impulse response, denoted by $h(t)$, the electrical output $r_q(t)$ of the photodetector is mathematically equivalent to convolution of the signal $q(t)$ and the detector

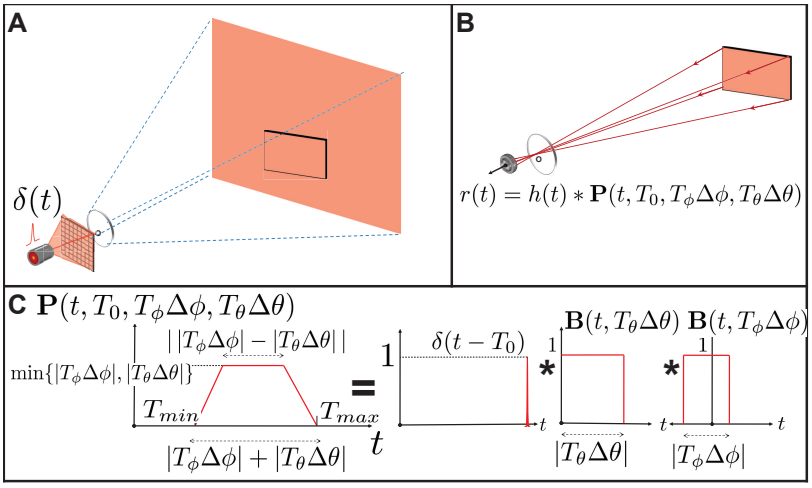


Figure 3-3: (A) All-ones scene illumination. (B) Scene response to all-ones scene illumination. (C) Diagrammatic explanation of the modeling of the parametric signal $p(t)$.

response $h(t)$:

$$r_q(t) \;=\; h(t) * a\,\delta(t - 2\,|OQ|) \;=\; a\,h(t - 2\,|OQ|).$$

Next, we use the expression for $r_q(t)$ to model the response of the scene in illumination to a fully transparent SLM pattern (see Fig. 3-3). The signal $r(t)$ obtained in this case is the total light incident at the photodetector from all possible positions of Q on the rectangular facet:

$$r(t) \;=\; a \int_{\phi_1}^{\phi_2} \int_{\theta_1}^{\theta_2} h(t - 2\,|OQ(\phi, \theta)|)\, d\theta\, d\phi, \tag{3.1}$$

presuming a linear detector response. From Fig. 3-2 we note that $|OQ(\phi, \theta)| \;=\; d_\perp \sqrt{\sec^2\phi + \tan^2\theta}$. Thus, substituting in Eq. (3.1) we have

$$\begin{aligned} r(t) \;&=\; a \int_{\phi_1}^{\phi_2} \int_{\theta_1}^{\theta_2} h\left(t - 2d_\perp \sqrt{\sec^2\phi + \tan^2\theta}\right) d\theta\, d\phi \\ &=\; a \int_{0}^{\Delta\phi} \int_{0}^{\Delta\theta} h\left(t - 2d_\perp \sqrt{\sec^2(\phi_1 + \phi) + \tan^2(\theta_1 + \theta)}\right) d\theta\, d\phi, \end{aligned} \tag{3.2}$$

where the equality in Eq. (3.2) follows from a change of variables $\phi \leftarrow (\phi - \phi_1)$ and $\theta \leftarrow (\theta - \theta_1)$. Since $\theta \in [0, \Delta\theta]$ and $\phi \in [0, \Delta\phi]$ are small angles, $\sqrt{\sec^2(\phi_1 + \phi) + \tan^2(\theta_1 + \theta)}$

is approximated well using a first-order expansion:

$$\begin{aligned}
&\sqrt{\sec^2(\phi_1+\phi)+\tan^2(\theta_1+\theta)} \\
&\quad\approx\quad \sqrt{\sec^2\phi_1+\tan^2\theta_1}+\frac{1}{\sqrt{\sec^2\phi_1+\tan^2\theta_1}}\left((\tan\phi_1\sec^2\phi_1)\,\phi+(\tan\theta_1\sec^2\theta_1)\,\theta\right) \qquad (3.3)
\end{aligned}$$

For notational simplicity, let $\gamma(\phi_1,\theta_1)=\sqrt{\sec^2\phi_1+\tan^2\theta_1}$. Using Eq. (3.3), Eq. (3.2) is approximated well by

$$\begin{aligned}
r(t) \;&=\; a\int_0^{\Delta\phi}\int_0^{\Delta\theta} h\left(t-2d_\perp\left(\gamma(\phi_1,\theta_1)+\frac{(\tan\phi_1\sec^2\phi_1)\,\phi+(\tan\theta_1\sec^2\theta_1)\,\theta}{\gamma(\phi_1,\theta_1)}\right)\right)\,d\theta\,d\phi \\
&=\; a\int_0^{\Delta\phi}\int_0^{\Delta\theta} h\left(t-\tau(\phi,\theta)\right)\,d\theta\,d\phi,
\end{aligned}$$

where

$$\tau(\phi,\theta) \;=\; 2d_\perp\gamma(\phi_1,\theta_1)+\frac{2d_\perp}{\gamma(\phi_1,\theta_1)}(\tan\phi_1\sec^2\phi_1)\phi+\frac{2d_\perp}{\gamma(\phi_1,\theta_1)}(\tan\theta_1\sec^2\theta_1)\theta. \qquad (3.4)$$

We now make an important observation. The time delay function $\tau(\phi,\theta)$ is a linear function of the angular variations $\phi_1\le\phi\le\phi_2$ and $\theta_1\le\theta\le\theta_2$. Thus, the time-difference-of-arrival of the returns from the closest point of the rectangular facet to the farthest point varies linearly. This is the central observation that allows us to model the returned signal using a parametric signal processing framework (as discussed next) and recover the scene depth variations using the proposed acquisition setup. Again for notational simplicity, let

$$T_0=2d_\perp\gamma(\phi_1,\theta_1),\quad T_\phi=\frac{2d_\perp}{\gamma(\phi_1,\theta_1)}\tan\phi_1\sec^2\phi_1,\quad T_\theta=\frac{2d_\perp}{\gamma(\phi_1,\theta_1)}\tan\theta_1\sec^2\theta_1.$$

Note that $T_0 > 0$ for all values of ϕ_1 and θ_1, but T_ϕ and T_θ may be negative or positive. With this notation and a change of variables, $\tau_1 \leftarrow T_\phi\,\phi$ and $\tau_2 \leftarrow T_\theta\,\theta$, we obtain

$$\begin{aligned} r(t) &= a\int_0^{\Delta\phi}\int_0^{\Delta\theta} h\left(t - T_0 - T_\phi\,\phi - T_\theta\,\theta\right)\,d\theta\,d\phi \\ &= \frac{a}{T_\phi\,T_\theta}\int_0^{T_\phi\,\Delta\phi}\int_0^{T_\theta\,\Delta\theta} h\left(t - T_0 - \tau_1 - \tau_2\right)\,d\tau_1\,d\tau_2 \\ &= \frac{a}{T_\phi\,T_\theta} h(t) * \delta(t - T_0) * \int_0^{T_\phi\,\Delta\phi}\delta(t-\tau_1)\,d\tau_1 * \int_0^{T_\theta\,\Delta\theta}\delta(t-\tau_2)\,d\tau_2 \\ &= \frac{a}{T_\phi T_\theta}\,h(t) * \delta(t - T_0) * \mathbf{B}(t, T_\phi\,\Delta\phi) * \mathbf{B}(t, T_\theta\,\Delta\theta) \end{aligned}$$

where $\mathbf{B}(t,T)$ is the *box function* with width $|T|$ as shown in Fig. 3-3C and defined as

$$\mathbf{B}(t,T) \;=\; \begin{cases} 1, & \text{for } t \text{ between 0 and } T; \\ 0, & \text{otherwise.} \end{cases}$$

The function $\mathbf{B}(t,T)$ is a *parametric function* that can be described with a small number of parameters despite its infinite Fourier bandwidth [67, 68]. The convolution of $\mathbf{B}(t, T_\phi\,\Delta\phi)$ and $\mathbf{B}(t, T_\theta\,\Delta\theta)$, delayed in time by T_0, is another parametric function as shown in Fig. 3-3C. We call this function $\mathbf{P}(t, T_0, T_\phi\,\Delta\phi, T_\theta\,\Delta\theta)$. It is piecewise linear and plays a central role in our depth acquisition approach for piecewise-planar scenes. With this notation, we obtain

$$r(t) \;=\; \frac{a}{T_\phi T_\theta}\,h(t) * \mathbf{P}(t, T_0, T_\phi\,\Delta\phi, T_\theta\,\Delta\theta).$$

The function $\mathbf{P}(t, T_0, T_\phi\,\Delta\phi, T_\theta\,\Delta\theta)$ is nonzero over a time interval $t \in [T_{\min}, T_{\max}]$ that is precisely the time interval in which reflected light from the points on the rectangular planar facet arrives at the detector. Also, for intuition, note that T_0 is equal to the distance between O and the lower left corner of the rectangular plane, but it may or may not be the point on the plane closest to O. With knowledge of $T_{\min}$ and $T_{\max}$ we obtain a region of certainty in which the rectangular facet lies. This region is a spherical shell centered at O with inner and outer radii equal to $T_{\min}$ and $T_{\max}$ respectively (see Fig. 3-4). Within this shell, the rectangular planar facet may have many possible orientations and positions.

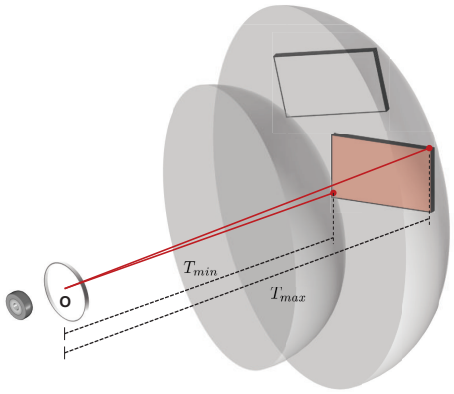


Figure 3-4: The signal $p(t)$ only provides information regarding the depth ranges present in the scene. It does not allow us to estimate the position and shape of the planar facet in the FOV of the imaging system. At best, the facet can be localized to lie between spherical shells specified by T_{min} and T_{max}. In this figure two possible positions for the rectangular facet are shown.

3.2.2 Parameter recovery

We wish to estimate the function $\mathbf{P}(t, T_0, T_\phi\,\Delta\phi, T_\theta\,\Delta\theta)$ and hence the values of T_{min} and T_{max} by processing the digital samples $r[k]$ of the function $r(t)$. The detector impulse response $h(t)$ is generally modeled as a bandlimited lowpass filter. Thus, the general deconvolution problem of obtaining $\mathbf{P}(t, T_0, T_\phi\,\Delta\phi, T_\theta\,\Delta\theta)$ from samples $r[k]$ is ill-posed and highly sensitive to noise. However, our modeling shows that the light transport function $\mathbf{P}(t, T_0, T_\phi\,\Delta\phi, T_\theta\,\Delta\theta)$ is piecewise linear. This knowledge makes the recovery of $\mathbf{P}(t, T_0, T_\phi\,\Delta\phi, T_\theta\,\Delta\theta)$ a *parametric deconvolution* problem that we solve using the parametric signal processing framework described in [48].

It is important to emphasize that the analysis up to this point is independent of the tilt α and orientation of the rectangular plane with respect to the global coordinate system (X, Y, Z); i.e., the tilt α has not appeared in any mathematical expression. Thus, the parametric function $\mathbf{P}(t, T_0, T_\phi\,\Delta\phi, T_\theta\,\Delta\theta)$ describing the light transport between the imaging device and the rectangular planar facet is independent of the orientation of the line OC. This is intuitive because all the results were derived by considering a new frame of reference involving the rectangular plane and the normal to the plane from the origin, OC.

The derived parametric light signal expressions themselves did not depend on how OC is oriented with respect to the global coordinate system but rather depend on the relative position of the plane with respect to OC. This explains why it is not possible to infer the position and orientation of the planar facet in the FOV of the system from the estimates of $\mathbf{P}(t, T_0, T_\phi\,\Delta\phi, T_\theta\,\Delta\theta)$. Recovery of the position and orientation of a rectangular planar facet is accomplished in Step 2 of our method using patterned illuminations as described in Section 3.3 below.

3.3 Response of a single rectangular facet to binary SLM pattern

3.3.1 Notation

As discussed in Section 3.1, the SLM pixels discretize the FOV into small squares of size $\Delta \times \Delta$. We index both the SLM pixels and the corresponding scene points by (i, j). Since we illuminate the scene with a series of M different binary SLM patterns, we also assign an index p for the illumination patterns. The full collection of binary SLM values is denoted $\{c_{ij}^p : i = 1, \ldots, N,\, j = 1, \ldots, N,\, p = 1, \ldots, M\}$.

Let $\mathbf{D}$ denote the *depth map* that we wish to construct. Then $\mathbf{D}_{ij}$ is the depth in the direction of illumination of SLM pixel (i, j), assuming rays in that direction intersect the rectangular facet; set $\mathbf{D}_{ij}$ to zero otherwise. More specifically, we use the lower-left corner of the projection of the pixel onto the planar facet, as shown in Fig. 3-5A. It is convenient to also define the *index map*, $\mathbf{I} = \{\mathbf{I}_{ij} : i = 1, \ldots, N,\, j = 1, \ldots, N\}$, associated with the rectangular facet through

$$\mathbf{I}_{ij} \;=\; \begin{cases} 1, & \text{if rays along SLM illumination pixel } (i, j) \text{ intersect the rectangular facet;} \\ 0, & \text{otherwise.} \end{cases}$$

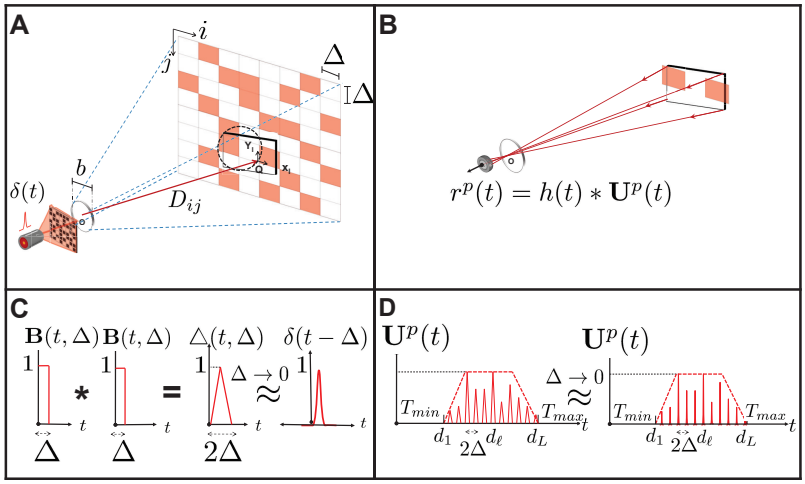


Figure 3-5: (A) Binary patterned scene illumination. (B) Scene response to all-ones scene illumination. (C) Diagrammatic explanation of the high-resolution SLM (small Δ) approximation. (D) Modeling of the parametric signal $\mathbf{U}^p(t)$ as a weighted sum of equally-spaced Diracs. Note that $\mathbf{U}^p(t)$ has the same time envelope as the signal $\mathbf{P}(t, T_0, T_\phi\,\Delta\phi, T_\theta\,\Delta\theta)$.

3.3.2 Scene response.

If we consider the rectangular facet as being composed of smaller rectangular facets of size $\Delta \times \Delta$, then following the derivation described in Section 3.2.1 we find that the light signal received at the detector in response to patterned, impulsive illumination of the rectangular facet is given by

$$r^p(t) = \sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij}^p \mathbf{I}_{ij} \left(a\, h(t) * \int_0^\Delta \int_0^\Delta \delta(t - 2\mathbf{D}_{ij} - 2x_\ell - 2y_\ell)\, dx_\ell\, dy_\ell \right) \tag{3.5}$$

$$= \sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij}^p \mathbf{I}_{ij} \left(\frac{a}{4}\, h(t) * \delta(t - 2\mathbf{D}_{ij}) * \mathbf{B}(t,\Delta) * \mathbf{B}(t,\Delta) \right)$$

$$= \frac{a}{4}\, h(t) * \left(\sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij}^p \mathbf{I}_{ij} \left(\delta(t - 2\mathbf{D}_{ij}) * \mathbf{B}(t,\Delta) * \mathbf{B}(t,\Delta) \right) \right). \tag{3.6}$$

Next, define the signal $\mathbf{U}^p(t)$ as

$$\mathbf{U}^p(t) = \sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij}^p \mathbf{I}_{ij} \left(\delta(t - 2\mathbf{D}_{ij}) * \mathbf{B}(t,\Delta) * \mathbf{B}(t,\Delta) \right). \tag{3.7}$$

The function $\triangle(t,\Delta) = \mathbf{B}(t,\Delta) * \mathbf{B}(t,\Delta)$ has a triangular shape with a base width of 2Δ as shown in Fig. 3-5C. In practice, when the SLM has high spatial resolution then Δ is very small, i.e., $\Delta \ll W$, $\Delta \ll L$, and $\triangle(t,\Delta)$ approximates a Dirac delta function $\delta(t)$. Thus, for a high-resolution SLM the signal $\mathbf{U}^p(t)$ is a weighted sum of uniformly-spaced impulses where the spacing between impulses is equal to 2Δ. Mathematically, we use $\lim_{\Delta\to 0} \mathbf{B}(t,\Delta) * \mathbf{B}(t,\Delta) = \lim_{\Delta\to 0} \delta(t-\Delta) = \delta(t)$ in Eq. (3.7) to obtain

$$\lim_{\Delta\to 0} \mathbf{U}^p(t) = \sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij}^p \mathbf{I}_{ij} \left(\delta(t-2\mathbf{D}_{ij}) * \delta(t)\right) = \sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij}^p \mathbf{I}_{ij}\, \delta(t-2\mathbf{D}_{ij}). \tag{3.8}$$

The parametric signal $\mathbf{U}^p(t)$ is obtained in the process of illuminating the scene with a patterned illumination and collecting light from illuminated portions of the scene ($c_{ij}^p = 1$) where the rectangular planar facet is present ($\mathbf{I}_{ij} = 1$). In particular, for a small value of Δ and fully-transparent SLM pattern (all-ones or $c_{ij}^p = 1 : i = 1,\ldots,N,\ j = 1,\ldots,N$) we have the following relation:

$$r^{\text{all-ones}}(t) = \lim_{\Delta\to 0} \sum_{i=1}^{N}\sum_{j=1}^{N} \mathbf{I}_{ij} \left(a\, h(t) * \int_0^{\Delta}\int_0^{\Delta} \delta(t-2\mathbf{D}_{ij}-2x_\ell-2y_\ell)\, dx_\ell\, dy_\ell \right) \tag{3.9}$$

$$= a \int_{\phi_1}^{\phi_2}\int_{\theta_1}^{\theta_2} h(t-2\,|OQ(\phi,\theta)|)\, d\theta\, d\phi = r(t) \tag{3.10}$$

where Eq. (3.10) follows from the fact that the double-summation approximates the double integral in the limiting case ($\Delta \to 0$). Additionally, Eq. (3.10) implies that $\mathbf{U}^{\text{all-ones}}(t) = \mathbf{P}(t, T_0, T_\phi\,\Delta\phi, T_\theta\,\Delta\theta)$. An important observation that stems from this fact is that for any chosen illumination pattern, the signal $\mathbf{U}^p(t)$ and the signal $\mathbf{P}(t, T_0, T_\phi\,\Delta\phi, T_\theta\,\Delta\theta)$, which is obtained by using the all-ones or fully-transparent illumination pattern, have support in time $[T_{\min}, T_{\max}]$. To be precise, if the points on the rectangular planar facet that are closest and farthest to O are illuminated, then both $\mathbf{U}^p(t)$ and $\mathbf{P}(t, T_0, T_\phi\,\Delta\phi, T_\theta\,\Delta\theta)$ have exactly the same duration and time delay. In practice, the binary patterns are randomly chosen with at least half of the SLM pixels "on," so it is highly likely that at least one point near the point closest to O and at least one point near the point farthest from O are illuminated. Hence, $\mathbf{U}^p(t)$ and $\mathbf{P}(t, T_0, T_\phi\,\Delta\phi, T_\theta\,\Delta\theta)$ are likely to have approximately

the same time support and time delay offset. This implies $\mathbf{D}_{ij} \in [T_{\min}, T_{\max}]$ (because the speed of light is normalized to unity).

3.3.3 Sampled data and Fourier-domain representation

Digital samples of the received signal $r^p[k]$ allow us to recover the depth map $\mathbf{D}$. First, note that the set of distance values, $\{\mathbf{D}_{ij} : i = 1, \ldots, N,\ j = 1, \ldots, N\}$, may contain repetitions; i.e., several (i, j) positions may have the same depth value $\mathbf{D}_{ij}$. All these points will lie on a circular arc on the rectangular facet as shown in Fig. 3-5A. Each $\mathbf{D}_{ij}$ belongs to the set of equally-spaced distinct depth values $\{d_1, d_2, \ldots, d_L\}$ where

$$L = \frac{T_{\max} - T_{\min}}{2\Delta}, \quad d_1 = T_{\min}, \quad d_\ell = d_1 + 2\Delta\ell, \quad \ell = 1, \ldots, L.$$

Note that the linear variation of the depths $d_1, \ldots, d_L$ is a direct consequence of Eq. (3.4), which states that there is a linear variation of distance from O of the closest point on the rectangular facet to the farthest. In the case of all-ones SLM illumination discussed in Section 3.2.1, we obtain the continuous signal $\mathbf{P}(t, T_0, T_\phi\,\Delta\phi, T_\theta\,\Delta\theta)$; in the patterned illumination case, we obtain a signal $\mathbf{U}^p(t)$ that is a weighted sum of uniformly-spaced impulses. With this new observation we have

$$\lim_{\Delta\to 0} \mathbf{U}^p(t) \;=\; \sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij}^p \mathbf{I}_{ij}\, \delta(t - 2\mathbf{D}_{ij}) \;=\; \sum_{\ell=1}^{L} \left(\sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij}^p I_{ij}^\ell \right) \delta(t - 2d_\ell), \tag{3.11}$$

where we define the matrix I^ℓ as

$$I_{ij}^\ell = \begin{cases} 1, & \text{if } \mathbf{D}_{ij} = d_\ell; \\ 0, & \text{otherwise,} \end{cases}$$

so $\mathbf{I}_{ij} = \sum_{\ell=1}^{L} I_{ij}^\ell$ and $\mathbf{D}_{ij} = \sum_{\ell=1}^{L} d_\ell I_{ij}^\ell$. With this new notation, the depth map $\mathbf{D}$ associated with the rectangular facet is the weighted sum of the index maps $\{I^\ell : \ell = 1, \ldots, L\}$ (see Fig. 3-6). Thus, constructing the depth map is now solved by finding the the L binary-valued index maps.

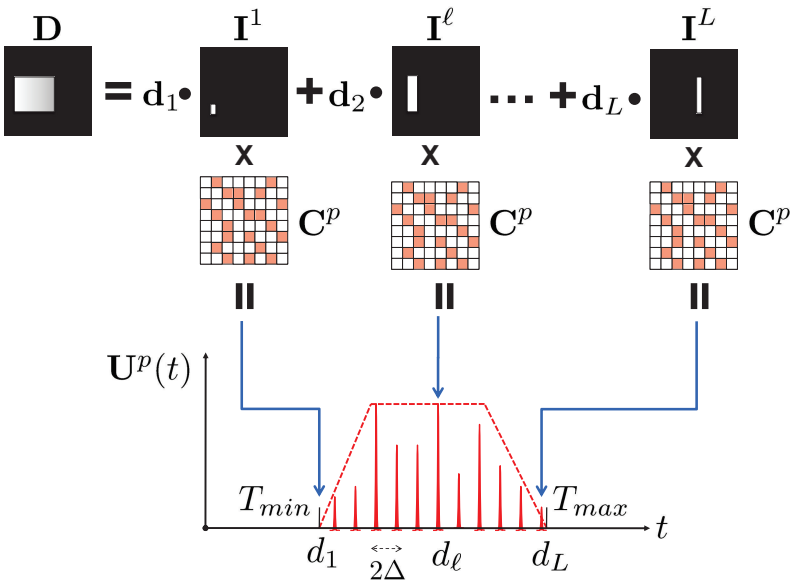


Figure 3-6: Depth masks are binary-valued $N \times N$ pixel resolution images which indicate the presence (1) or absence (0) of a particular depth at a particular position (i, j) in the discretized FOV of the sensor. Depending on Δ and the sampling rate, we obtain a uniform sampling of the depth range and hence obtain L depth masks, one per depth value. The depth map of a scene is the weighted linear combination of depth masks where the weights are the numerical values of the discretized depth range, $\{d_1, d_2, \ldots, d_L\}$.

Taking the Fourier transform $\mathfrak{F}\{\cdot\}$ of the signals on both sides of Eq. (3.11) we get

$$\begin{aligned}\mathfrak{F}\left\{\lim_{\Delta\to 0}\mathbf{U}^p(t)\right\} &= \mathfrak{F}\left\{\sum_{\ell=1}^{L}\left(\sum_{i=1}^{N}\sum_{j=1}^{N}c_{ij}^p I_{ij}^\ell\right)\delta(t-2d_\ell)\right\} \\ &= \sum_{\ell=1}^{L}\left(\sum_{i=1}^{N}\sum_{j=1}^{N}c_{ij}^p I_{ij}^\ell\right)\mathfrak{F}\left\{\delta(t-2d_\ell)\right\} = \sum_{\ell=1}^{L}\left(\sum_{i=1}^{N}\sum_{j=1}^{N}c_{ij}^p I_{ij}^\ell\right)e^{-\mathbf{i}\omega 2d_\ell}\end{aligned}$$

where $\mathbf{i} = \sqrt{-1}$. From elementary Fourier analysis and Eq. (3.6) we know that

$$\mathfrak{F}\left\{r^p(t)\right\} = \frac{a}{4}\mathfrak{F}\left\{h(t) * \mathbf{U}^p(t)\right\} = \frac{a}{4}\mathfrak{F}\left\{h(t)\right\}\mathfrak{F}\left\{\mathbf{U}^p(t)\right\}.$$

Let the ADC sample the signal incident on the photodetector at a sampling frequency of f

samples per second. Then, using elementary sampling theory [69], we obtain the relation

$$\mathfrak{F}\{r^p[k]\} \;=\; \frac{af}{4}\,\mathfrak{F}\{h[k]\}\,\mathfrak{F}\{\mathbf{U}^p[k]\} \;\Longrightarrow\; \frac{\mathfrak{F}\{r^p[k]\}}{\mathfrak{F}\{h[k]\}} \;=\; \frac{af}{4}\sum_{\ell=1}^{L}\left(\sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij}^{p} I_{ij}^{\ell}\right) e^{-\mathbf{i}(4\pi f\, d_\ell)\, k}.$$

Let K denote the total number of samples collected by the ADC and let the discrete Fourier transform (DFT) of the samples $\{r^p[k] : k = 1, \ldots, K\}$ be denoted by $\{R^p[k] : k = 1, \ldots, K\}$. Similarly define $\{H^p[k] : k = 1, \ldots, K\}$ for the impulse response samples $\{h^p[k] : k = 1, \ldots, K\}$. Then

$$\frac{R^p[k]}{H[k]} \;=\; \frac{af}{4}\sum_{\ell=1}^{L}\left(\sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij}^{p} I_{ij}^{\ell}\right) e^{-\mathbf{i}(4\pi f d_\ell)\, k}, \qquad k = 1, \ldots, K. \tag{3.12}$$

For notational simplicity let

$$y_\ell^p \;=\; \sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij}^{p} I_{ij}^{\ell}, \qquad \ell = 1, \ldots, L. \tag{3.13}$$

The constants a and f are computed using calibration and are computationally compensated using normalization. Since the values $\{d_1, d_2, \ldots, d_L\}$ are known, Eq. (3.12) can be represented as a system of linear equations as follows:

$$\begin{bmatrix} R^p[1]/H[1] \\ \vdots \\ R^p[k]/H[k] \\ \vdots \\ R^p[K]/H[K] \end{bmatrix} \;=\; \begin{bmatrix} 1 & \cdots & 1 & \cdots & 1 \\ \vdots & & \vdots & & \vdots \\ e^{-\mathbf{i}(4\pi f d_1)\, k} & \ldots & e^{-\mathbf{i}(4\pi f d_\ell)\, k} & \ldots & e^{-\mathbf{i}(4\pi f d_L)\, k} \\ \vdots & & \vdots & & \vdots \\ e^{-\mathbf{i}(4\pi f d_1)\, K} & \ldots & e^{-\mathbf{i}(4\pi f d_\ell)\, K} & \ldots & e^{-\mathbf{i}(4\pi f d_L)\, K} \end{bmatrix} \begin{bmatrix} y_1^p \\ \vdots \\ y_\ell^p \\ \vdots \\ y_L^p \end{bmatrix},$$

which can be compactly written as

$$\mathbf{R}^p/\mathbf{H} = \mathbf{V}\mathbf{y}^p \tag{3.14}$$

(where the division is elementwise). The matrix V is a *Vandermonde* matrix; thus $K \geq L$

ensures that we can uniquely solve the linear system in Eq. (3.14). Furthermore, a larger value of K allows us to mitigate the effect of noise by producing least square estimates of $\mathbf{y}^p$. Next, from Eq. (3.13) we see that $\mathbf{y}^p$ can also be represented with a linear system of equations as follows:

$$\begin{bmatrix} y_1^p \\ \vdots \\ y_\ell^p \\ \vdots \\ y_L^p \end{bmatrix} = \begin{bmatrix} I_{11}^1 \cdots I_{1N}^1 & I_{21}^1 \cdots I_{2N}^1 & \cdots & I_{N1}^1 \cdots I_{NN}^1 \\ \vdots \quad\quad \vdots & \vdots \quad\quad \vdots & & \vdots \\ I_{11}^\ell \cdots I_{1N}^\ell & I_{21}^\ell \cdots I_{2N}^\ell & \cdots & I_{N1}^\ell \cdots I_{NN}^\ell \\ \vdots \quad\quad \vdots & \vdots \quad\quad \vdots & & \vdots \\ I_{11}^L \cdots I_{1N}^L & I_{21}^L \cdots I_{2N}^L & \cdots & I_{N1}^L \cdots I_{NN}^L \end{bmatrix} \begin{bmatrix} c_{11}^p \\ \vdots \\ c_{1N}^p \\ c_{21}^p \\ \vdots \\ c_{2N}^p \\ \vdots \\ c_{N1}^p \\ \vdots \\ c_{NN}^p \end{bmatrix}. \tag{3.15}$$

From the M different binary SLM illumination patterns, we get M instances of Eq. (3.15) that can be combined into the compact representation

$$\underbrace{\mathbf{y}}_{L\times M} = \underbrace{\begin{bmatrix} I^1 \cdots I^\ell \cdots I^L \end{bmatrix}^T}_{L\times N^2} \underbrace{\mathbf{C}}_{N^2\times M}. \tag{3.16}$$

This system of equations is under-constrained since there are $L \times N^2$ unknowns (corresponding to the unknown values of $\begin{bmatrix} I^1 \dots I^\ell \dots I^L \end{bmatrix}$) and only $L \times M$ available transformed data observations $\mathbf{y}$. Note that $\mathbf{y}$ is computed using a total of $K \times M$ samples of the light signals received in response to $M \ll N^2$ patterned illuminations.

3.3.4 Algorithms for depth map reconstruction

Our goal is now to recover the depth map $\mathbf{D}$, which has $N \times N$ entries. To enable depth map reconstruction even though we have much fewer observations than unknowns, we exploit the structure of scene depth. We know that the depth values $\mathbf{D}_{ij}$ correspond to the distances

from O to points that are constrained to lie on a rectangular facet and that the distances $\mathbf{D}_{ij}$ are also linearly spaced between d_1 and d_L. The planar constraint and linear variation imply that the depth map $\mathbf{D}$ is *sparse* in the second-finite difference domain as shown Fig. 2-1. By exploiting this sparsity of the depth map, it is possible to recover $\mathbf{D}$ from the data $\mathbf{y}$ by solving the following constrained ℓ_1-regularized optimization problem:

$$\begin{aligned}\textbf{OPT:} \quad \underset{\mathbf{D}}{\text{minimize}} \quad & \left\|\mathbf{y} - \left[I^1 \dots I^\ell \dots I^L\right]^T \mathbf{C}\right\|_{\mathrm{F}}^2 + \left\|\left(\Phi \otimes \Phi^T\right) \mathbf{D}\right\|_1 \\ \text{subject to} \quad & \sum_{\ell=1}^{L} I_{ij}^\ell \;=\; 1, \quad \text{for all } (i,j), \qquad \sum_{\ell=1}^{L} d_\ell I^\ell \;=\; \mathbf{D}, \qquad \text{and} \\ & I_{ij}^\ell \in \{0,1\}, \quad \ell = 1,\dots,L, \quad i = 1,\dots,N, \quad j = 1,\dots,N.\end{aligned}$$

Here the Frobenius matrix norm squared $\|.\|_{\mathrm{F}}^2$ is the sum-of-squares of the matrix entries, the matrix Φ is the second-order finite difference operator matrix

$$\Phi = \begin{bmatrix} 1 & -2 & 1 & 0 & \cdots & 0 \\ 0 & 1 & -2 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & -2 & 1 \end{bmatrix},$$

and $\otimes$ is the standard Kronecker product for matrices.

The optimization problem **OPT** has an intuitive interpretation. Our objective is to find the depth map $\mathbf{D}$ that is most consistent with having a piecewise-planar scene. Such scenes are characterized by $\mathbf{D}$ having a discrete two-dimensional Laplacian $\left(\Phi \otimes \Phi^T\right) \mathbf{D}$ with a small number of nonzero entries (corresponding to the boundaries of the planar facets). The number of nonzero entries (the "ℓ_0 pseudonorm") is difficult to use because it is nonconvex and not robust to small perturbations, and the ℓ_1 norm is a suitable proxy with many optimality properties [44]. The problem **OPT** combines the above objective with maintaining fidelity with the measured data by keeping $\|\mathbf{y} - \left[I^1 \dots I^\ell \dots I^L\right] \mathbf{C}\|_{\mathrm{F}}^2$ small. The constraints $I_{ij}^\ell \in \{0,\, 1\}$ and $\sum_{\ell=1}^{L} I_{ij}^\ell = 1$ for all (i,j) are a mathematical rephrasing of the fact that each point in the depth map has a single depth value, so different depth values cannot be assigned to, one position (i,j). The constraint $\sum_{\ell=1}^{L} d_\ell I^\ell = \mathbf{D}$ expresses how the

depth map is constructed from the index maps.

While the optimization problem **OPT** already contains a convex relaxation in its use of $\|\Phi\,\mathbf{D}\|_1$, it is nevertheless computationally intractable because of the integrality constraints $I_{ij}^{\ell} \in \{0,\,1\}$. Using a further relaxation of $I_{ij}^{\ell} \in [0,\,1]$ yields the following tractable formulation.

$$\begin{aligned}
\textbf{R-OPT:}\quad \underset{\mathbf{D}}{\text{minimize}} \quad & \left\|\mathbf{y} - \left[I^1 \ldots I^{\ell} \ldots I^L\right]^T \mathbf{C}\right\|_{\mathrm{F}}^2 + \|\left(\Phi \otimes \Phi^T\right)\,\mathbf{D}\|_1 \\
\text{subject to} \quad & \sum_{\ell=1}^{L} I_{ij}^{\ell} \;=\; 1, \quad \text{for all } (i,j), \qquad \sum_{\ell=1}^{L} d_{\ell} I^{\ell} \;=\; \mathbf{D}, \qquad \text{and} \\
& I_{ij}^{\ell} \in [0,1] \quad \ell = 1,\ldots,L, \quad i = 1,\ldots,N, \quad j = 1,\ldots,N.
\end{aligned}$$

We solved the convex optimization problem **R-OPT** using `CVX`, a package for specifying and solving convex programs [70].

Summarizing, the procedure for reconstructing the depth map of a scene with a single rectangular planar facet is as follows:

1. Measure the digital samples of the impulse response of the photodetector $\{h[k] : k = 1,\ldots,K\}$. We assume that the ADC samples at least twice as fast as the bandwidth of the photodetector (Nyquist criterion).
2. Illuminate the entire scene with an impulse using an all-ones, fully-transparent SLM pattern and measure the digital samples of the received signal $\{r[k] : k = 1,\ldots,K\}$. In case the source is periodic, such as an impulse train, the received signal $r(t)$ will also be periodic and hence the samples need to be collected only in one period.
3. Process the received signal samples $\{r[k] : k = 1,\ldots,K\}$ and the impulse response samples, $\{h[k] : k = 1,\ldots,K\}$ using the parametric signal deconvolution algorithm described in [48] to estimate the piecewise-linear function $\mathbf{P}(t, T_0, T_{\phi}\,\Delta\phi, T_{\theta}\,\Delta\theta)$.
4. Using the estimate of $\mathbf{P}(t, T_0, T_{\phi}\,\Delta\phi, T_{\theta}\,\Delta\theta)$, infer the values of $T_{\min}$ and $T_{\max}$.
5. Illuminate the scene $M = N^2/20$ times using the randomly-chosen binary SLM patterns $\{c_{ij}^{p} : p = 1,\ldots,M\}$, again using an impulsive light source. Record K digital

time samples of the light signal received at the photodetector in response to each of the patterned illuminations $\{r^p[k] : k = 1, \ldots, K,\, p = 1, \ldots M\}$.

6. For each pattern, compute the transformed data $\mathbf{y} = [\mathbf{y}^1, \ldots, \mathbf{y}^M]$ as described in Section 3.3.2.
7. Construct the matrix $\mathbf{C}$ from the binary SLM patterns.
8. Solve the problem **R-OPT** to reconstruct the depth map $\mathbf{D}$ associated with the rectangular facet. This depth map contains information about the position, orientation and shape of the planar facet.

3.4 Depth map acquisition for general scenes

In this section we generalize the received signal model and depth map reconstruction developed in Sections 3.2 and 3.3 to planar facets of any shape and scenes with multiple planar facets.

3.4.1 General planar shapes

The signal modeling described in Section 3.2.1 applies to a planar facet with non-rectangular shape as well. For example, consider the illumination of a single triangular facet with the fully transparent SLM pattern as shown in Fig. 3-7 (left panel). In this case, the light signal received at the detector is

$$r(t) \;=\; a \int_{\phi_1}^{\phi_2} \int_{\theta_1(\phi)}^{\theta_2(\phi)} h(t - 2\,|OQ(\phi,\theta)|)\, d\theta\, d\phi.$$

Contrasting with Eq. (3.1), since the shape is not a rectangle, the angle θ does not vary over the entire range $[\theta_1, \theta_2]$. Instead, for a fixed value of angle ϕ, the angle θ can only vary from between some $\theta_1(\phi)$ and some $\theta_2(\phi)$. These limits of variation are determined by the shape of the object as shown in Fig. 3-7 (right panel).

Since the planar facet is in the far field, the distances of plane points from O still vary linearly. As a result, $r(t)$ is still equal to the convolution of the detector impulse response

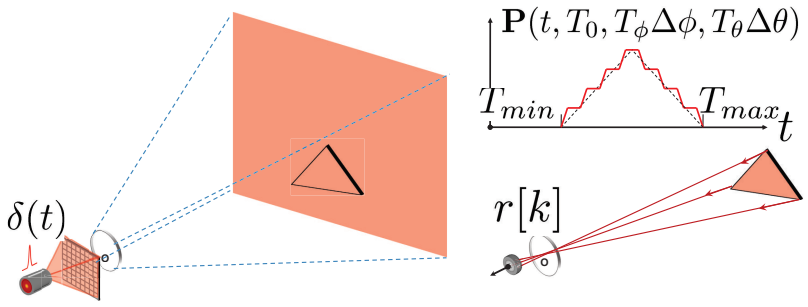


Figure 3-7: Parametric modeling for non-rectangular planes. The piecewise linear fit (shown in dotted black) is a good fit to the true parametric scene response from a triangular planar facet. This fit allows us to robustly estimate T_{min} and T_{max}.

with a parametric signal whose shape depends on the shape of the planar facet. For example, as shown in Fig. 3-7 (right panel), the profile of the signal $\mathbf{P}(t, T_0, T_\phi\, \Delta\phi, T_\theta\, \Delta\theta)$ is triangular with jagged edges. The task of estimating the signal $\mathbf{P}(t, T_0, T_\phi\, \Delta\phi, T_\theta\, \Delta\theta)$ corresponding to a general shape, such as a triangle, from the samples $r[k]$ is more difficult than estimating $\mathbf{P}(t, T_0, T_\phi\, \Delta\phi, T_\theta\, \Delta\theta)$ in the case of a rectangular facet. However, as we can see from Fig. 3-7 (right panel), a good piecewise-linear fit is still obtained using the samples of $r[k]$. This piecewise-linear approximation, although not exact, suffices for our purpose of estimating the shortest and farthest distance to the points on the planar facet. Thus it is possible to estimate the values T_{min} and T_{max} using the samples $r[k]$ without any dependence on the shape of the planar facet. Once T_{min} and T_{max} are estimated, we use the framework described in Section 3.3 to recover the depth map of the scene, which will also reveal the exact shape and orientation of the planar facet.

3.4.2 Multiple planar facets

When the scene has multiple planar facets, as shown in Fig. 3-8-A, the linearity of light transport and the linear response of the detector together imply that the detector output is the sum of the signals received from each of the individual planar facets. This holds equally well for the cases of fully-transparent and patterned SLM illumination.

Fig. 3-8A illustrates a scene composed of two planar facets illuminated with a fully-transparent

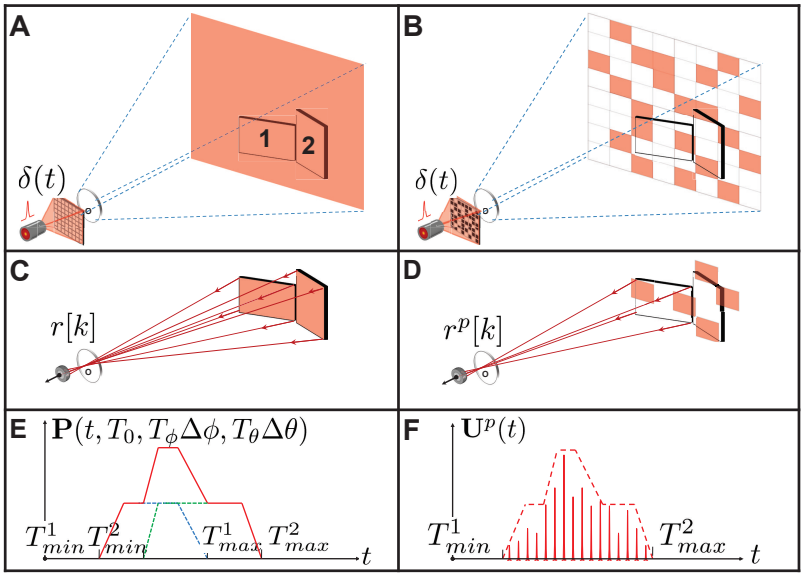


Figure 3-8: Parametric modeling in scenes with multiple planar facets. Since light transport is linear and assuming light adds linearly at the detector, the parametric signal that characterizes the scene response is the sum of multiple parametric signals. Thus even in the case of multiple planar facets, a piecewise-linear fit to the observed data allows us to reliably estimate the scene's depth range.

SLM setting. The total response is given by

$$r(t) \;=\; r_1(t) + r_2(t) \;=\; \mathbf{P}_1(t, T_{0,1}, T_{\phi,1}\Delta\phi_1, T_{\theta,1}\Delta\theta_1) + \mathbf{P}_2(t, T_{0,2}, T_{\phi,2}\Delta\phi_2, T_{\theta,2}\Delta\theta_2),$$

where $r_i(t)$ and $\mathbf{P}_i$ denote the response from planar facet i. The total response is thus a parametric signal. When points on two different planar facets are at the same distance from O (see Fig. 3-8C), there is time overlap between $\mathbf{P}_A(t, T_{0_A}, T_{\phi_A}\Delta\phi_A, T_{\theta_A}\Delta\theta_A)$ and $\mathbf{P}_B(t, T_{0_B}, T_{\phi_B}\Delta\phi_B, T_{\theta_B}\Delta\theta_B)$ (see Fig. 3-8E). In any case, closest distance $T_{\min}$ and farthest distance $T_{\max}$ can be estimated from $r(t)$. Thus the framework developed in Section 3.3 for estimating the distance set $\{d_1, d_2, \ldots, d_L\}$ applies here as well. Note that we do not need any prior information on how many planar facets are present in the scene.

Fig. 3-8B illustrates the same scene illuminated with a patterned SLM setting. Since the response to pattern p follows

$$r^p(t) \;=\; r^p_1(t) + r^p_2(t),$$

where $r_i^p(t)$ is the response from planar facet i, we can similarly write

$$\mathbf{U}^p(t) \;=\; \mathbf{U}_1^p(t) + \mathbf{U}_2^p(t).$$

Thus the problem of depth map reconstruction in case of scenes constituted of multiple planar facets is also solved using the convex optimization framework described in Section 3.3.

Fig. 3-8 illustrates rectangular facets that do not occlude each other, but the lack of occlusion is not a fundamental limitation. If a portion of a facet is occluded, it effectively becomes non-rectangular, as described in Section 3.4.1.

3.5 Experiments

3.5.1 Imaging setup and measurement

The proof-of-concept experiment to demonstrate the single-sensor compressive depth acquisition framework is illustrated in Fig. 3-9. The periodic light source was a mode-locked Ti:Sapphire femtosecond laser with a pulse width of 100 fs and a repetition rate of 80 MHz operating at a wavelength of 790 nm. It illuminated a MATLAB-controlled Boulder Nonlinear Systems liquid-crystal SLM with a pixel resolution of 512×512 pixels, each 15×15 μm. Pixels were grouped in blocks of 8×8 and each block phase-modulated the incident light to either 0° or 180° phase. The phase-modulated beam was passed through a half-wave plate followed by a polarizer to obtain the binary intensity pattern. A total of 205 binary patterns of 64×64 block-pixel resolution, were used for illumination. Each pattern was randomly chosen and had about half of the 4096 SLM blocks corresponding to zero phase (zero intensity after the polarizer). The average power in an illumination pattern was about 40 to 50 mW. The binary patterns were serially projected onto the scene comprised of two to four Lambertian planar shapes (see Fig. 3-10A) at different inclinations and distances. Our piecewise-planar scenes were composed of acrylic cut-outs of various geometric shapes coated with Edmund Optics NT83-889 white reflectance coating. The effects of speckle and interference were minimized by using convex lenses to project the SLM patterns on

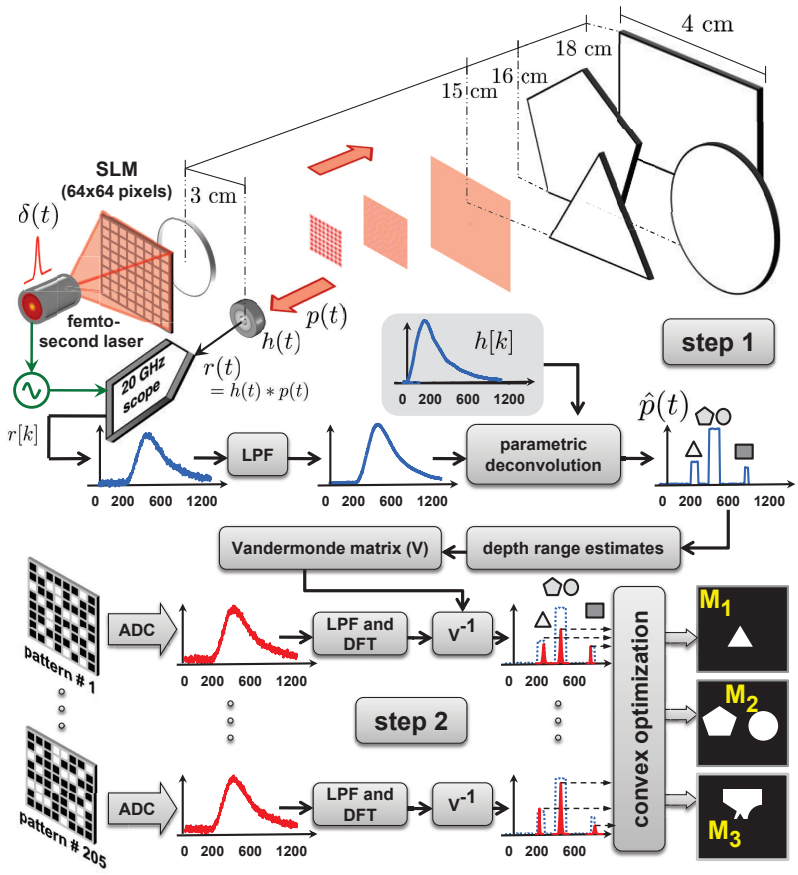


Figure 3-9: Schematic experimental setup to demonstrate depth estimation using our proposed framework. See text for details.

the scene. At a distance of 10 cm from the detector, each pixel in the scene was about 0.1 mm^2. For each pattern, the light reflected from all the illuminated portions of the scene was focused on a ThorLabs DET10A Si PIN diode with a rise time of 0.7 ns and an active area of 0.8 mm^2. A transparent glass slide was used to direct a small portion of the light into a second photodetector to trigger a 20 GHz oscilloscope and obtain the time origin for all received signals.

The depth map recovery is a two-step process: first we estimate the depth range within which the scene is present, and then we estimate the spatial locations, orientations and shapes of the planar facets. In Step 1, the scene was first illuminated with an all-ones pattern. The resulting convolution, $r(t)$, of the scene's true parametric response $\mathbf{P}(t)$ and

the detector's impulse response $h(t)$ was time sampled using the 20 GHz oscilloscope to obtain 1311 samples. These samples, $r[k]$, are lowpass filtered (LPF) to reduce sensor noise and processed using parametric deconvolution [48, 67, 71] to obtain the estimate $\hat{\mathbf{P}}(t)$ and hence the estimates of the distance ranges in which the planar facets lie. In Step 2, to recover the shapes and positions of the planar shapes, the scene is illuminated with 205 (5% of $64 \times 64 = 4096$) randomly-chosen binary patterns. The time samples collected in response to each patterned illumination are again low pass filtered (LPF) for denoising. The DFT of the filtered samples is processed using the Vandermonde matrix constructed using range estimates obtained in Step 1, to yield as many coefficients as there are distinct depth ranges (three in Fig. 3-9). These coefficients correspond to the product of the projected pattern and a binary-valued depth mask ($\mathbf{M_1}$, $\mathbf{M_2}$ and $\mathbf{M_3}$) that identifies the locations in the scene where the particular depth (d_1, d_2 and d_3) is present (see Fig. 3-6). The resulting 205×3 estimated coefficients are processed using a convex optimization framework that exploits the sparsity of the Laplacian of the depth map to recover the positions and shapes of the planar objects relative to the acquisition setup in the form of the three depth masks. Finally, these depth masks are weighted with the true depth values from Step 1 to reconstruct complete scene depth maps.

3.5.2 Depth map reconstruction results

Figs. 3-10A and 3-10B show the relative positions and approximate distances between the SLM focusing lens, the photodetector, and the two scenes constituted of white colored, Lambertian planar facets of different shapes and sizes. In Fig. 3-10A (also see Fig. 3-9), the dimensions of the planar facets are about 10 times smaller than the separation between SLM/photodetector and scene. Thus, there is little variation in the times-of-arrival of reflections from points on any single planar facet, as evidenced by the three concentrated rectangular pulses in the estimated parametric signal $\hat{\mathbf{P}}(t)$ in Fig. 3-10C. The time delays correspond to the three distinct depth ranges (15 cm, 16 cm and 18 cm). In Fig. 3-10B, there is significant variation in the times-of-arrival of reflections from points within each planar facet as well as overlap in the returns from the two facets. Thus, we get a broader

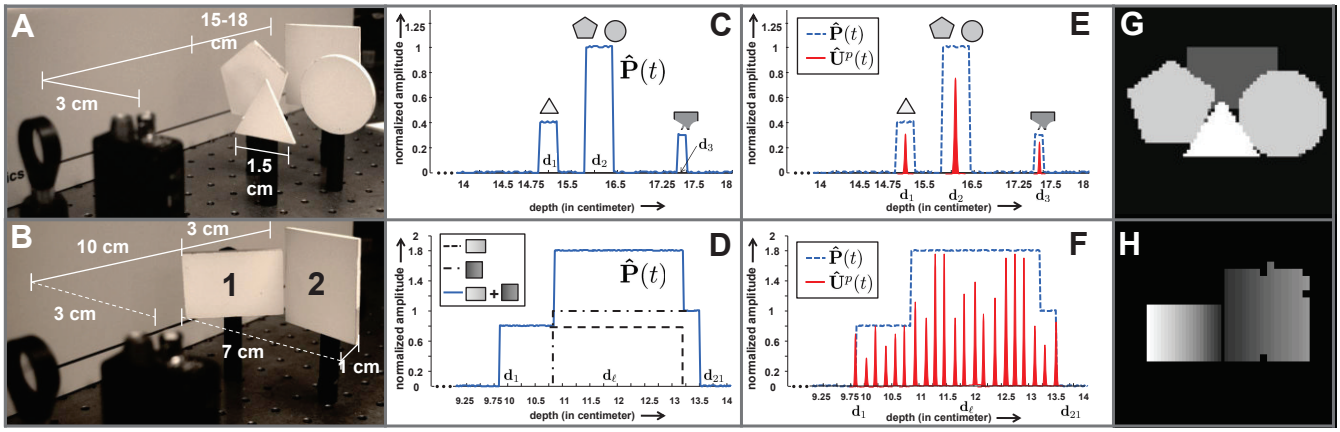


Figure 3-10: Photographs of experimental setup (A and B). Parametric signal estimate in response to all-transparent illumination (C and D). Parametric signal estimate in response to patterned illumination (E and F). Depth map reconstructions (G and H).

estimated parametric signal $\hat{\mathbf{P}}(t)$ that does not consist of disjoint rectangular pulses, and hence a continuous depth range as shown in Fig. 3-10D (solid blue curve). Overlaid on the experimental data in Fig. 3-10D are the computed separate contributions from the two planes in Fig. 3-10B (black dashed and black dash-dotted curves), conforming to our modeling in Section 3.2. Note that the depth range axis is appropriately scaled to account for ADC sampling frequency and the factor of 2 introduced due to light going back and forth. The normalized amplitude of the parametric signal $\hat{\mathbf{P}}(t)$ is an approximate measure of how much surface area of the scene is at a particular depth. The depth discretization and hence the range resolution is governed by the size of the projected SLM pixel, Δ. In our experiment the measured Δ is 0.1 mm and hence there are 21 discrete depths, $d_1, \ldots, d_{21}$ at a separation of 2Δ. Fig. 3-10E and Fig. 3-10F show the parametric signal $\mathbf{U}^p(t)$ that is recovered in the case of the first patterned illumination for the scenes in Fig. 3-10A and Fig. 3-10B, respectively. Figs. 3-10G and 3-10H show 64×64-pixel depth maps reconstructed using time samples from patterned binary illuminations of both the scenes. The distinct depth values are rendered in gray scale with closest depth shown in white and farthest depth value shown in dark gray; black is used to denote the scene portions from where no light is collected.

Our technique yielded accurate sub-cm depth maps with sharp edges. The range resolution

of our acquisition method – the ability to resolve close depths – depends on the bandwidth of the temporal light modulation, the response time of the photodetector, and the sampling rate of the ADC. The spatial resolution of our output depth map is a function of the number of distinct patterned scene illuminations; a complex scene with a large number of sharp features requires a larger number of SLM illuminations. In the presence of synchronization jitter and sensor noise, we average over multiple periods and use a larger number of illumination patterns to mitigate the effect of noise (see Fig. 3-9).

3.6 Discussion and extensions

The central novelty of our work relative to common LIDAR and TOF camera technologies is our mechanism for attaining spatial resolution through spatially-patterned illumination. In principle, this saves time relative to a LIDAR system because an SLM pattern can be changed more quickly than a laser position, and the number of acquisition cycles M is far fewer than the number of pixels in the constructed depth map. The savings relative to a TOF camera is in the number of sensors.

Our proposed depth acquisition technique also has two significant potential advantages over TOF cameras: First, our method is invariant to ambient light because only the low-frequency components of the recorded signals are affected by ambient light; low-frequency disturbances in turn only affect the overall scaling and do not affect the shape, duration and time delay of the parametric signal $\mathbf{P}(t)$. Second, there is potential for power savings: instead of constantly illuminating the scene with high-powered LED sources independent of the scene depth range, as is the case in TOF cameras, the scene range estimate from Step 1 of our method can be used to adaptively control the optical power output depending on how close the scene is to the imaging device.

The main limitation of our framework is inapplicability to scenes with curvilinear objects, which would require extensions of the current mathematical model. If we abandon the parametric signal recovery aspect of Step 1, we may still more crudely estimate the overall range of depths in the scene and proceed with Step 2. However, this will increase L and

thus increase the computational complexity of depth map recovery. The degree to which it necessitates an increase in M requires further study. More generally, the relationship between M and the depth map quality requires further study; while the optimization problems introduced in Section 3.3.4 bear some similarity to standard compressed sensing problems, existing theory does not apply directly.

Another limitation is that a periodic light source creates a wrap-around error as it does in other TOF devices [35]. For scenes in which surfaces have high reflectance or texture variations, availability of a traditional 2D image prior to our data acquisition allows for improved depth map reconstruction as discussed next.

3.6.1 Scenes with non-uniform texture and reflectance

Natural objects typically have surface texture and reflectance variations. In our experiments we only considered objects with uniform Lambertian reflectance. Here we briefly discuss the extension of our formulation to the case of planar facets with non-uniform texture and reflectance patterns. This extension assumes an SLM with a high number of pixels (small Δ) that performs grayscale light modulation. (Our experiments use only binary light modulation.)

Let the scene reflectance coefficient in the (i, j) direction be a_{ij}. Then the response to an all-ones (fully-transparent) SLM illumination is

$$\begin{aligned} r^0(t) &= \lim_{\Delta \to 0} \sum_{i=1}^{N} \sum_{j=1}^{N} a_{ij} \mathbf{I}_{ij} \left(h(t) * \int_0^{\Delta} \int_0^{\Delta} \delta(t - 2\mathbf{D}_{ij} - 2x_l - 2y_l)\, dx_\ell\, dy_\ell \right) \\ &= \int_{\phi_1}^{\phi_2} \int_{\theta_1}^{\theta_2} a(\phi, \theta)\, h(t - 2\,|OQ(\phi, \theta)|)\, d\theta\, d\phi. \end{aligned}$$

The presence of the unknown reflectance variations $a(\phi, \theta)$ prevents us from modeling $r^0(t)$ as a convolution of $h(t)$ and a piecewise-linear parametric signal as described in Section 3.2.1. However, if prior to data acquisition we have a conventional 2D image (photograph) of the scene that provides an estimate of the scene reflectance $\{a_{ij} : i = 1, \ldots, N,\, j = 1, \ldots, N\}$, it is possible to compensate for the reflectance using a grayscale SLM illumination. Specif-

ically, the "inverse" illumination pattern a/a_{ij}, $i = 1, \ldots, N$, $j = 1, \ldots, N$, where a is a chosen proportionality constant, yields response

$$\begin{aligned} r^{-1}(t) &= \lim_{\Delta \to 0} \sum_{i=1}^{N} \sum_{j=1}^{N} a_{ij} \frac{a}{a_{ij}} \mathbf{I}_{ij} \left(h(t) * \int_0^{\Delta} \int_0^{\Delta} \delta(t - 2\mathbf{D}_{ij} - 2x_\ell - 2y_\ell)\, dx_\ell\, dy_\ell \right) \\ &= a \int_{\phi_1}^{\phi_2} \int_{\theta_1}^{\theta_2} h(t - 2\,|OQ(\phi, \theta)|)\, d\theta\, d\phi \;=\; h(t) * \mathbf{P}(t, T_0, T_\phi\, \Delta\phi, T_\theta\, \Delta\theta), \end{aligned}$$

suitable for Step 1 of our method. Analogous inversion of the scene reflectance can be applied in Step 2 of our method.

3.6.2 Use of non-impulsive illumination sources

In our formulation and experiments we used a light impulse generator such as a femtosecond laser as our illumination source. However, we note that since the photodetector impulse response $h(t)$ is bandlimited, the overall imaging system is bandlimited. Thus it is possible to use non-impulsive sources that match the band limit of the detector without losing any imaging quality. Here we derive an expression for the signal received at the photodetector when we use a general time-varying source $s(t)$ instead of an impulse $\delta(t)$.

The scene defines a linear and time-invariant (LTI) system from illumination to detection. This is easy to verify: light transport is linear, and if we illuminate the scene with a time-delayed pulse, the received signal is delayed by the same amount. We have already modeled as $r(t)$ the output of the system in response to impulse illumination. Thus, the signal received at the photodetector in response to illumination using source $s(t)$ is given by $s(t) * r(t)$, the convolution of $r(t)$ with the source signal $s(t)$. Since $r(t) = h(t) * \mathbf{P}(t, T_0, T_\phi\, \Delta\phi, T_\theta\, \Delta\theta)$ we have

$$s(t) * r(t) = s(t) * \{h(t) * \mathbf{P}(t, T_0, T_\phi\, \Delta\phi, T_\theta\, \Delta\theta)\} = \{s(t) * h(t)\} * \mathbf{P}(t, T_0, T_\phi\, \Delta\phi, T_\theta\, \Delta\theta). \tag{3.17}$$

Eq. (3.17) demonstrates that if we use a non-impulsive source $s(t)$ then all our formulations developed in Sections 3.2 and 3.3 are valid with one small change: use $s(t) * h(t)$ in place of $h(t)$.

Chapter 4

Compressive Depth Acquisition Using Single Photon Counting Detectors

In the previous chapter, we introduced the theoretical framework behind a single pixel depth acquisition camera. Experimental results and analysis of this framework were presented for a short range (less than 1 ft) scenes. This chapter, demonstrates a compressive depth acquisition system for longer range scenes (5 m away). Further, we address the power consumption constraint in the formulation in this chapter. At long ranges and in mobile devices, optical power budget is an important consideration in the design of sensing systems. Typically, highly sensitive detectors reduce the requirement of high active optical illumination. Examples of such sensors include avalanche photodiodes (APD) that can detect arrivals of single photons when operated in Geiger mode. However, these sensors are not easily fabricated in large 2D arrays. Therefore, reducing sensing requirements to a single, sensitive APD is an important advantage in such scenarios. We demonstrate low power operation of a compressive depth camera using a single pixel APD for long range scenes of interest.

The extension of the compressive depth acquisition camera described in this chapter has

three new contributions from its short range counterpart.

- We demonstrate that it is possible to acquire a 2D depth map of a long range scene using a single time-resolved detector and no scanning components, with spatial resolution governed by the pixel resolution of the DMD array and the number of sensing patterns. In this experiment, we place pseudo-randomly chosen binary patterns at the detector unlike the previous set up which projected illumination patterns. This demonstrates the operation of the framework under a dual light transport configuration.

- We show that the parametric signal processing used in our computational depth map reconstruction achieves significantly better range resolution than conventional non-parametric techniques with the same pulse widths.

- We experimentally validate our depth acquisition technique for typical scene ranges and object sizes using a low-power, pulsed laser and an APD. We also demonstrate the effectiveness of our method by imaging objects hidden behind partially-transmissive occluders, without any prior knowledge about the occluder.

For the rest of this chapter, we will describe the imaging setup, reconstruction process and present new results. This chapter contains material that also appears in [27]. The author would like to acknowledge the efforts of collaborators at the University of Rochester who set up the experiment and provided data.

4.1 Imaging setup and data acquisition

In our imaging setup (see Fig. 4-1), an omnidirectional, pulsed periodic light source illuminates the scene. The illumination unit comprises a function generator that produces 2 ns square pulses that drive a near-infrared (780 nm) laser diode to illuminate the scene with 2 ns Gaussian pulses with 50 mW peak power and a repetition rate of 10 MHz. Note that the pulse duration is shorter than the repetition rate of the pulses. Light reflected from the

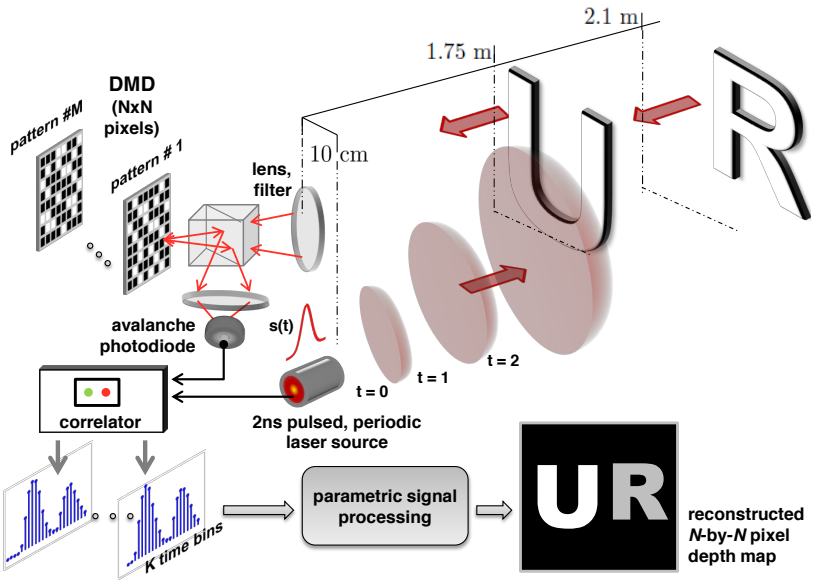


Figure 4-1: Compressive depth acquisition setup showing a 2 ns pulsed laser source $s(t)$, a DMD array with $N \times N$-pixel resolution, and a single photon-counting detector. For each sensing pattern, R illumination pulses are used to generate an intensity histogram with K time bins. This process is repeated for M pseudorandomly-chosen binary patterns and the $M \cdot K$ intensity samples are processed using a computational framework that combines parametric deconvolution with sparsity-enforcing regularization to reconstruct an $N \times N$-pixel scene depth map.

scene is focused onto a DMD which is then focused onto a single photon-counting detector. The detector is a cooled APD operating in Geiger mode. When a single photon is absorbed, the detector outputs a TTL pulse about 10 ns in width, with edge timing resolution of about 300 ps. After a photon is absorbed, the detector then enters a dead time of about 30 ns during which it is unable to detect photons. To build the histogram of arrival times, we use a correlating device (Picoquant Timeharp) designed for time-correlated single-photon counting. The correlator has two inputs: *start* and *stop*. The output of the laser pulse generator is wired to start, and the APD output is wired to stop. The device then measures differences in arrival times between these two inputs to build up timing histograms over an acquisition time t_a; this acquisition time was different for the two scenes in our experiment. The pulse repetition rate was 10 MHz. The photon counting mechanism and the process of measuring the timing histogram are shown in Fig. 4-3. Scenes are set up so that objects are placed fronto-parallelly between 0.3 m to 2.8 m from the device. Objects are 30 cm-by-

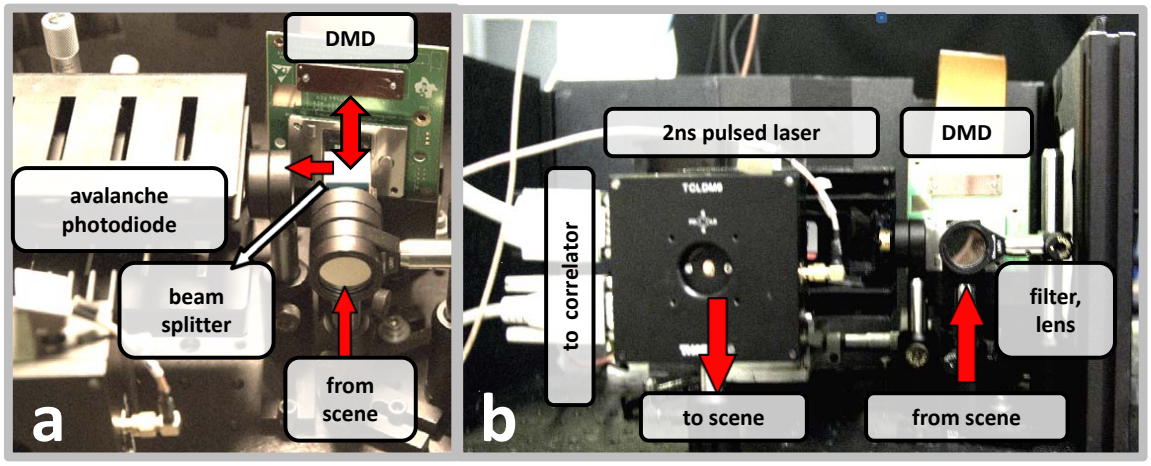


Figure 4-2: Experimental setup for compressive depth acquisition. (a) Close-up of the sensing unit showing the optical path of light reflected from the scene. (b) The complete imaging setup showing the pulsed source and the sensing unit.

30 cm cardboard cut-outs of the letters **U** and **R** at distances d_1 and d_2 respectively. Light reflected by the scene is imaged onto a DMD through a 10 nm filter centered at 780 nm with a 38 mm lens focused to infinity with respect to the DMD. We use a D4100 Texas Instruments DMD that has 1024×768 individually-addressable micromirrors. Each mirror can either be "ON" where it retro-reflects light to the APD or "OFF" where it reflects light away from the detector. Light that is retro-reflected to the APD provides input to the correlator. For the experiment, we used only 64×64 pixels of the DMD to collect reflected light. For each scene we recorded a timing histogram for 2000 patterns; these were 64×64 pseudorandomly-chosen binary patterns placed on the DMD. The pattern values are chosen uniformly at random to be either 0 or 1.

4.2 Signal modeling and depth map reconstruction

4.2.1 Parametric response of fronto-parallel facets

Consider a piecewise-planar scene comprising two fronto-parallel objects as shown in Fig. 4-1. When an omnidirectional pulsed source illuminates the scene, the signal $r(t)$ received at the single time-resolved photodetector is the convolution of a parametric signal $p(t)$ with the

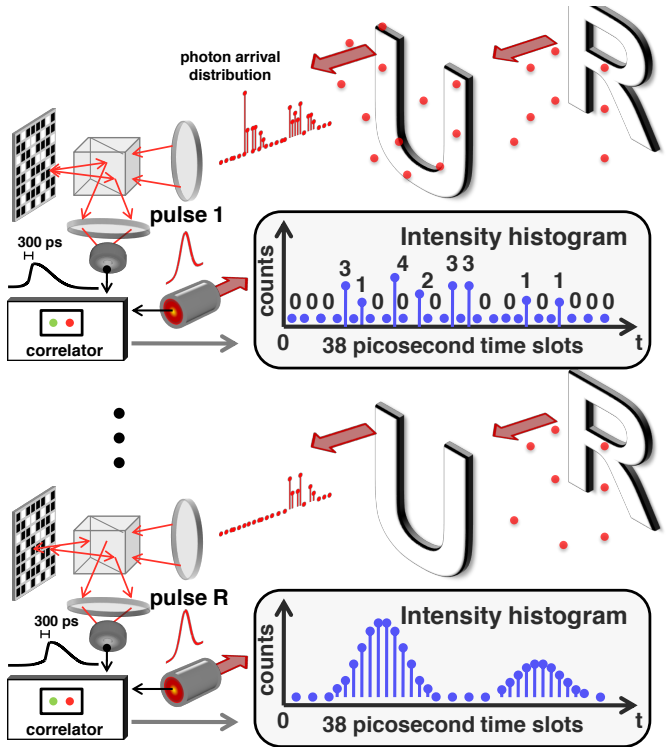


Figure 4-3: The process of generating a timing histogram. (top) For a fixed DMD pattern, scene illumination with the first pulse results in a low photon flux with a Poisson distributed photon arrival. The APD + correlator combination records the time of arrival of these photons with a 38 ps accuracy and increments the photon count in the respective time bins. (bottom) This counting process is repeated for R pulses and the entire intensity profile is generated.

pulse shape $s(t)$. The parametric signal $p(t)$ comprises time-shifted returns corresponding to the objects at depths d_1 and d_2. The returns from each object are highly concentrated in time because the scene is in far-field and the object dimensions are small compared to the distances d_1 and d_2. Thus, the parametric signal $p(t)$ is completely characterized by four parameters: the locations and amplitudes of the two peaks. (For complete scene response modeling we refer the reader to [26]). The problem of recovering these signal parameters from the discrete samples of $r(t)$, a lowpass filtered version of $p(t)$, is a canonical finite rate of innovation (FRI) sampling problem [48]. Note that the parameters of the scene response $p(t)$ only furnish information about the depth ranges present in the scene; they convey no information about the object shapes and their positions in the field-of-view of the imaging

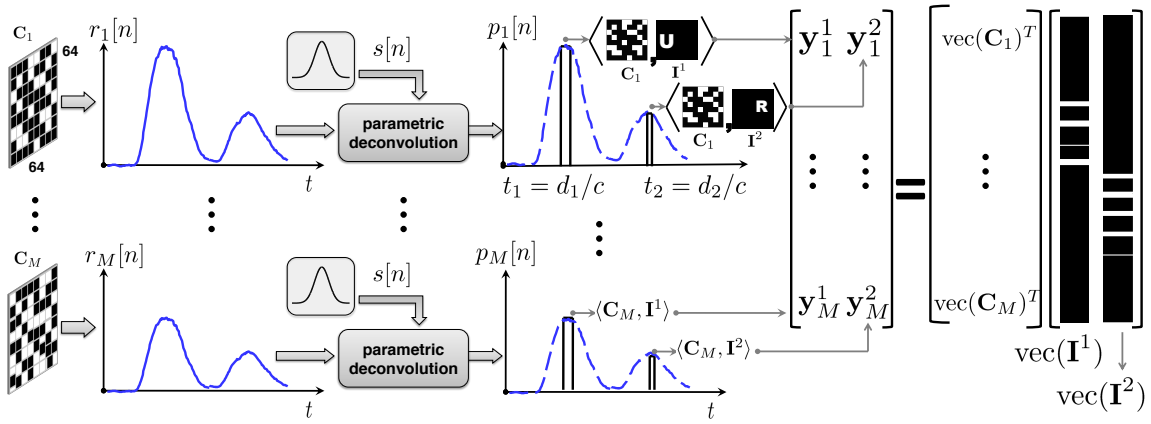


Figure 4-4: Depth map reconstruction algorithm. The intensity samples $r_i[n]$ acquired for each binary pattern $\mathbf{C}_i$ are first processed using parametric deconvolution to recover scene response $p_i[n]$. The positions of peak amplitudes in $p_i[n]$ provide depth estimates, d_1 and d_2, and the amplitudes are used to recover the spatial structure of the depth map (i.e. the depth masks $\mathbf{I}^1$ and $\mathbf{I}^2$) at these depth locations. The spatial resolution is recovered by a convex program that enforces gradient-domain sparsity and includes a robustness constraint.

device.

4.2.2 Shape and transverse position recovery

The next step is to obtain the shapes of objects and their transverse positions in the depth map. A single patterned return only provides depth information. However, when repeated for multiple pseudorandomly-chosen binary patterns, we find that the the heights of the peaks in the returned signal contribute useful information that help identify object shape. Note that the depth map $\mathbf{D}$ is a weighted combination of the two depth masks $\mathbf{I}^1$ and $\mathbf{I}^2$, i.e., $\mathbf{D} = d_1\mathbf{I}^1 + d_2\mathbf{I}^2$ [26]. Each binary-valued depth mask identifies the positions in the scene where the associated depth is present, thereby identifying the shape and transverse position of the object present at that depth. Having estimated d_1 and d_2 using parametric recovery of the signal $p(t)$, the problem of estimating $\mathbf{I}^1$ and $\mathbf{I}^2$ is a linear inverse problem. This is because the amplitude of the signal at the time instances corresponding to depths d_1 and d_2 is equal to the inner product of the DMD pattern $\mathbf{C}$ with the depth masks $\mathbf{I}^1$ and $\mathbf{I}^2$ respectively. Furthermore, the assumption that the scene is fronto-parallel translates to

the Laplacian of the depth map being sparse. Hence, we may possibly recover $\mathbf{I}^1$ and $\mathbf{I}^2$ using far fewer patterns, M, than the number of pixels N^2.

For each pattern $\mathbf{C}_i$, $i = 1, \ldots, M$, the digital samples of the received signal, $r_i[n]$, are processed using the parametric deconvolution framework to obtain the amplitudes of the recovered parametric signals $p_i[n]$ corresponding to the inner products $y_i^1 = \langle \mathbf{C}_i, \mathbf{I}^1 \rangle$ and $y_i^2 = \langle \mathbf{C}_i, \mathbf{I}^2 \rangle$. This data can be compactly represented using the linear system

$$\underbrace{[\mathbf{y}^1\ \mathbf{y}^2]}_{M\times 2} = \underbrace{\mathbf{C}}_{M\times N^2}\ \underbrace{[\mathrm{vec}(\mathbf{I}^1)\ \mathrm{vec}(\mathbf{I}^2)]}_{N^2\times 2}.$$

This is an underdetermined system of linear equations because $M \ll N^2$. But, since the Laplacian of the depth map $\mathbf{D}$ is sparse, we can potentially solve for good estimates of the depth masks $\mathbf{I}^1$ and $\mathbf{I}^2$ using the sparsity-enforcing joint optimization framework outlined in the next section.

4.2.3 Depth map reconstruction

We propose the following optimization program for recovering the depth masks $\mathbf{I}^1$ and $\mathbf{I}^2$ and hence, the depth map $\mathbf{D}$ from the observations $[\mathbf{y}^1\ \mathbf{y}^2]$:

$$\textbf{OPT: } \min_{\mathbf{D}} \left\| [\mathbf{y}^1\ \mathbf{y}^2] - \mathbf{C}[\mathrm{vec}(\mathbf{I}^1)\ \mathrm{vec}(\mathbf{I}^2)] \right\|_{\mathrm{F}}^2 + \left\| \left(\Phi \otimes \Phi^{\mathrm{T}}\right)\ \mathbf{D} \right\|_1$$

subject to

$$\mathbf{I}_{k\ell}^0 + \mathbf{I}_{k\ell}^1 + \mathbf{I}_{k\ell}^2 \ = \ 1, \quad \text{for all } (k, \ell),$$

$$\mathbf{D} = d_1\mathbf{I}^1 + d_2\mathbf{I}^2,$$

$$\text{and} \mathbf{I}_{k\ell}^0, \mathbf{I}_{k\ell}^1, \mathbf{I}_{k\ell}^2 \in \{0, 1\}, \quad k, \ell = 1, \ldots, N.$$

$\mathbf{I}_{k\ell}^0$ is the depth mask corresponding to the portions of the scene that did not contribute to the returned signal. Here the Frobenius matrix norm squared $\|.\|_{\mathrm{F}}^2$ is the sum-of-squares of

the matrix entries, the matrix Φ is the second-order finite difference operator matrix

$$\Phi = \begin{bmatrix} 1 & -2 & 1 & 0 & \cdots & 0 \\ 0 & 1 & -2 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & -2 & 1 \end{bmatrix},$$

and $\otimes$ is the standard Kronecker product for matrices. The number of nonzero entries (the "ℓ_0 pseudonorm") is difficult to use because it is nonconvex and not robust to small perturbations; the ℓ_1 norm is a suitable proxy with many optimality properties [44]. The problem **OPT** combines the above objective with maintaining fidelity with the measured data by keeping $\left\| [\mathbf{y}^1 \ \mathbf{y}^2] - \mathbf{C}[\mathrm{vec}(\mathbf{I}^1) \ \mathrm{vec}(\mathbf{I}^2)] \right\|_{\mathrm{F}}^2$ small. The constraints $\mathbf{I}^0_{k\ell}, \mathbf{I}^1_{k\ell}, \mathbf{I}^2_{k\ell} \in \{0, 1\}$ and $\mathbf{I}^0_{k\ell} + \mathbf{I}^1_{k\ell} + \mathbf{I}^2_{k\ell} = 1$ for all (k, ℓ) are a mathematical rephrasing of the fact that each point in the depth map has a single depth value, so different depth values cannot be assigned to one position (k, ℓ). The constraint $\mathbf{D} = d_1\mathbf{I}^1 + d_2\mathbf{I}^2$ expresses how the depth map is constructed from the index maps. While the optimization problem **OPT** already contains a convex relaxation in its use of $\|\Phi\,\mathbf{D}\|_1$, it is nevertheless computationally intractable because of the integrality constraints $\mathbf{I}^0_{k\ell}, \mathbf{I}^1_{k\ell}, \mathbf{I}^2_{k\ell} \in \{0, 1\}$. We further relax this constraint to $\mathbf{I}^0_{k\ell}, \mathbf{I}^1_{k\ell}, \mathbf{I}^2_{k\ell} \in [0,\, 1]$ to yield a tractable formulation. We also show in Section 4.3 that this relaxation allows us to account for partially-transmissive objects in our scenes. We solved the convex optimization problem with the relaxed integrality constraint using `CVX`, a package for specifying and solving convex programs [70]. Note that this approach solves a single optimization problem and does not use range gating. Techniques employed by [47] assume knowledge of ranges of interest *a priori* and solve a CS-style optimization problem per range of interest. In the next section we discuss our experimental prototype and computational reconstruction results.

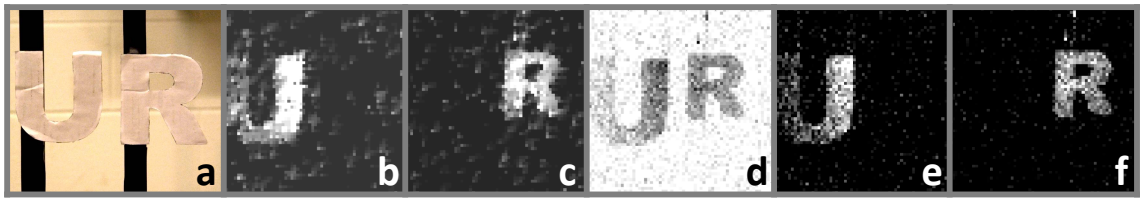


Figure 4-5: Reconstruction results. (a) Scene setup. (b), (c), and (d) Depth masks $\mathbf{I}^1$ and $\mathbf{I}^2$ and the background mask $\mathbf{I}^0$ reconstructed using 500 patterns. (e) and (f) Depth masks reconstructed using 2000 patterns.

4.3 Experimental results

In this section, we discuss constructions of depth maps of two scenes using varying number of measurements, M. The first scene (see Fig. 4-5) has two cardboard cut-outs of the letters **U**, **R** placed at 1.75 m and 2.1 m respectively from the imaging setup. Depths are identified from the time-shifted peaks in the timing histogram. Recovery of spatial correspondences proceeds as described in Section 4.2.2. We solve a single optimization problem to recover depth masks corresponding to each object. In Fig. 4-5 b-f, we see depth masks for our first experimental scene (Fig. 4-5 a) for different numbers of total patterns used. At 500 patterns (12% of the total number of pixels), we can clearly identify the objects in depth masks $\mathbf{I}^1$, $\mathbf{I}^2$ (Fig. 4-5 b, c) with only some background noise; we also see the background depth mask corresponding to regions that do not contribute any reflected returns (see Fig. 4-5 d). Using 2000 patterns (48.8% of the total number of pixels) almost completely mitigates any background noise while providing accurate a depth map (Fig. 4-5 e, f).

Imaging scenes with unknown transmissive occluders. In the second scene we consider a combination of transmissive and opaque objects and attempt to recover a depth map. The scene is shown in Fig. 4-6 a. Note that the burlap placed at 1.4 m from the imaging device completely fills the field of view. A 2D image of the scene reveals only the burlap. However, located at 2.1 m from the imaging device are cardboard cut-outs of U and R – both at the same depth. These objects are completely occluded in the 2D reflectance image. Also seen in Fig. 4-6 is a timing histogram acquired with acquisition time $t_a = 4$ s.

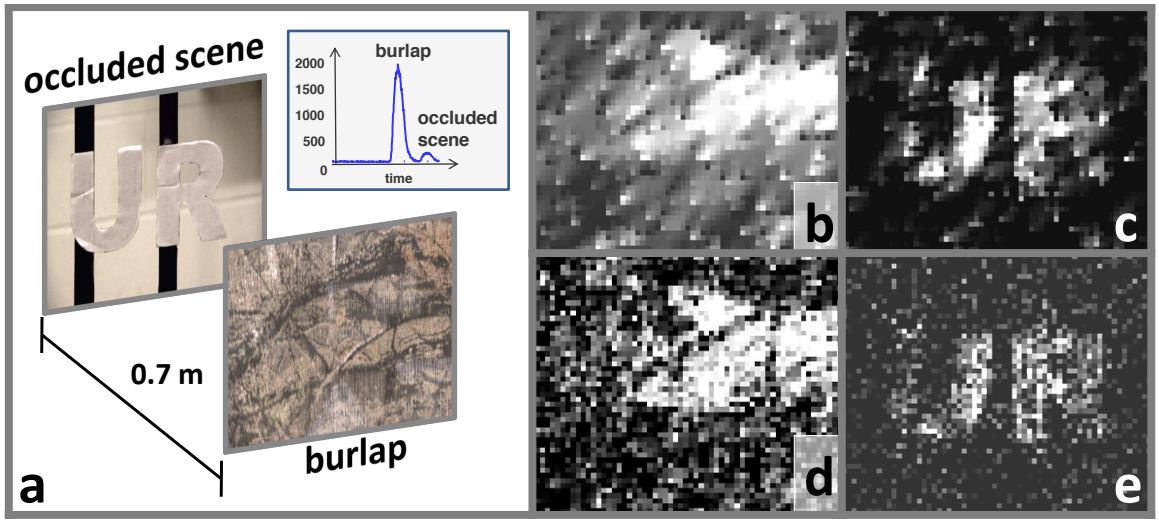


Figure 4-6: Occluded scene imaging. (a) Setup for the scene occluded with a partially-transmissive burlap; the shapes U and R are at the same depth. (b) and (c) Reconstructed depth masks for burlap and scene using 500 patterns; (d) and (e) using 2000 patterns. Note that no prior knowledge about the occluder was required for these reconstructions.

The histogram shows that the burlap contributes a much larger reflected signal (12 times stronger) than the contribution of the occluded objects. Figs. 4-6 b, c show depth masks $\mathbf{I}^1$, $\mathbf{I}^2$ for the burlap and occluded objects respectively for 500 patterns while Figs. 4-6 d, e show depth masks obtained using 2000 patterns. The reconstruction of the depth map in the presence of a transmissive occluder is possible because of the relaxation of the integrality constraint.

High range resolution with slower detectors. When planar facets are separated by distances that correspond to time differences greater than the pulse width of the source, the time shift information can be trivially separated. The more challenging case is when facets are closely spaced or there are large number of distinct facets. A detailed analysis of these cases for recovering depth information can be found in [26]. However, in this work we focus on well-separated fronto-parallel planar facets. We briefly address the case where our scenes are illuminated by a system with longer source pulse width. This results in time shift information from planar facets bleeding into each other, that is, peaks that are not well separated in the returned signal. The use of a parametric signal modeling and

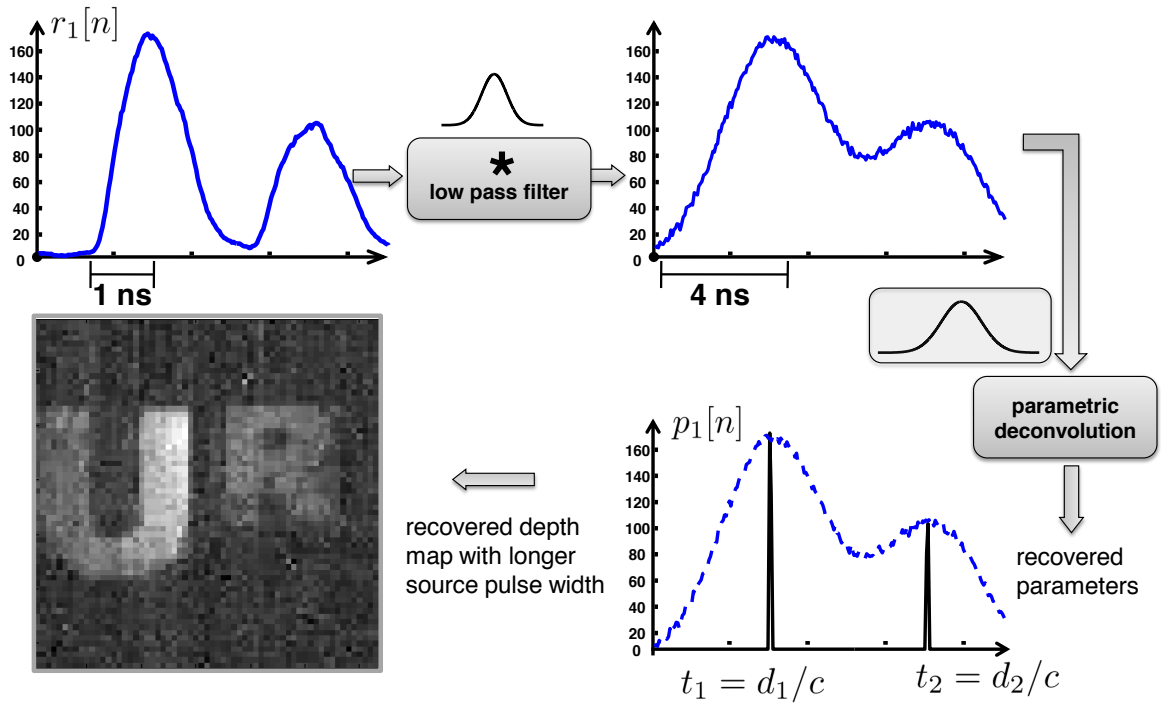


Figure 4-7: Depth map reconstruction with simulated scene response to longer source pulse width. We simulate poor temporal resolution by lowpass filtering the captured intensity profiles so that the Gaussian pulses overlap and interfere with the signal amplitudes. The effectiveness of the parametric deconvolution technique is demonstrated by accurate recovery of the depth map.

recovery framework [67] enables us to achieve high depth resolution relative to the speed of the sampling at the photodetector. We demonstrate this through simulating longer source pulse width by smearing the timing histograms to correspond to a source four times as slow as the source in the experiments.

Additionally, we address the case of recovering depth information when the pulse width of the source is longer than the time-difference corresponding to the minimum distance between objects. Techniques such as those implemented in [47] will fail to resolve depth values from a returned signal that suffers interference of information between nearby objects. Achieving range resolution higher than that possible with inherent source bandwidth limitations is an important contribution made possible by the parametric recovery process introduced in this work.

4.4 Summary

We have described a depth acquisition system that can be easily and compactly assembled with off-the-shelf components. It uses parametric signal processing to estimate range information followed by a sparsity-enforcing reconstruction to recover the spatial structure of the depth map.

LIDAR systems require N^2 measurements to acquire an $N \times N$-pixel depth map by raster scanning and TOF camera requires N^2. Our framework shows that measurements (or patterns), M, as low as 12% of the total number of pixels, N^2, provide a reasonable reconstruction of a depth map. This is achieved by modeling the reflected scene response as a parametric signal with a finite rate of innovation [67] and combining this with compressed sensing-style reconstruction. Existing TOF cameras and LIDAR techniques do not use the sparsity inherent in scene structure to achieve savings in number of sensors or scanning pixels.

We also achieve high range resolution by obtaining depth information through parametric deconvolution of the returned signal. In comparison LIDAR and TOF that do not leverage the parametric nature of the reflected signal are limited in range resolution by inherent

source pulse width, i.e., the use of a longer pulse width would make it infeasible to recover depth information and hence spatial information correctly. The compressive LIDAR framework in [47] is also limited in range resolution by the source-detector bandwidths, that is, the use of a source with longer pulse width would make it challenging to resolve depth information and hence spatial correspondences correctly.

The processing framework introduced in this chapter solves a single optimization problem to reconstruct depth maps. In contrast, the previous systems such as the system demonstrated in [47] relies on gating the returned signals in *a priori* known range intervals and hence solves as many optimization problems as there are depths of interest in the scene. Consequently, direct limitations are lack of scalability in the presence of increasing depth values and inaccuracies introduced by insufficient knowledge of range intervals. Additionally, the robustness constraint used in our optimization problem is also key to jointly reconstructing the depth map using a single optimization problem to recover a depth map with a smaller number of patterns.

Our experiments acquire depth maps of real-world scenes in terms of object sizes and distances. The work presented in [26] focused on objects of smaller dimensions (less than 10 cm) and at shorter ranges (less than 20 cm). The experiments in this chapter are conducted at longer ranges (up to 2.1 m from the imaging device) with no assumptions on scene reflectivity and more importantly at low light levels. We also address the case when transmissive occluders are present in the scene. In [26], illumination patterns were projected on to the scene with a spatial light modulator. When these patterns are projected at longer distances they suffer distortions arising from interference. The setup described in this chapter uses patterns at the detector thereby implicitly resolving the aforementioned challenge in patterned illumination.

4.4.1 Limitations

Performance analysis for our acquisition technique entails analysis of the dependence of accuracy of depth recovery and spatial resolution on the number of patterns, scene complexity

and temporal bandwidth. While the optimization problems bear some similarity to standard compressed sensing problems, existing theory does not apply directly. This is because the amplitude data for spatial recovery is obtained after the scene depths are estimated in step 1 which is a nonlinear estimation step. The behavior of this nonlinear step in presence of noise is an open question even in the signal processing community. Moreover, quantifying the relationship between the scene complexity and the number of patterns needed for accurate depth map formation is a challenging problem. Analogous problems in the compressed sensing literature are addressed without taking into account the dependence on acquisition parameters; in our active acquisition system, illumination levels certainly influence the spatial reconstruction quality as a function of the number of measurements. Analysis of trade-offs between acquisition time involved with multiple spatial patterns for the single-sensor architecture and parallel capture using a 2D array of sensors (as in time-of-flight cameras) is a question for future investigation. The main advantage of our proposed system is in acquiring high resolution depth maps where fabrication limitations make 2D sensor arrays intractable.

Chapter 5

Mime: Low-Power Mobile 3D Sensing

This chapter focuses on three dimensional sensing with a new set of constraints. We aim to reduce computation and hardware overheads, and integrate sensing with mobile use cases. We take an application-specific approach to sensing – specifically our application domain of interest is the use of touchless hand gestures in mobile devices. Our context of mobile devices includes battery powered devices that have limited touch display size – handheld smart phones, tablets and wearables such as watches and glasses. Existing state-of-the-art 3D sensors cannot be embedded in mobile platforms because of their prohibitive power requirements, bulky form factor, and hardware footprint. The computationally intensive depth acquisition techniques and experiments described in Chapters 3 and 4 present an additional hardware overhead of a projector-like device. Moreover, for application specific sensing with direct implications to the user interface of a mobile device, we are interested in understanding supported mobile use cases. For this purpose, real-time sensing is an important performance metric. This chapter focuses on developing a real-time sensing framework while meeting the aforementioned mobile constraints. It differs from the computationally intensive problem in Chapters 3 and 4 in important ways:

- Spatial patterning: Previous work has used spatially-patterned illumination with a

piezoelectric spatial light modulator (SLM) [26] or spatially-patterned measurement with a digital micromirror device (DMD) [72] to obtain transverse spatial resolution. Here, we require neither an SLM nor a DMD.

- Far-field assumption: The mathematical model in Chapters 3 and 4 employs a far-field assumption that is valid for planar facets that occupy a small fraction of the sensor field-of-view (FOV). This simplifies the parametric form of the impulse response to a sum of trapezoidal functions. Here, the challenges of dealing with the more general form of the scene impulse response are considered.

By avoiding the use of an SLM or DMD, we greatly reduce the hardware cost and size.

Outline

The rest of this chapter will introduce the compact, low-power, application specific sensor, Mime. We will expose the design considerations and constraints derived from our applications of interest in Section 5.1 followed by a technical overview, comparison of technical differences with existing sensing approaches (Section 5.2). Section 5.3 discusses the 3D localization problem formulation and solution while Section 5.5 describes the hardware implementation. We also present a summary of performance evaluation in Section 5.6 and gesture recognition capabilities (Section 5.7). Preliminary interaction techniques are presented in Section 7.2 which will build the foundation for details in Chapter 7. This chapter contains material that also appears in [28].

5.1 Design considerations

Mobile devices in various form factors have become our best digital swiss army tool; users can perform a wide variety of computing and communication tasks through these devices. Currently, touch-screen input is the primary interaction modality for smart devices which require a display no matter how small. Recently, head mounted displays (HMDs) have gained

widespread attention in anticipation of the launch of several consumer-priced units such as Google Glass[1] and Oculus Rift.[2] Historically, hand-held, point-and-click controllers [73], one-handed keyboards like the Twiddler [74], and voice commands [75] have been the primary modes of interaction with HMDs.

Flat screen touch interfaces do not fully take advantage of human dexterity. Touch indeed has its own set of inherent limitations – it requires the user to be in constant contact with the device, touching the screen for input occludes the display, and even simple tasks like menu navigation require tedious, repetitive actions. Equipping users with better input tools for more complex and visually demanding tasks will be important in enabling new applications and making the interaction experience more intuitive and efficient. Further, in the realm of wearable devices, joysticks and keypad controllers are external to the HMD unit; these hand-held controllers are often inconvenient and undesirable in free-form interaction scenarios and mobile settings. Voice-activated control offers limited input functionality, its accuracy greatly varies from user to user and depends on ambient noise levels. Voice input also raises user-privacy concerns when used in public spaces.

Short range 3D gestural control meets several of the aforementioned challenges and limitations in touch interfaces and wearable device input techniques. An input technology intended for mobile handheld or wearable device control and interaction should ideally possesses the following characteristics:

- **Technical**: High accuracy, low power, low latency, small size, daylight insensitivity, and robust performance in cluttered, noisy and fast-changing environments.
- **User experience**: Interacting with the device should be intuitive and should not induce fatigue upon prolonged use. The input device must be able to support both motion- and position-controlled gestures in 2D and 3D.
- **User convenience**: The sensor should be embedded within the device to enable unencumbered user interaction. The user should not be required to wear markers [24] or external sensors [22, 25] or carry additional touch pads.

[1] Project Glass. www.google.com/glass
[2] Oculus VR. www.oculusvr.com

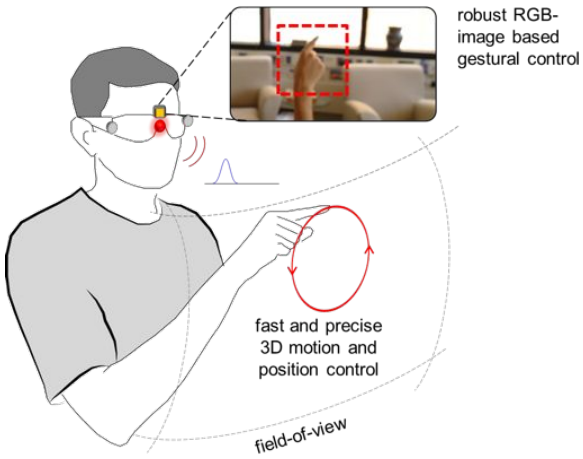


Figure 5-1: Mime combines a new TOF sensor for precise 3D tracking with RGB-image based gesture recognition.

To meet these constraints, we introduce Mime – a compact, low-power 3D sensor for short-range and single-handed gestural control of mobile and wearable devices. Mime provides fast and accurate 3D gesture sensing. The sensor's performance derives from a novel signal processing pipeline that combines low-power time-of-flight (TOF) sensing for 3D hand-motion tracking with RGB image-based computer vision algorithms for shape-based gestural control (see Fig. 5-1).

The Mime sensor is built using off-the-shelf hardware and is easily reproducible. As shown in Fig. 5-2, it comprises three unfocused, baseline-separated photodiodes; an omnidirectional, pulsed light-emitting diode (LED); and a standard RGB camera. Mime can be embedded in the HMD unit, mobile phone or tablet, or attached as a peripheral, thereby eliminating the need for markers, hand-worn sensors, or mobile controllers.

5.2 Technical overview and comparison

In this section, we briefly overview Mime operation and highlight key technical distinctions from RGB cameras and depth sensors used in HMD input and control.

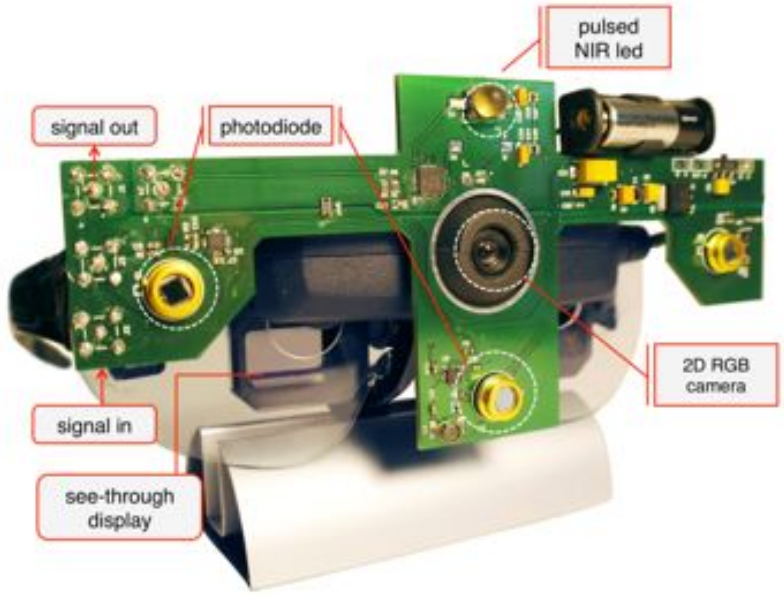


Figure 5-2: The battery powered Mime sensor prototype.

5.2.1 Operation and assumptions

The Mime hardware comprises two modules:

1. A low-power time-of-flight triangulation module built using a pulsed LED and a linear array of three photodiodes.

2. A standard RGB camera module.

Mime operates by first using TOF triangulation for accurately localizing the 3D hand position in the sensor field-of-view (FOV) (see Fig. 5-3a). This is based on assuming interaction with a single hand in the sensor FOV which produces a sharply-localized return time pulse. This is not unreasonable since, in typical interaction scenarios, a user does not have objects in close proximity to his or her head – other than possibly a smooth surface like a wall, which results in a smooth temporal response and does not disrupt hand localization.

Once this region of interest (ROI) is identified, the corresponding RGB image patch is processed to obtain detailed hand gesture information using well-known computer vision techniques (see Fig. 5-3b).

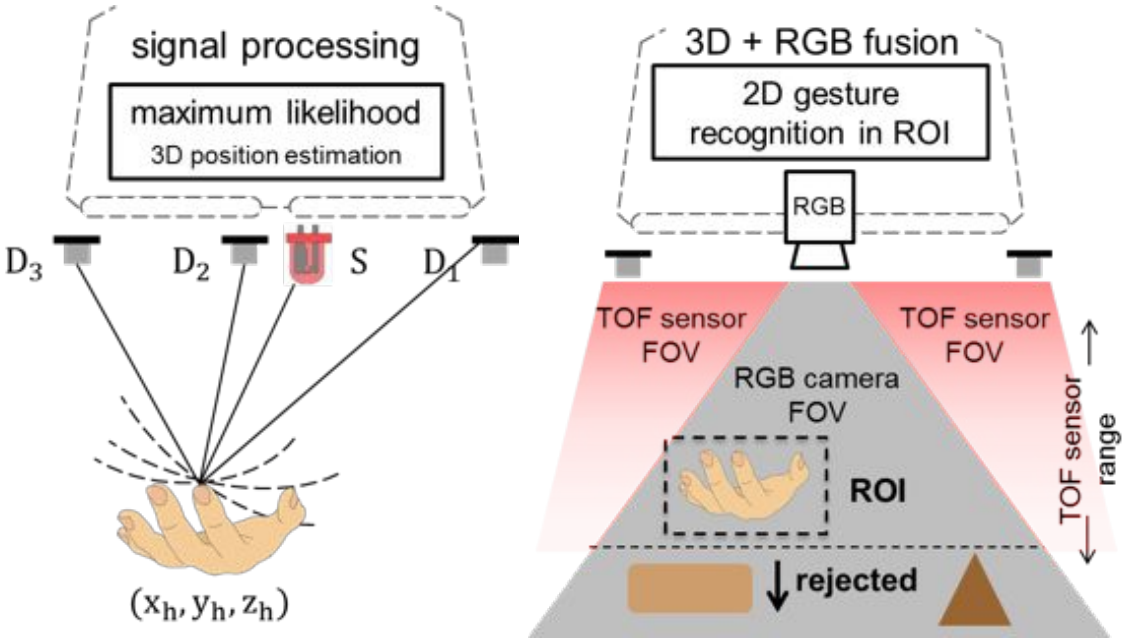


Figure 5-3: The signal processing pipeline for Mime. (a). The scene is illuminated using a pulsed LED and the backscattered light is time-sampled using 3 photodiodes. The time samples are processed using a maximum likelihood estimation framework to acquire 3D hand coordinates. (b) These coordinates are further used to identify a region-of-interest in the RGB image, and process it to recognize hand gestures.

Mime is intended for short-range gesture sensing with a $0 - 4$ ft working range. This is adequate for close-to-body interaction, and essential for avoiding fatigue upon prolonged use.

5.2.2 Key technical distinctions

Hybrid RGB and TOF sensing: RGB image-based gesture input – including stereo methods – uses compact, low-power and ubiquitous 2D cameras; however, it is inaccurate, unreliable, and quickly fails in cluttered environments containing objects of different shape, color, and texture. In contrast, gesture recognition using depth cameras is robust and accurate even in complex scenes; however, 3D sensors that use active illumination are power intensive, sensitive to daylight, and require large heat sinks, and hence they currently are not used in mobile devices.

Mime is neither a general-purpose depth camera nor an RGB image-based computer vision system. It combines the advantages of RGB image-based techniques for gesture control and TOF-based 3D sensing while mitigating their respective disadvantages at the expense of generality.

Application-specific sensing: Mime is an application-specific sensor – it is only intended for single-handed gestural interaction. It sacrifices generality of use to simultaneously satisfy performance criteria (high precision and low latency) and technical constraints (low power and compact form factor). Thus, like the Leap Motion Controller and Digits [22], Mime is developed around the design philosophy of sacrificing generality for performance or power.

Disrupting the computer vision pipeline: The conventional vision pipeline involves capturing a full-resolution color image or depth map using a 2D sensor array, followed by processing it to identify and extract features of interest. Mime disrupts this standard processing pipeline for the specific problem of single-handed 3D gestural recognition. Mime directly senses the feature of interest, i.e., the 3D hand position, using a three-pixel TOF sensor. It further uses this 3D information to identify and process an ROI in the RGB image to extract finer features like orientations of fingers. In essence, Mime uses coarse 3D information to improve the accuracy and robustness of traditional RGB image-based gestural recognition by rejecting outliers and false positives.

Depth super-resolution: Mime's novel TOF signal processing framework achieves precise 3D localization and accurate range resolution relative to hardware specifications. The key step in this processing pipeline is 3D triangulation. To achieve fine range resolution relative to working range, conventional TOF-based triangulation systems require a combination of very short pulses, a large baseline, and an array with large number of detectors. Mime has a small baseline of $5 - 7$ cm compared with the working range of $0 - 1.2$ m; it also uses a broad light pulse and a matched photodetector with pulse width of 200 nanoseconds, and a sub-Nyquist sampling bandwidth of 5 MHz. Despite these low hardware specifications, Mime achieves centimeter-accurate 3D localization. This super-resolution is made possible through physically-accurate signal modeling and novel processing detailed later.

Daylight insensitivity: Like other mobile devices, smart wearables are intended to be

used both indoors and outdoors. Strong ambient light (such as daylight) causes structured light sensors like Kinect or Leap Motion Controller to completely fail, and it significantly degrades the performance of TOF cameras even though they have background cancellation technology [3]. Moreover, mobile devices use involves constantly-changing light conditions, to which the aforementioned devices are sensitive as well. Mime's signal processing only makes use of high-frequency information, enabling it to be robust to daylight and light fluctuations by rejecting low-frequency ambient light.

5.2.3 Comparisons with Mime

Fig. 5-4 shows the accuracy vs. power trade-offs comparing Mime with other real-time sensors useful for HMD interaction that are compact and enable unencumbered interaction. Compared with these other sensing modalities, Mime offers fast and precise 3D gesture sensing at low power.

Unencumbered input is an important user interface design consideration for practical applications and daily use cases. Fig. 5-5 compares Mime with other input techniques on performance vs. encumbrance axes. Along with other 3D gestural control techniques, Mime offers high performance and unencumbered interaction, with the added advantage of being embedded in the bezel of the HMD unit.

5.3 Mime time-of-flight module for 3D hand localization

We now describe Mime's signal processing pipeline, which is based on parametric modeling of scene response described in Chapter 3. The first step is the estimation of 3D hand position using time-resolved measurements.

Localization setup: As shown in Fig. 5-6, there are three baseline-separated photodiodes, D_1, D_2 and D_3, with a total baseline length L. An incoherent, near infrared (NIR), pulsed light source, S, with repetition period T is co-located at the origin with D_2. The light source and photodiodes are omnidirectional and lack spatial resolution.

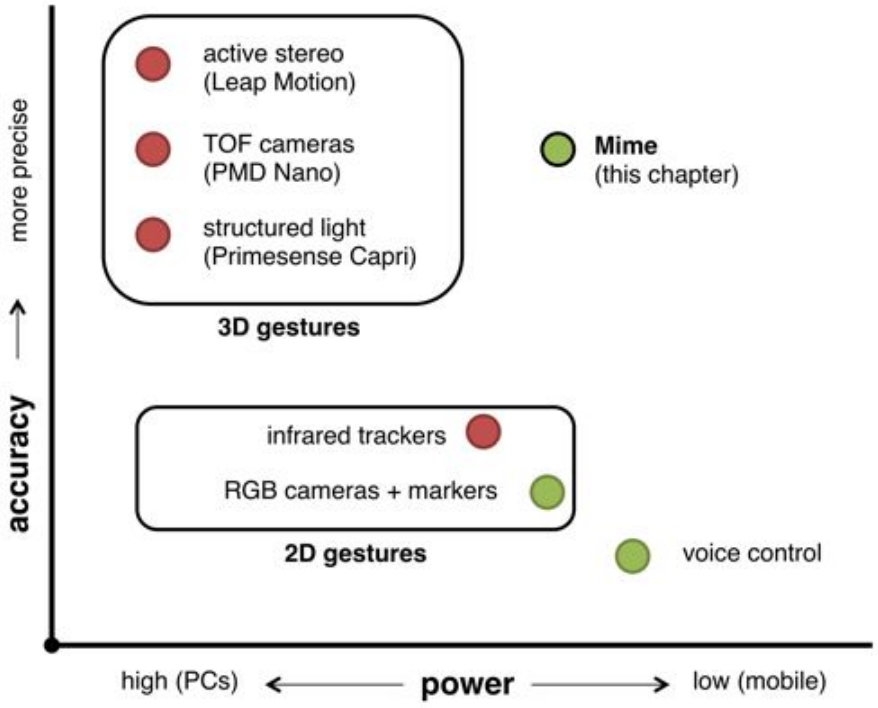


Figure 5-4: Accuracy vs. power comparison for compact, real-time HMD input technologies. The red dots denote devices with high daylight sensitivity, and green dots denote devices that operate under strong ambient light.

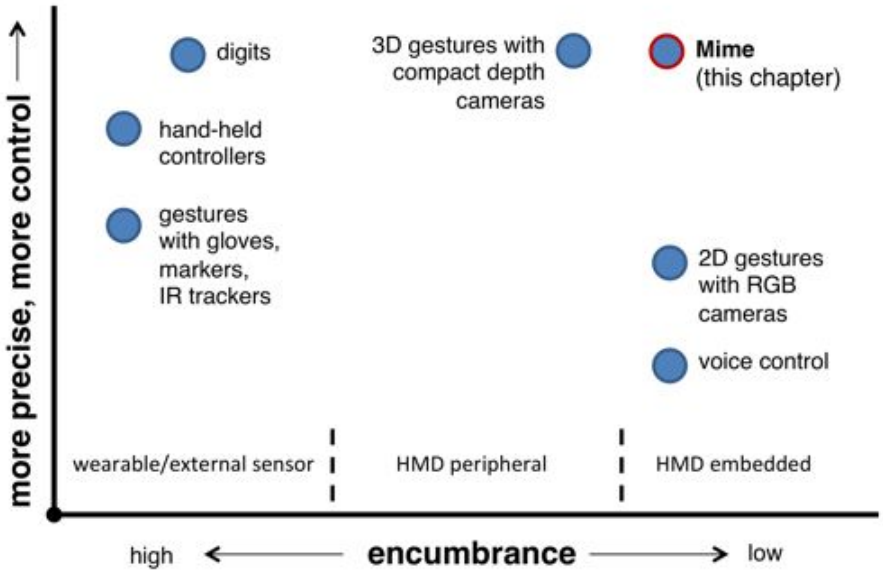


Figure 5-5: Performance vs. encumbrance comparison for HMD input technologies.

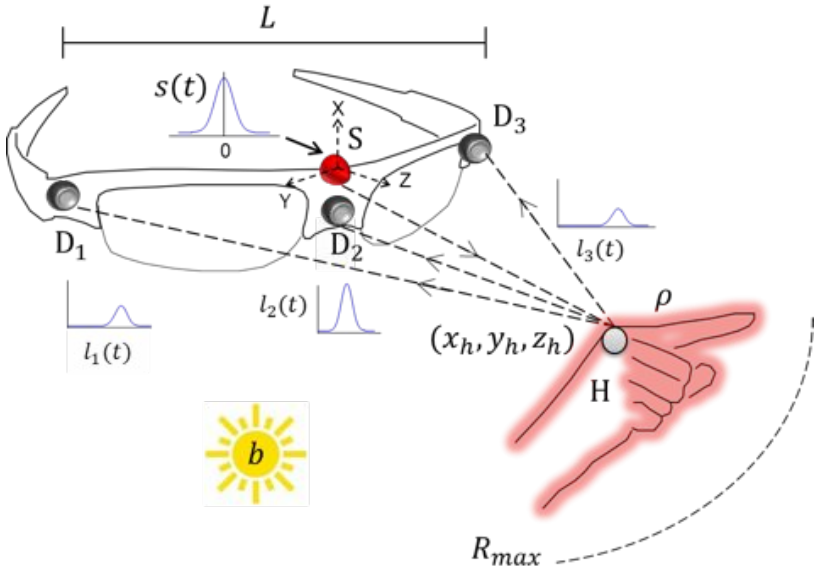


Figure 5-6: The Mime TOF sensing setup indicating the system variables and the geometric arrangements.

The user's hand, denoted by H, is considered to be a diffuse reflector with reflectivity ρ at unknown coordinates (x_h, y_h, z_h). For numerical reasons, we find it convenient to operate in spherical coordinates (R_h, θ_h, ϕ_h) rather than Cartesian coordinates. We denote the maximum range by R_{max}, so $0 \leq R_h \leq R_{max}$. Since Mime is a front-facing sensor, we also have $0 \leq \theta_h$, $\phi_h \leq \pi$. The placement of light source and sensors on a horizontal line creates two-fold vertical symmetry (i.e., even symmetry around $\phi = \pi/2$).

The hand is effectively modeled as a point source since temporal variations attributed to reflected light coming from different regions of H are at high frequencies relative to Mime's low system bandwidth and sub-Nyquist sampling rate.

The outputs of the photodiodes are digitized with a sampling period T_s. The analog-to-digital converters and LED pulsing are synchronized to measure absolute time delays.

Signal modeling: Denote the illumination pulse shape by $s(t)$. The light backscattered by the user's hand is incident at the photodiodes. Let $l_i(t)$ denote the scene impulse response from the source to photodiode i for $i = 1, 2, 3$. Using Lambert's law for diffuse reflectors we

have

$$l_i(t) = \frac{\rho}{4\pi^2 \|HS\|^2 \|D_i H\|^2} \; s\left(t - \frac{\|HS\| + \|D_i H\|}{c}\right) + b,$$

where $\|\cdot\|$ denotes the Euclidean distance between two points in 3D space, c denotes the speed of light, and b is the contribution due to sunlight or other ambient light sources. Note that we have not included the effect of the cosine factor from Lambert's law because the change in perceived luminance is dominated by radial fall-off.

Sampling and background rejection: Let $h(t)$ denote the impulse response of the three identical photodiodes. We apply an analog notch filter with impulse response $f(t)$ at the output of the photodiode to filter out background signal at low frequencies (less than 100 Hz). The digital samples measured with sampling period T_s are

$$r_i[n] \;\; = \;\; \frac{\rho}{4\pi^2 \|HS\|^2 \|D_i H\|^2} \; g\left(nT_s - \frac{\|HS\| + \|D_i H\|}{c}\right),$$

$n = 1, \ldots, N = \lfloor T/T_s \rfloor$, where $g(t) = f(t) * h(t) * s(t)$.

Size, shape, orientation, and skin color of human hands: The reflectivity ρ includes the effects of both skin color and geometric parameters. Mime uses a fixed value of ρ obtained through calibration rather than an estimated value. The color factor is approximately constant because the illumination is at NIR wavelength [76]. The use of a single value of ρ despite variations in geometry is supported experimentally; radial falloff is the dominant cause of signal attenuation and a fixed value of ρ was sufficient to achieve desirable 3D localization performance irrespective of the variability in hand sizes and shape changes during gestural interaction. We also found that Mime's 3D localization accuracy decreases for small hands, simply because of decrease in signal-to-noise ratio (SNR).

Maximum likelihood (ML) hand localization: Given ρ, L, $s(t)$, T, T_s, and the $3N$ noisy data samples $\{r_i[n]\}_{n=1}^{N}$, $i = 1, 2, 3$, we would like to estimate the 3D hand position (R_h, θ_h, ϕ_h). We assume the noise is additive Gaussian, yielding a nonlinear least square estimation problem.

$$(\hat{R}_h, \hat{\theta}_h, \hat{\phi}_h) = \underset{P=(R,\theta,\phi)}{\arg\min} \sum_{i=1}^{3} \sum_{n=1}^{N} \left\{ r_i[n] - \frac{\rho \, g\left(nT_s - \frac{\|PS\| + \|D_i P\|}{c}\right)}{4\pi^2 \|PS\|^2 \|D_i P\|^2} \right\}^2$$

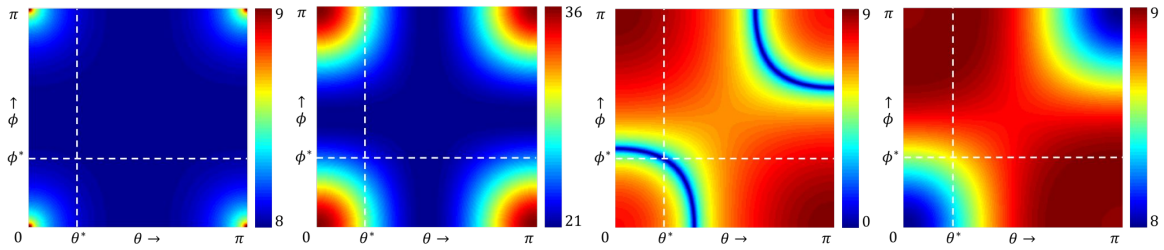


Figure 5-7: Slices of the 3D likelihood function for four values of radial distance R when the true value is $R^* = 0.46$ m. Parameters from the hardware prototype are used: illumination by a Gaussian pulse with duration 200 ns, baseline $2L = 10$ cm, $T_s = 30$ ns, $T = 100$ μs, and $SNR = 20$ dB.

The hand position that minimizes the cost function is chosen as the estimate. This minimization is implemented using gradient descent in spherical coordinates that is initialized using a minimization on a coarse grid; we found the use of spherical coordinates to lead to a simpler cost function with fewer local minima. To further speed up computation, we keep track of previous estimates using a fast Kalman tracker; the estimates from the Kalman tracker are used to narrow the grid search.

Note here that even in the noiseless case, the problem is degenerate because of two-fold vertical symmetry. This appears as symmetry in constant-radius slices of the likelihood function, as shown in Fig. 5-7. The degeneracy could be alleviated by adding a fourth photodiode at an off-axis location. Instead, we resolve this degeneracy by using the RGB camera to check for the presence of human hand in the ROI corresponding to the two degenerate solutions. Aside from the two-fold symmetry, fine vertical resolution is achievable even though the light source and sensors lie on a horizontal line.

The estimated 3D coordinate could lie anywhere on the hand surface, depending on the hand shape, position, and gesture. While this may seem a limitation, the position within the hand remains stable through hand motions, so the gesture identification and tracking performance is not affected.

Comparison with classical source localization: Conventional source localization based on TOF measurements using a detector array is solely based on time delay estimation; it does not incorporate any target reflectivity information. This is important because the

target reflectivity is typically unknown, and inaccurate models or estimates for ρ introduce large errors in position estimation. First the samples $r_i[n]$ are used to estimate time delay or light path length, $\|PS\| + \|D_iP\|$, at each detector. Then, the individual delay estimates are used for 3D coordinate estimation using triangulation. It is well known that the estimation accuracy depends on system bandwidth, baseline, working range and SNR [77]. Mime's TOF processing pipeline achieves better localization accuracy by exploiting target reflectivity rather than discarding it. Incorporating a physically-accurate reflectivity model in the ML solution framework improves 3D position estimation accuracy.

Next, we discuss how the acquired 3D hand coordinates are used with RGB image-based gestural recognition.

5.4 Region of interest RGB-image processing

The camera embedded in the HMD unit captures an RGB image of the scene in front of the user. This image alone may be used for hand gesture recognition using well-known computer vision techniques [78], but this approach has two problems: it does not possess range or depth information required for 3D gestures, and it has poor practical performance. Real-world scenes are complex and contain objects of various shapes, textures and colors which cause the conventional methods to fail.

Mime uses a RGB camera whose FOV is registered with the TOF sensor's FOV. Instead of processing the entire image to identify hand gestures, Mime uses the 3D hand coordinates to first select an ROI in the RGB image and then applies gesture recognition algorithms. Specifically, the ROI data is processed using skin color segmentation, followed by shape recognition, feature extraction, and gesture recognition using a trained SVM classifier [78].

Mime's approach of fusing the 3D information with RGB data via ROI identification is simple, yet powerful. It has two major advantages over traditional RGB gesture sensing: it significantly reduces false positives to drastically improve the robustness and accuracy of computer vision techniques, and computation overhead is reduced since only a small portion

of the image needs to be processed to extract gestural information. In our prototype, the ROI window size is chosen based on the range value, R_h. Closer objects occupy larger pixel area in the 2D camera's FOV and therefore require a larger ROI window.

In the next few sections, we discuss the hardware implementation, calibration, gesture sensing and experiments to validate performance.

5.5 Hardware implementation

The Mime sensor – shown schematically in Fig. 5-8 – is built using the following off-the-shelf optoelectronic components:

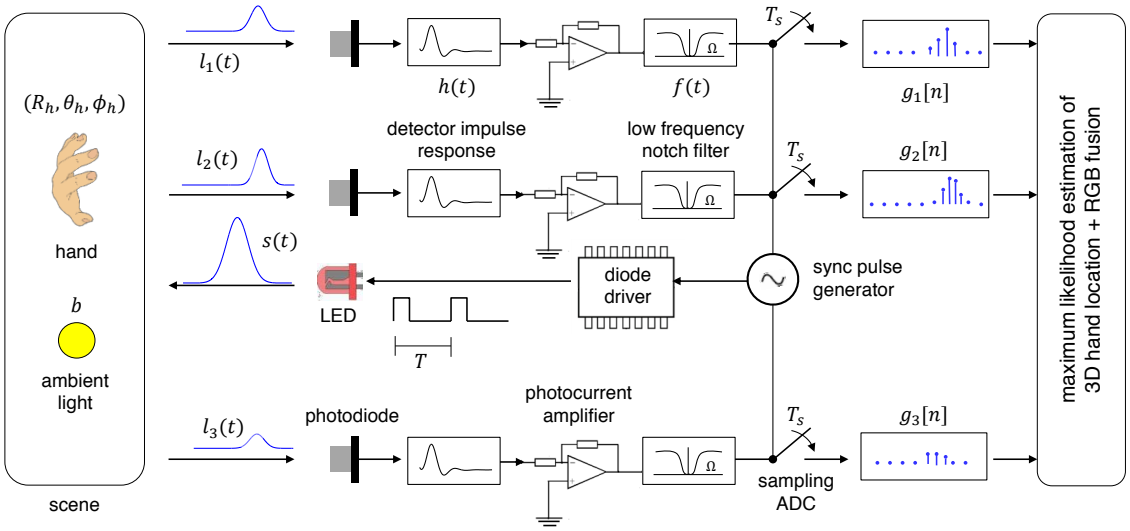


Figure 5-8: Mime data acquisition pipeline showing the system parameters and hardware components.

Light source: Our illumination source is a high-speed NIR (850 nm wavelength) LED with high peak power from OSRAM semiconductor (SFH 4236). The LED has half angle of 20° with diffusion of 90° over the range of interest and is eye safe. We pulse this LED using a laser diode driver from Maxim with high-peak current. High peak power leads to high SNR, and therefore we require less averaging via repeated pulsing. We use a low pulse repetition

frequency of 10 KHz and a pulse width of 200 ns with a rise time of 70 ns (modulation bandwidth ≈ 5 MHz).

Photodiodes: We use fast PIN silicon photodiodes from Thorlabs (part no. FDS100) with a rise time of $50 - 70$ ns (typical). The photodiode output is filtered using an analog notch filter to reject low frequency, followed by signal amplification using a Maxim low-noise amplifier (MAX 3806).

Sampling: We used a low-jitter 4-channel USB oscilloscope with a built-in function generator (PicoScope model 6404b) for sampling the amplified light signal and also to generate the sync signal to measure absolute time delays. We also use a scope bandwidth of 5 MHz (sub-Nyquist compared with pulse modulation) and a sampling period of $T_s = 30$ ns.

The Mime sensor was implemented in a compact form factor with careful circuit design. The sensor was mounted on a pair of Vuzix smart glasses that has a full-FOV see-through display for both eyes and a built-in RGB camera.

5.5.1 Calibration

FOV registration: We registered the FOVs of the RGB camera and the TOF sensor in the transverse or $x - y$ plane by tracking the user's finger in a clutter-free setup using both modalities. Using the $x - y$ coordinates generated by the RGB image-based and Mime's TOF sensor, we computed the rigid body transformation (rotation and translation) using point cloud registration algorithms [78].

System impulse response: The system impulse response $g(t)$ was measured by directly illuminating the photodiodes with the pulsed LED, sampling the output signal finely, and applying parametric curve fitting. We observed that our system response is approximated well by a Gaussian pulse shape.

Hand reflectivity: We estimated ρ empirically by measuring the photocurrent amplitude under strong diffuse NIR illumination to eliminate the effect of radial fall-off. To achieve this, we conducted this measurement in a dark room to eliminate background, and set the

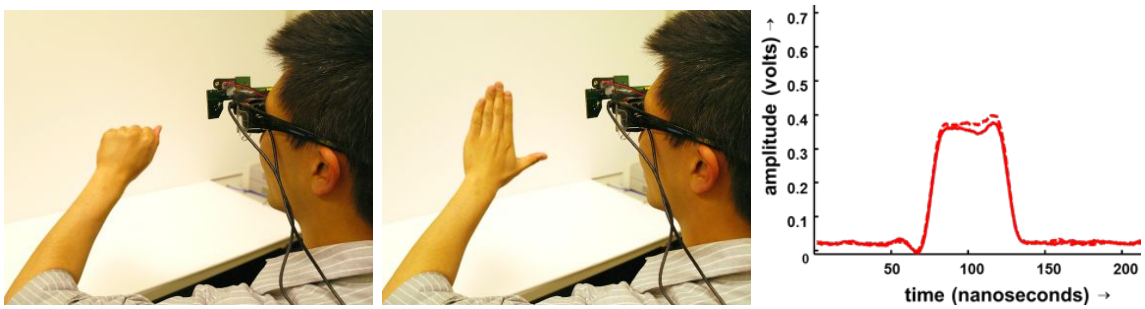


Figure 5-9: Radial fall-off dominates signal amplitude. The responses detected at a photodiode for a fixed hand position and varying hand gestures are very similar. This enables Mime to achieve robust 3D hand localization for different skin colors and gesture shapes.

LED in a high-power continuous illumination mode. Also, as shown in Fig. 5-9, the radial fall-off dominates the signal amplitude over the hand reflectivity and surface area.

5.6 Performance evaluation

Localization accuracy and resolution: For gestural control applications, sensing changes in hand position is critical. We tested resolution, defined as the ability to detect small changes in hand position as it moves across the sensor FOV. Fig. 5-10 shows plots of the data captured at two close hand positions. Although the variation in data is noticeable, it is still small. But using the ML estimation framework discussed earlier, Mime is able to successfully resolve this challenging resolution test case. Fig. 5-10 also demonstrates the two-fold vertical symmetry in our system – the responses from the top and bottom halves are identical. The symmetry is resolved using the RGB camera. Adding a fourth, off-axis sensor would also allow us to resolve the ambiguity.

Daylight insensitivity: In Fig. 5-11, we show outdoor hand tracking in bright daylight. The depth is coded in the color and size of the tracked squares. Mime's accuracy and frame rates remain unaffected using our hardware-based low-frequency rejection. Notice the accurate performance despite the presence of pedestrians in the camera's FOV.

Power and working range: The bulk of the power in a system with active illumination is consumed by the light source. Mime's TOF illumination unit consumes 25 mW and

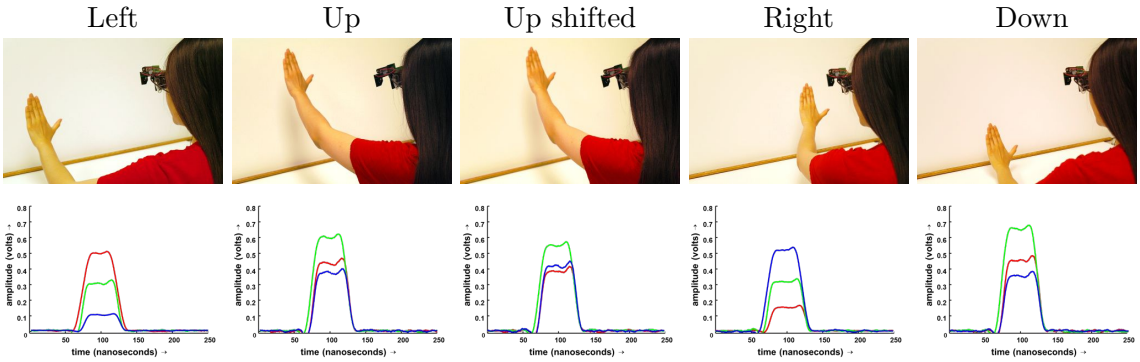


Figure 5-10: Sensor data visualization for varying hand positions. The red, green, and blue curves are the responses at the left, center, and right photodiode, respectively. Note that the sensor closest to the hand has the strongest response and least time shift. The Mime sensor has a vertical symmetry – the responses corresponding to *Up* and *Down* positions are very similar despite a large hand displacement. However, in the top and bottom vertical halves, Mime achieves perfect localization as demonstrated by significant changes in detected response as the hand moves from the *Up* to the *Up shifted* positions.

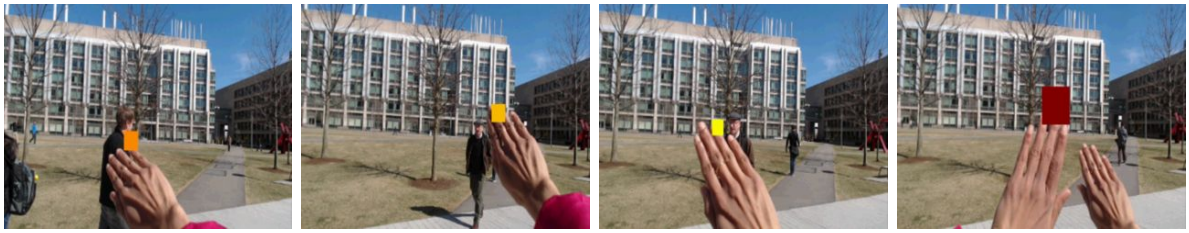


Figure 5-11: Mime operation in strong sunlight conditions.

the three amplified photodiodes require a total of 20 mW. Greater working ranges require stronger illumination and hence more device power. For the same working range of 4 ft, PMD's depth camera consumes 3.5 W, compared with the Mime TOF module's 45 mW total power budget. Such low powers make Mime an ideal candidate for mobile implementations. The mobile RGB camera consumes an additional 75 mW.

Latency: The Mime TOF system (data acquisition, transfer to memory, and computation) achieves a latency of $8-10$ ms, which translates to 3D localization at more than 100 frames per second (fps) (see supplementary video). When integrated with the RGB camera, the overall frame rate is limited by image capture operating at 30 fps.

Using the Mime sensor, we implemented a number of motion-activated and shape-based gestures.

5.7 Gesture sensing using Mime

Gestures using 3D motion tracking: Mime's 3-pixel TOF sensor accurately captures 3D hand position at high frame rates. We track user's hand movement over time and implement several motion-controlled gestures using Mime's TOF module alone. These are swipe (left-to-right, up-to-down), point-and-click, zoom in and out using depth, circle and arc gestures (see Fig. 5-12 and supplementary video). To verify repeatability, we asked 3 users with different hand sizes to test each motion gesture implemented using our prototype. Mime achieved at least 95% detection accuracy and less than 3% false positive rates for all implemented motion gestures. Extensive user evaluation is a subject of future work.

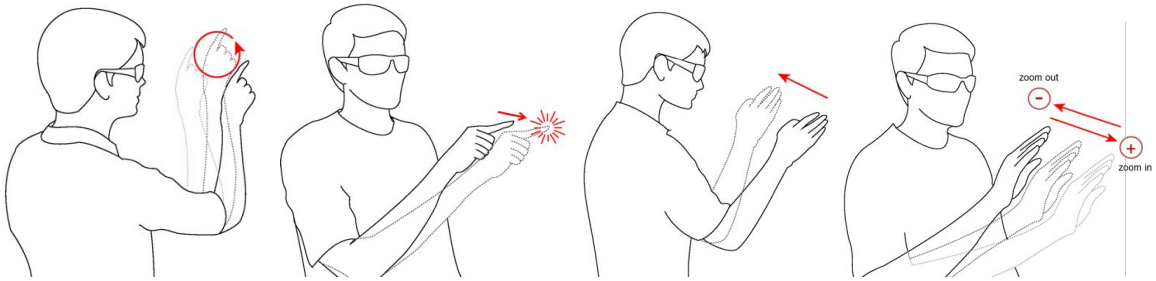


Figure 5-12: Motion-controlled gestures implemented using only the 3D coordinate data acquired by Mime's TOF sensor.

Shape-based gestures using RGB+3D data fusion: Using the RGB+3D fusion approach, Mime adds finer shape-based gestural control. By capturing finger arrangements, we implement gestures like holding fingers in an L-shape to take a picture, C-shape for crop, the OK gesture and thumbs-up (see Fig. 7-4). We also note that Mime does not acquire a complete 3D hand model, as is captured by Leap Motion Controller or Digits [22]. Compared with these sensors, Mime supports fewer 3D gestures and has reduced accuracy; however, it also consumes less power and is not encumbering.

The performance of gesture recognition using 2D color images is well studied [51]. These methods work accurately in the absence of clutter, which causes outliers and false positives. In Mime's processing framework, such outliers are rejected using the 3D and 2D data fusion approach described earlier, thereby leading to an overall increase in system robustness

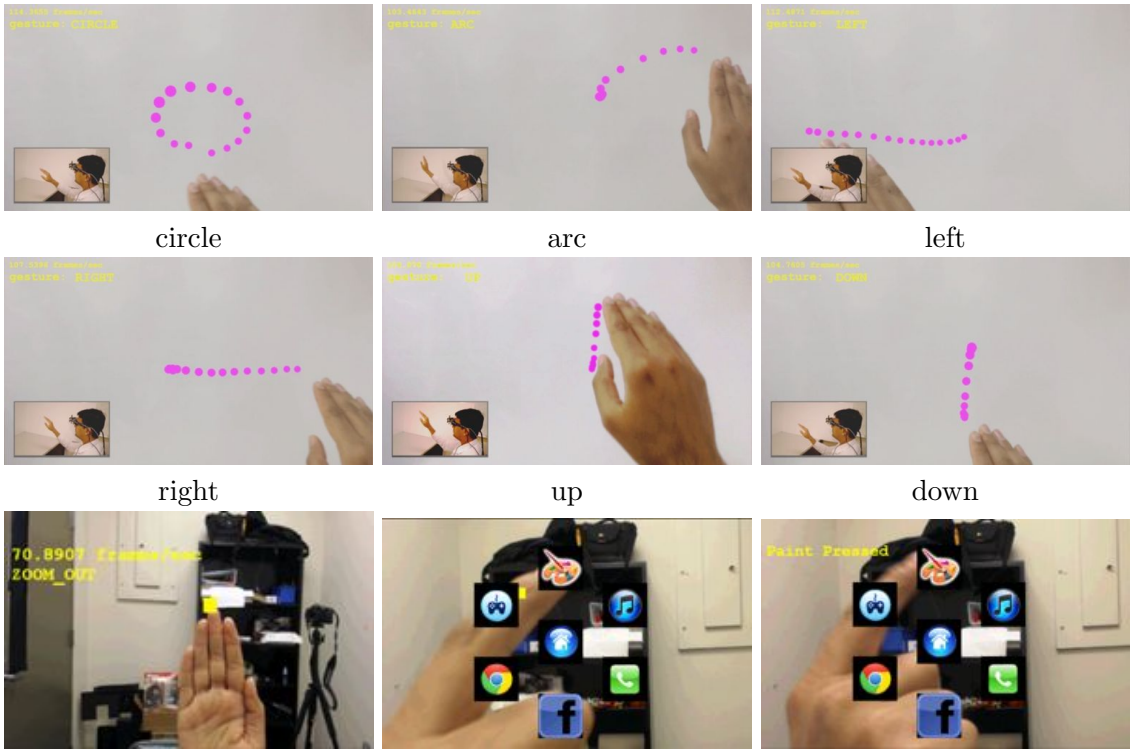


Figure 5-13: Implementation of 3D motion-activated gestures using only the 3D hand coordinates. Mime's motion tracking works accurately in cluttered environments.

and gesture recognition rates. To test how well Mime's clutter rejection works, we placed multiple user hands and faces in the Mime sensor's FOV. Only one of these hands was within the TOF sensor's range, and the goal was to identify the ROI containing this hand. As shown in supplementary video and Fig. 5-14, Mime is able to successfully reject the outliers and increase detection accuracy.

5.8 Limitations

The sensor system introduced in this chapter is designed to provide free-form input to touch limited mobile devices. The Mime sensor is optimized to capture interaction-specific input from the user – specifically, single-hand (and finger) tracking close to the mobile or wearable device. This is an example of the increasing number of application-specific

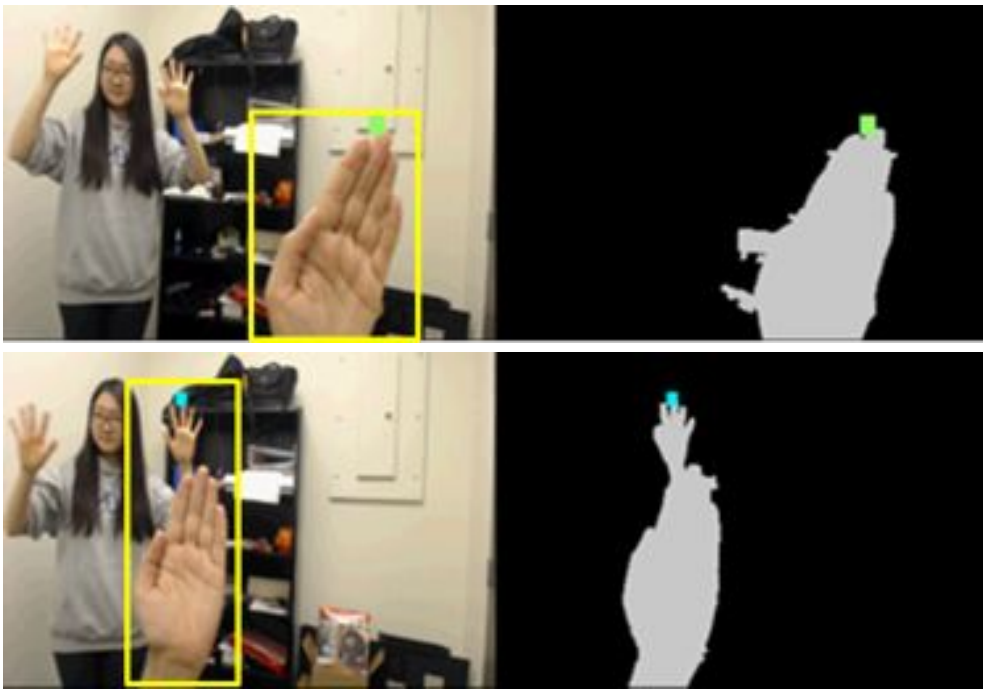


Figure 5-14: The Mime sensor's 3D localization data improves the robustness of RGB-image based gesture recognition. (Top) Notice that the skin colored targets (human hands) in the RGB camera's FOV are rejected because they do not fall inside the identified 2D ROI. (Bottom) Mime's algorithmic framework cannot be used to reject cases in which there are multiple skin-colored objects in the identified RGB ROI.

sensors that trade generality for performance in order to create a better experience for the user. However, a substantive understanding of the limitations of the sensor is necessary to evaluate the gamut of applications this input technique can support. Here, we will go over the limitations of the Mime sensor and optimal conditions of operation to provide the reader a holistic understanding of trade-offs.

1. **Single-handed operation.** The Mime sensor is designed for single-handed multiple-finger operation. This could prove to be a limitation for more complex gestural activity such as digital 3D sculpting.

2. **Multiple finger detection.** Since multiple finger detection relies on the RGB component of the Mime sensor, the performance of the detection algorithm will depend on the RGB sensor and its limitations.

3. **Region of interest definition.** The region of interest in the current Mime implementation is assigned as a function of distance measured under the assumption that closer objects occupy a larger FOV area. This can often lead to the presence of objects other than the hand in the ROI which will impact performance in the case of multi-finger detection.

Chapter 6

Theoretical Extensions for Tracking Multiple Planar Objects

In this chapter, we present both the measurement method and signal processing for sensing 3D structure of scenes comprising a small number of planar objects. Again, similar to the technique presented in the previous chapter, we are performing *task-specific* 3D acquisition or *3D feature acquisition* rather than full depth map generation; thus the sacrifice is the restricted set of scenes.

Several applications of interest, such as multiple hand tracking (see Fig. 6-1) and generating physically-accurate rendered augmentations (see Fig. 6-2), rely on the estimation of a few scene features such as object pose and position. For these applications, the current image processing and computer vision pipeline operates by first capturing a full 2D or 3D image of the scene, then processing to detect objects, and finally performing object parameter estimation. This pipeline works for general scenes but requires significant computation and acquisition resources. It obtains a full-resolution image even though the objects of interest are simple and few. With few sensors and low computational complexity, the acquisition architecture introduced in this chapter can track hands or infer planar object pose and orientation. This obviates high-complexity processing on high-resolution image data.

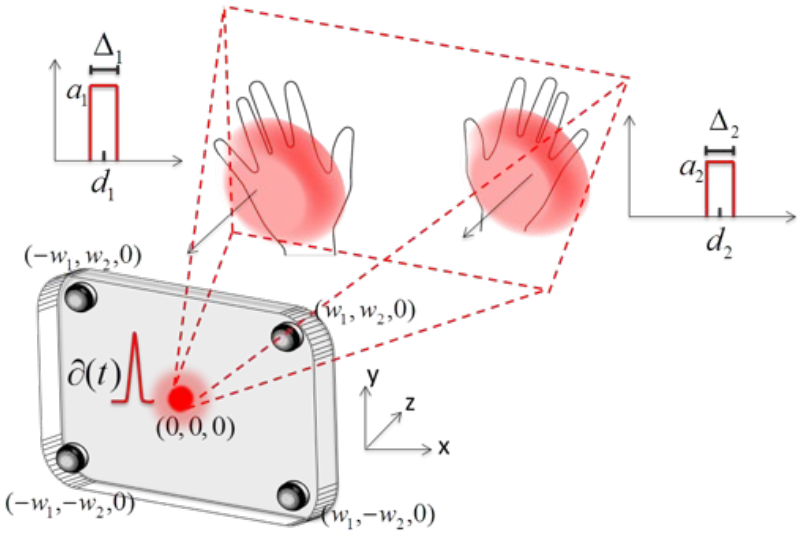


Figure 6-1: Device and scene setup for imaging two hands.

However, we have lost the portion of the architecture of [26, 72] that enables transverse spatial resolution. Instead using 4 sensors, as illustrated in Figs. 6-1 and 6-2, we obtain diversity in the form of 4 source-detector pairs. We exploit this in the manner of [79], though that prior work does not use parametric modeling in any way. This work has some similarities with [26, 72, 79] but addresses different imaging scenarios with different techniques. We show that it is possible to directly recover the scene features of interest – namely object position and pose – by processing samples of the scene impulse response acquired at 4 or more separate detector locations.

The rest of the chapter is organized as follows: Section 6.1 develops a mathematical model of the scenes of interest and the data samples acquired; Section 6.2 describes algorithms for estimating the scene features; Section 6.3 presents simulations of the proposed framework.

This chapter contains material that also appears in [29] and [80].

6.1 Imaging Setup and Signal Models

We consider two scenes of interest shown in Figs. 6-1 and 6-2. These scenes correspond to practical scenarios, namely hand tracking and physically-accurate rendering of 3D models

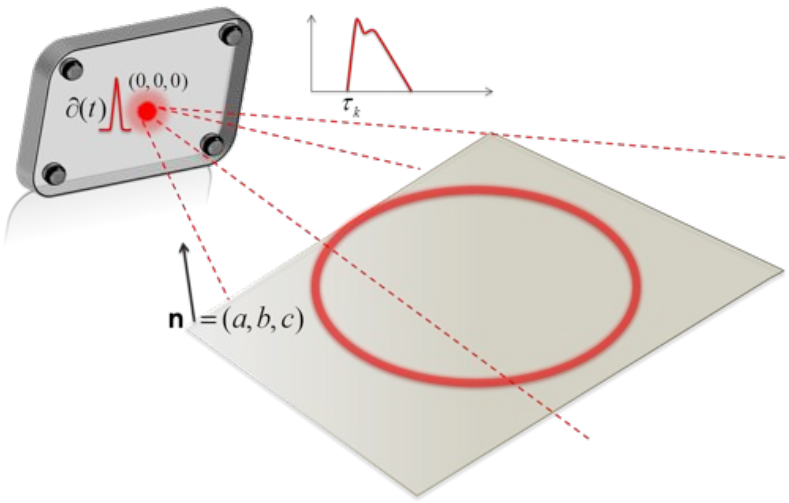


Figure 6-2: Device and scene setup for imaging single plane.

in augmented reality applications. In the first scenario, the features of interest are the 3D locations of the two hands; in the second scenario, we are interested in estimating the pose and position of the plane relative to the imaging device.

Our proposed imaging architecture comprises a single intensity-modulated light source that illuminates the scene with a T-periodic signal $s(t)$ and 4 time-resolved detectors. The intensity of reflected light at Sensor k is $r_k(t)$, $k = 1, \ldots, 4$. The light source and detectors are synchronized to a common time origin. We also assume that the illumination period T is large enough to avoid distance aliasing [37]. To derive the scene impulse response, we let $s(t) = \delta(t)$.

6.1.1 Two Hands

Suppose each of the two hands in the scene occupy a small area in the sensor FOV. Considering each as a small planar facet, it was shown in [26] that the impulse response is well modeled as

$$g_k(t) = a_1 \, \mathrm{B}(t - d_{1k}/c, \Delta_1) + a_2 \, \mathrm{B}(t - d_{2k}/c, \Delta_2),$$

where c is the speed of light, a_i is the reflectance of Object i, d_{ik} is the total length of the path from source to Object i to Detector k, and

$$\mathrm{B}(t, \Delta) = u(t) - u(t - \Delta)$$

denotes the causal box function of width Δ. The scene impulse response is the sum of two box functions that are time shifted in proportion to the respective object distances and scaled in amplitude by the object reflectances. The box function widths are governed by the object poses, positions and areas. As $\Delta \to 0$, we approximate $\mathrm{B}(t, \Delta) \approx \Delta\delta(t)$, so the response for two small objects can be approximated simply as a sum of two scaled, shifted Diracs:

$$g_k(t) = a_1' \, \delta(t - d_{1k}/c) + a_2' \, \delta(t - d_{2k}/c).$$

In this case, the locations and amplitudes of the Diracs constitute the signal parameters we wish to recover.

6.1.2 Planar Scene

Now consider a scene comprising a single plane occupying the entire FOV. Following [79], let $\mathbf{x} = (x_1, x_2) \in [0, L]^2$ be a point on the scene plane, let $d^{(s)}(\mathbf{x})$ denote the distance from illumination source to $\mathbf{x}$, and let $d_k^{(r)}(\mathbf{x})$ denote the distance from $\mathbf{x}$ to Sensor k. Then $d_k^{(t)}(\mathbf{x}) = d^{(s)}(\mathbf{x}) + d_k^{(r)}(\mathbf{x})$ is the total distance traveled by the contribution from $\mathbf{x}$. This contribution is attenuated by the reflectance $f(\mathbf{x})$, square-law radial fall-off, and $\cos(\theta(\mathbf{x}))$ to account for foreshortening of the surface with respect to the illumination, where $\theta(\mathbf{x})$ is the angle between the surface normal at $\mathbf{x}$ and a vector from $\mathbf{x}$ to the illumination source. Using $s(t) = \delta(t)$, the amplitude contribution from point $\mathbf{x}$ is the light signal $a_k(\mathbf{x}) \, f(\mathbf{x}) \, \delta(t - d_k^{(t)}(\mathbf{x})/c)$ where

$$a_k(\mathbf{x}) = \cos(\theta(\mathbf{x})) / \Big(d^{(s)}(\mathbf{x}) \, d_k^{(r)}(\mathbf{x}) \Big)^2. \tag{6.1}$$

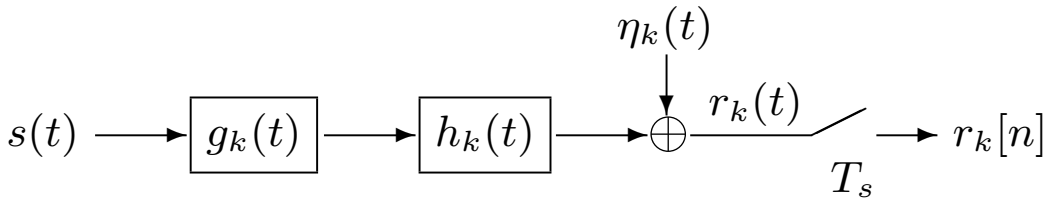


Figure 6-3: Signal flow diagram for signal acquisition at Sensor k.

Combining contributions over the plane, the total light incident at Sensor k at time t is

$$g_k(t) = \int_0^L \int_0^L a_k(\mathbf{x})\, f(\mathbf{x})\, \delta(t - d_k^{(t)}(\mathbf{x}))\, dx_1\, dx_2. \tag{6.2}$$

The intensity $g_k(t)$ thus contains the contour integrals over the object surface where the contours are ellipses. As we will later illustrate and exploit, $g_k(t)$ is zero until a certain onset time τ_k and then well approximated by a polynomial spline of degree at most 2.

6.1.3 Sampling the Scene Response

An implementable digital system requires sampling at the detectors. Moreover, a practical detector has an impulse response, $h_k(t)$, and a Dirac impulse illumination is an abstraction that cannot be realized in practice. Using the fact that light transport is linear and time invariant, we accurately represent the signal acquisition pipeline at Sensor k using the flow diagram in Fig. 6-3.

At Sensor k, we acquire N digital samples per illumination period using a sampling period of $T_s = T/N$:

$$r_k[n] = \left[g_k(t) * h_k(t) * s(t) + \eta_k(t) \right]\big|_{t=nT_s}, \quad n = 1, \ldots, N,$$

where $\eta_k(t)$ represents photodetector noise. Except at very low flux, $\eta_k(t)$ is modeled well as signal-independent, zero-mean, white and Gaussian with noise variance σ_k^2. Assume for simplicity that the 4 detectors have identical responses and noise variances: $h_k(t) = h(t)$ and $\sigma_k^2 = \sigma^2$ for $k = 1, \ldots, 4$.

The top row of Fig. 6-4 shows the continuous-time scene impulse responses for the scenes

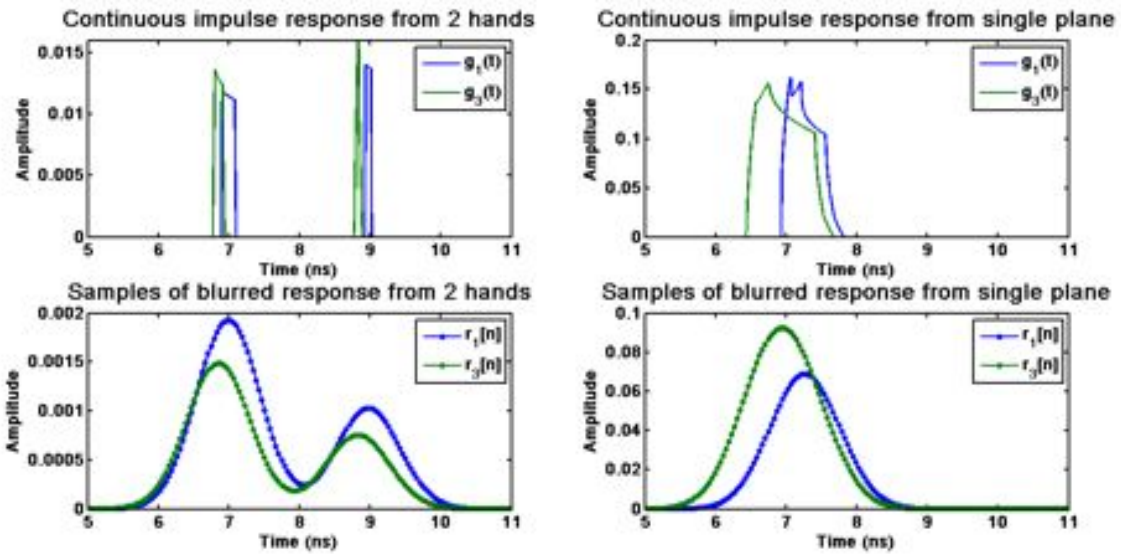


Figure 6-4: Typical continuous-time scene impulse responses $g_k(t)$ and samples $r_k[n]$ for our two types of scenes.

under consideration, and the bottom row shows the samples acquired at the individual detectors in the absence of noise for Gaussian $s(t)$ and $h_k(t)$ specified in Section 6.3. Our goal in the next section is to use the samples, $r_k[n]$, $k = 1, \ldots, 4$, to estimate the desired scene features.

6.2 Scene Feature Estimation

In the case of a scene with two hands, we are interested in estimating the 3D locations of the objects in the sensor's FOV; in the case of a single plane, we wish to estimate the plane position and orientation. In each case, we apply a two-step process to the feature acquisition problem:

1. Use parametric deconvolution to estimate scene impulse response $\hat{g}_k(t)$ from the acquired samples $r_k[n]$.

2. Use the set of estimated signal parameters from the 4 scene impulse responses to recover the scene features.

Note that our proposed technique directly captures the scene features without requiring acquisition of a complete 2D or 3D image.

In both cases, we define our coordinate system relative to the device with the illumination source as the origin and the device lying in the xy-plane. The detectors are located at $(\pm w_1, \pm w_2, 0)$. We choose the imaging direction to be in the positive z-direction so that the sensor's FOV lies in the halfspace where $z > 0$.

6.2.1 Two Hands

Following Step 1 of our two-step process, we can directly estimate the amplitudes and time shifts of the Diracs in the scene impulse response. Since we are trying to localize two objects, we assume model order 2 in our parametric deconvolution scheme and recover the scene impulse response as a sum of 2 Diracs. From the time shifts, we can estimate distances $\hat{d}_{Ak}$ and $\hat{d}_{Bk}$. Note that to recover the spatial locations of the two objects in the FOV, we specifically use the distances $\hat{d}_{ik}$.

Once we have estimates $\hat{d}_{Ak}$ and $\hat{d}_{Bk}$ for each detector k from Step 1, we begin the recovery with Step 2. We first determine which estimated distance corresponds to which object. We can accomplish this by finding the equations describing the 8 total ellipsoids for which the total distance from the source to a point on the ellipsoid and back to receiver k are $\hat{d}_{Ak}$ and $\hat{d}_{Bk}$. In the ideal noiseless case, we can partition the 8 ellipsoids into two disjoint sets of 4 ellipsoids each, with the first set defined by $\hat{d}_{1k}$ and the second set defined by $\hat{d}_{2k}$, such that each set intersects in one unique point. These two points of intersection $\hat{\mathbf{x}}_1$ and $\hat{\mathbf{x}}_2$ are the estimates for the locations of the two objects.

In the noisy case, the two sets will nearly intersect in one point. Define $d_k(\mathbf{x})$ as the total distance traveled by the contribution from point $\mathbf{x}$ in the detector's FOV. To estimate the locations of the objects under noisy conditions, we solve the following optimization problem that finds the point for which the sum of squared differences between total distances to the

point and estimated total distances is minimized:

$$\hat{\mathbf{x}}_i = \arg\min_{\mathbf{x}} \sum_k \left(d_k(\mathbf{x}) - \hat{d}_{ik}\right)^2.$$

These $\hat{x}_i$ are the recovered locations of the two objects. Note that estimates in the ideal noiseless case also satisfy this minimization problem.

6.2.2 Planar Scene

We have seen that the impulse responses of large planar scenes can be modeled as continuous piecewise-polynomial signals with several kinks, or discontinuities in slope. We first note that the kinks directly correspond to spatial locations of scene features (such as nearest points, edges, and vertices), and thus the parameters we wish to recover from the signal are the time locations of the kinks. For Step 1 of our process, we employ the method for recovering piecewise-polynomial signals described in [67,68] to determine both the locations in time and amplitudes of these kinks. Though a typical signal can be seen to have as few as 4 or 5 kinks, recovery in practice was more accurate assuming a higher model order (number of kinks) of 10 or 12 to begin and rejecting kinks with low amplitude. To determine the location and orientation of our large planar scene, we specifically require the time location of the first kink, or onset time τ_k, of the impulse response $g_k(t)$. These onset times correspond to the times at which the light that travels the shortest total path length is incident on the detector. Thus, from each τ_k we can calculate the shortest path length $\hat{d}_k^{min} = c\tau_k$ for each source-detector pair k.

We describe our plane $P(\mathbf{n})$ by the point $\mathbf{n} = (a, b, c)$ on the plane with minimum distance to the origin. The plane is also equivalently described by the equation $\mathbf{n} \cdot \mathbf{x} = \mathbf{n} \cdot \mathbf{n}$. For any plane not passing through the origin, the ordered triple $\mathbf{n}$ uniquely determines both the normal direction and a point on the plane. Let $d_k^{min}(\mathbf{n})$ be the minimum path length from the origin to $P(\mathbf{n})$ and back to Detector k. With the shortest path length $\hat{d}_k^{min}$ for each source-detector pair k, we solve the following optimization problem that finds the plane $P(\hat{\mathbf{n}})$ for which the sum of squared differences between total distances to the plane and

estimated total distances is minimized:

$$\hat{\mathbf{n}} = \arg\min_{\mathbf{n}} \sum_{k} \left(d_k^{min}(\mathbf{n}) - \hat{d}_k^{min} \right)^2$$

The resulting plane $P(\hat{\mathbf{n}})$ is our estimate for the plane. Since we are imaging in the positive z halfspace and have 3 parameters defining P, fitting a plane using only 3 receiver profiles is sufficient. However, incorporating all 4 receivers provides robustness against noise.

6.3 Simulations

We simulated imaging using a device of dimension 25 cm $\times$ 20 cm, which is the size of a typical current-generation tablet device. The illumination source $s(t)$ was a Gaussian pulse of width 1 ns with a pulse repetition rate of 50 MHz (signal period $T = 20$ ns) with $N = 501$ samples per repetition period. To demonstrate the framework for the two cases we examined in this chapter, we considered:

1. two small rectangular planes of dimension 5 cm $\times$ 10 cm (approximately the size of average human hands) fronto-parallel to the device; and
2. a single, large tilted rectangular plane of dimension 50 cm $\times$ 50 cm defined by nearest point and normal direction $\mathbf{n} = (0.6, 0, 0.8)$ relative to the device.

Fig. 6-5 shows signal parameter estimation from Step 1 of our framework. We are able to recover the important times and distances needed for estimating scene features. The time locations d_{ik} of the hands and the onsets τ_k of the large plane are captured accurately. Note that the exact amplitudes of the piecewise-polynomial fit for the scene impulse response of the large plane are not completely preserved due to the mismatch in the model, but the time locations of the kinks are still preserved.

Fig. 6-6 shows the effects of noise on accuracy for localizing each of the two hands in the detector's FOV and similarly the effects of noise on accuracy for recovering plane location

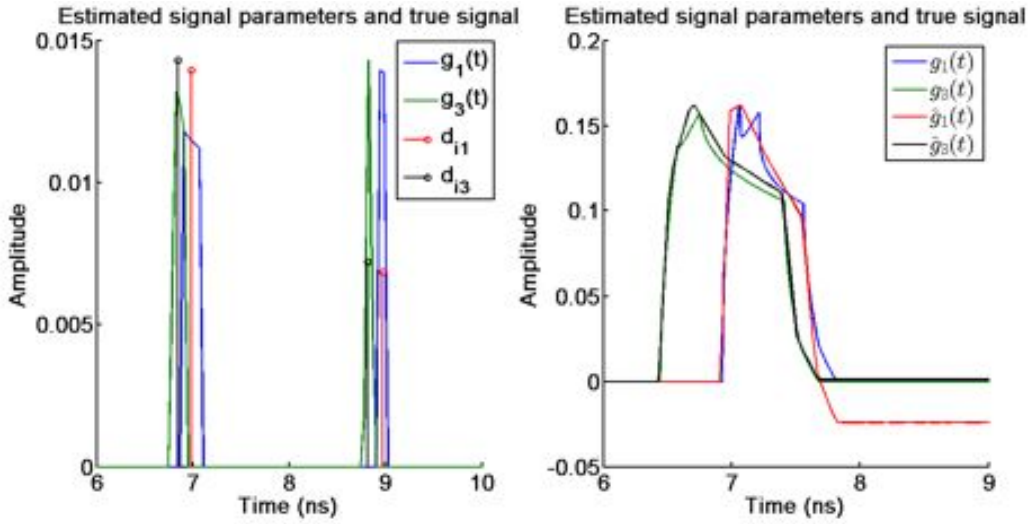


Figure 6-5: Recovered impulse responses of Detectors 1 and 3 from our simulated scenes superimposed with actual noiseless scene impulse responses.

and orientation. We calculated the normalized MSE averaged over 500 trials at each SNR level. We see that the recovery of two hands is more robust to noise than the recovery of the single plane due to the lower complexity and better fit of signal model.

6.3.1 Discussion

This 3D imaging framework demonstrates that we are able to directly estimate features from two scenes of practical interest by recovering signal parameters.

The two-step process is able to estimate the location and orientation of a single large tilted plane by fitting the scene impulse response with a piecewise-linear spline. In addition, we are able to estimate the locations of two small fronto-parallel planes accurately under fairly noisy conditions.

The signal parameter recovery in Step 1 of our framework is somewhat susceptible to noise for large planar scenes due to the model mismatch. In addition, this framework estimates scene features assuming that recovered signal parameters vary according to a normal distribution, which is not necessarily the best fit based on the parameter recovery method. Performance improvements could be achieved by increasing the parametric deconvolution

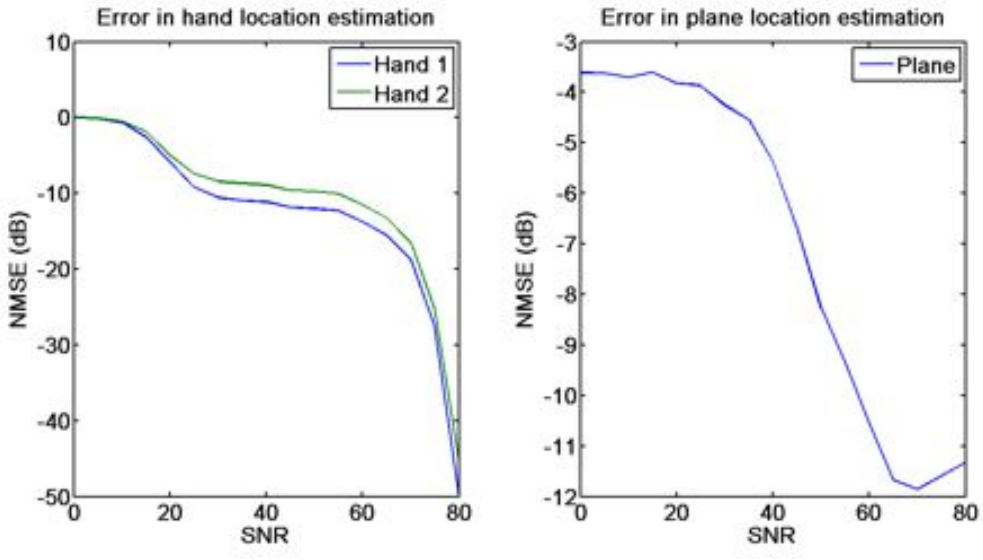


Figure 6-6: Effects of noise on scene feature estimation for (a) two fronto-parallel "hands"; and (b) a single large tilted plane.

performance in Step 1 and incorporating the distribution of recovered parameters when estimating scene features in Step 2.

Chapter 7

Applications and Interaction Techniques

In this chapter we discuss interaction opportunities enabled by the computational sensing frameworks presented in this thesis. Since real-time performance is central to building a useful interaction experience, we will restrict the scope of sensing to the Mime sensor discussed in Chapter 5. In the rest of this chapter, we carefully investigate how we could apply the Mime sensor to user interactions. We present example applications for smart glasses and phones that draw from our observations and design guidelines.

To design interaction techniques with the Mime sensor, we work within the capabilities and limitations of the sensor. Here, we review supported sensing capabilities and constraints that will guide the interaction techniques.

- **Portability:** The low-power attribute of the Mime sensor makes it suitable to mobile devices. A powerful benefit of the sensor is its ability to perform in diversity of lighting conditions – bright sunlight to dark conditions – which is usually a limitation in most optical sensors.
- **Single-handed operation:** The Mime sensor prototype is only capable of sensing a single hand in its current implementation. This is both a limitation and an advantage

for the design of interaction techniques. It is a limitation in that it only supports actions that a single hand is capable of performing. In scenarios where one of the user's hands is occupied, it is advantageous to not require both hands for interaction; this is typical for smart phones and tablets that are usually held in one hand. Additionally, two handed operation of any device requires coordination which sometimes consumes visual attention.

- **Minimal instrumentation:** Using Mime with a smart mobile device only requires retrofitting the device with the sensor. This means the user and her environment are not instrumented which makes the interaction experience easy to build and use.

- **Form-factor:** The baseline separated photodiodes in the Mime sensor present minimum dimensions required for meaningfully mounting or embedding the sensor on a mobile device. Because the longer dimension of baseline separation is at least $10cm$, the frame of a head mounted device or the taller side of a smart phone are ideal locations for placing the sensor.

- **Hand vs. finger:** A unique capability of the Mime sensor's tracking is to interchangeably track a single hand or single moving finger. This allows for seamless transition between the two modes as needed. It can be used for more macro actions (like swiping) or finer actions like accurate pointing and tracking.

The sensor features and constraints expanded above present an ideal fit for the style of *unobtrusive wearable interaction* prescribed by Rekimoto [81, 82]. For wearable devices this implies that the user can be fully engaged in real-world tasks while the wearable device continuously monitors context and input. The Mime sensor is ideal in that it does not require the user to hold it, as is the case with controllers like Twiddler [74]. Unlike glove-based input systems [83] it does not preclude performing other real world tasks with the use of hands. Another guideline is ensuring that input and the device itself is socially acceptable. We address this issue when we discuss types of hand movements that are captured in various applications. The mounting location of the Mime sensor is an important factor in determining how and where input actions take place, consequently influencing social

acceptance. We discuss this guideline and opportunities for addressing it with our system in Chapter 8.

7.1 Input using our hands

Our primary input mechanisms to all our computing devices are centered around our hands. To understand how these natural input tools can be used to support gestural interaction, we will first attempt to categorize how people use their hands. Over years of evolution, we have developed acute control of the motor capabilities of our limbs, particularly our hands. For communicative purposes, our hands play an important non-verbal role. We will now develop a taxonomy of different gesture types.

7.1.1 Pointing gesture

Finger pointing is a natural human gesture universally used to draw attention to an object or location or to provide a spatial anchor for additional supplementary context [84]. This pointing gesture is categorized as a *deictic gesture* – gestures which *indicate real, implied or imaginary objects and are strongly related to their environment.* Deictic gestures tend to be the first communicative gestures infants use. Thus, using finger pointing provides us the most intuitive starting place to build upon; most of our digital devices utilize this pointing action or some derivative of pointing in their interfaces. The mouse or trackpad for example with a visual cursor provide for pointing in desktop environments. Touch devices use finger pointing directly with the visual cursor dissolving to the point of contact directly.

From the above observations our interaction techniques will attempt to use finger pointing in two ways.

- First, we use finger pointing to draw attention to the environment. This acts as a cue for the wearable device to follow the user's interest or intention.

- Secondly, because finger pointing can be tied to a variety of objects or persons, real or imaginary, by being able to point anywhere, any space could literally become a rich input canvas. That is, pointing can be augmented with supporting alternate modalities or information from the object or surrounding region.

7.1.2 Shape-based gestures

The use of hand shapes as gestures is more complex in terms of categorization as well as implied or direct meaning extracted from these gestures. We will go over the main types of hand shapes – either whole hand or specific shapes contributed by hand digits – that are relevant to user input to digital devices. Our categorization reveals three main types; the list is not meant to be exhaustive but will provide the underpinnings for our applications in Sections 7.2 and 7.3.

Symbolic gestures: Some types of hand shapes are inherently *symbolic* possibly with cultural bias. Often these are gestures are highly lexicalized but their cultural meanings leave interpretation dependent on external factors. A good example is the **thumbs-up** gesture meaning good or well done in many cultures but is considered offensive in others. Another example is the **A-ok** gesture made by connecting the index finger and thumb while stretching out the remaining three digits. This gesture typically symbolizes completion. We extend some interaction techniques to utilize these types of symbolic shape based gestures. For our interaction techniques these gestures will translate the meaning attached by the user to the application thereby reducing vocabulary learning challenges.

Letter gestures: This category exposes shapes that represents letters from the English alphabet. We design and discuss this category since it is important in mobile devices that lose physical input keys that correspond to letters. A direct example is representing keyboard shortcuts by the main letter involved. Examples include, creating the 'C' shape to indicate copying, a 'V' sign to indicate paste and so on. The scale of these gestures obviously is quite restricted to only a handful of letters that are easily mapped. The advantage again

is in reduced cognitive overhead that vocabulary learning presents. A related extension to create letter gestures is tracing the specific letter via in-air writing. One the trace has ended the shape that corresponds to the trace will be used to interpret a letter gesture. A good example is tracing 'S' to indicate the shortcut for the *save* command. It is our observation that tracing letters will be easier when the letter can be traced in one motion.

Visual function gestures: This category derives gestures from input elements used in digital applications and physical devices. These gestures are primarily shape based or spatial motion combined with shape. Interfaces to digital applications often virtualize some of our analog (or physical) actions. Thus, they present application specific intuitive actions that provide a rich input vocabulary. An example from a digital application is the picture editing *crop* option visually represented by orthogonal lines spanning a rectangular boundary. A visual function gesture derived from this element is an action where the thumb and index finger of each hand are extended (while other digits are folded) and placed to represent a rectangular boundary. Another example is volume control, where a hand turning gesture would directly map analog volume control via knobs. Alternatively, moving a hand up and down vertical could suggest controling volume but may also represent equalizer control in an audio application. These examples suggest that mappings in the case of visual function gestures are application and context dependent.

7.1.3 Supplementary modalities

The goal of emerging form-factors of wearable devices is to situate the user in real world interactions while providing swift computing tools that augment real world tasks either with information or input capture and manipulation. Often this may mean that text input per se is either inconvenient or undesirable. In hands-free wearable devices, speech is a common input modality. For communicative purposes hand gestures, speech and visual context often occur together. An early example of such multi-modal input was presented in Put That There [85]. It is complex to interpret speech or visual input in isolation. Therefore, we combine the simple yet powerful pointing action with these other modalities to supplement

input to wearable devices. We use speech input along with pointing as explicit input or implied annotation. When used as explicit input, it could be used to convey a more complex input action. Implicit annotation is captured for post-facto input in the form of tags.

Another source of supplementary input is visual information seen by a camera on the device. The real world abounds in rich visual cues. Again, we can use finger pointing to provides glimpses of where the user is looking at and capture those parts from a larger visual canvas. Here, we use the optical camera as a digital eye of the system. We reduce the task of searching through a vast stretch of visual information by using the user's hand movements as a guide to visual interest.

The aim of this chapter is to investigate applications for mobile devices (smart phones, tablets) and wearables (smart glasses) by harnessing natural hand movements. We focus on crafting input that maps to the types of tasks mobile devices and wearables are designed for. From many possibilities of using hand gestures, we limit our interest to single-handed input via finger pointing and shape based gestures from the repertoire presented above. Hand based movement and in some cases with supplementary modalities will be the foundation of our applications presented in sections that follow.

7.2 Interaction techniques with head mounted displays

This section presents demonstrative styles of interaction utilizing the Mime sensor in various real-world HMD use cases. The Mime sensor was mounted on the Vuzix glasses; the application is seen through the display of the smart glasses. For each application, we highlight components of the Mime sensor that are utilized to make the interaction possible.

Augmented menu interaction: We demonstrate navigating an augmented interface through the glasses display (see Fig. 7-1). The user navigates to any element by appropriately moving their hand to the target icon. Activating an application is achieved by controlling the z-dimension, that is, by an analog "push" gesture that is measured through relative z-motion over a highlighted icon. This is a type of **visual function gesture** al-

Figure 7-1: Augmented menu interaction using motion gestures: arc to activate menu, and point-and-click to select item.

though the application does not use hand shape. This example only requires the 3D TOF tracking without any RGB component. The resolution in all three dimensions is sufficiently accurate to achieve real-time, in-air menu navigation. Thus, this scenario is invariant to clutter in the environment, daylight presence, or poor lighting conditions. The in-air point and click requirements for such an on-the-go application are clearly achieved by the Mime sensor.

Using Mime's TOF module we also implemented swipe, zoom, arc and circles which are natural, intuitive gestures for navigating carousel-style interfaces.

Figure 7-2: In-air writing and drawing implemented only using Mime's TOF sensor.

In-air drawing and annotations: In the second use case, we present an in-air drawing interface that tracks and visualizes a user's hand motion to create a paint trail (see Fig. 7-2) as well as write and draw in air. We envision the applicability of this use case in generating quick annotations to documents on-the-go or simplifying search through drawing letters while skimming through a contact list for names that begin with a specific letter. Again, this set of applications only requires the 3D TOF tracking module and hence is invariant

Figure 7-3: The popular game fruit ninja played using the Mime sensor's 3D tracking data. Notice the fast response time and precision that allows the user to slash two fruits that surfaced up at the same time.

to the user's surroundings, making it ideal for quick input without the need for external keypads. This type of input falls well within the category of **letter gestures**.

Immersive gaming: In addition to content consumption, portable devices are increasingly used for recreation. Bringing an immersive gaming experience closer to the user through HMDs has been explored in several contexts [86]. The Mime sensor provides gaming control without the need for additional controllers. We demonstrate fast and precise performance of the Mime sensor through a gaming application. Specifically, we use the Mime sensor to play the popular Fruit Ninja game (see Fig. 7-3). The application requires short reaction times both by the user and the sensing system to effectively advance through the game.

Interactive capture tools for photography: We now demonstrate how 3D tracking combined with ROI-based RGB processing provides fine gestural control. In this application we use the Mime sensor to allow manipulation of visual content as we capture it. While capturing pictures from a vantage point the user typically cannot manipulate it on the display screen itself because of the smaller degrees of freedom of control available and the inherently limited display size. Our application allows the user to manipulate visual content during the capture pipeline through gestures that map to the scene or region of interest being captured. The key concept is that the user is using gesture to interact with the image while the photo is being taken and the scene and desired view is fresh and alive. As shown in Fig. 7-4, the user makes an L-shaped gesture, which activates the camera and begins taking a picture. This is an example of a **visual function gesture**. Depth based zoom allows the user to control focus. After gesturing to the HMD to take the picture, the user crops the image and finalizes the selection using C-shape gestures which is an example of **letter gestures**. A key feature of smart glasses is the ability to capture visual information while we see it. Hence, input options that enable shortcuts to such applications will determine

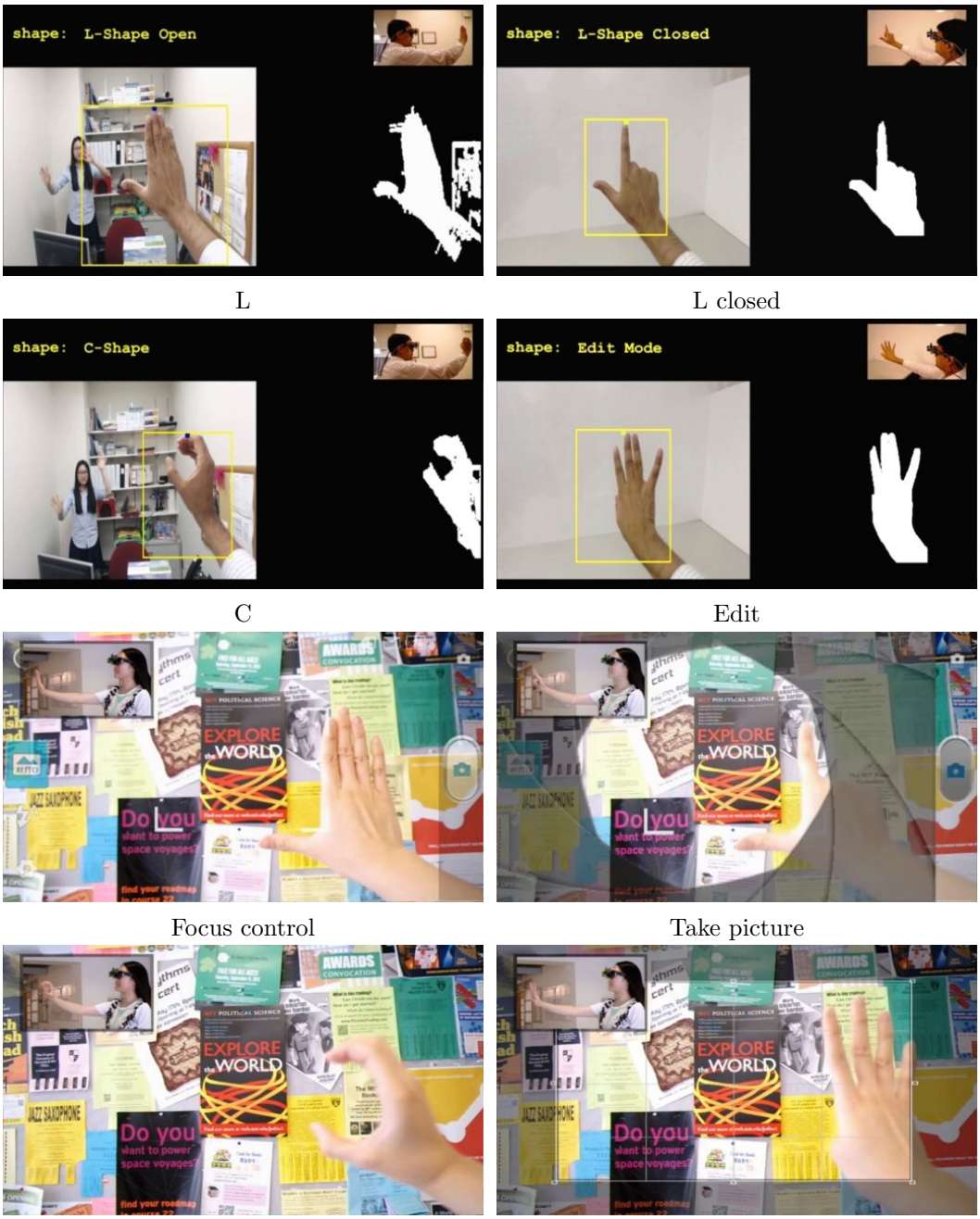


Figure 7-4: Interactive photography application with shape-based gestures implemented using Mime's 3D and RGB data fusion approach. All gestures work accurately even in cluttered environments with competing targets.

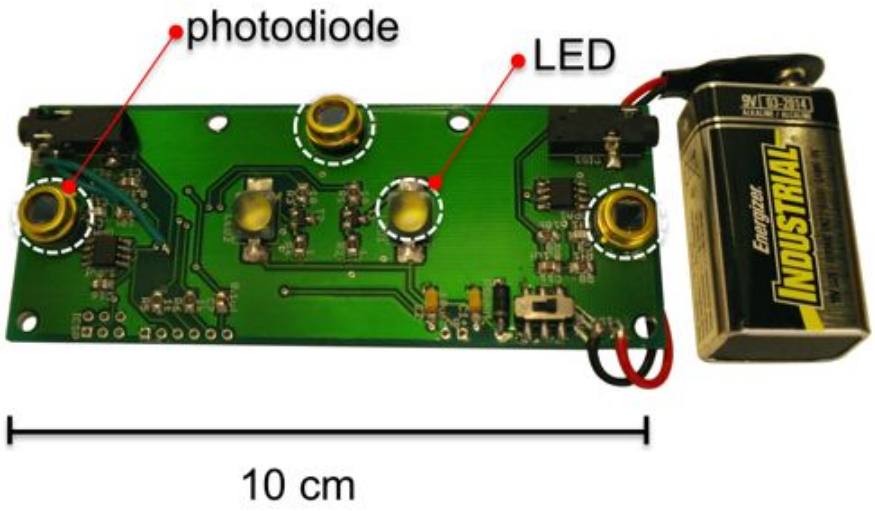


Figure 7-5: Iterated version of Mime sensor designed to map dimensions and form-factor of Google Glass and a smart phone.

the utility of such features. Both these use cases first use 3D TOF information to localize the hand target and then use RGB processing to identify finer gestures in the ROI.

7.3 Applications with the Google Glass

In this section we discuss system design and integration of the Mime sensor with the Google Glass. We introduce applications built with this framework and discuss user scenarios that these applications enable. The Glass device presents interesting user interface constraints that are different than the Vuzix smart glass we used in the previous section. Since the Google Glass has a small display in the upper-right region of the user's visual field, it is more ideal for augmented reality applications as opposed to full-display virtual reality systems. The form-factor of the device was an important consideration in the more rectangular design of our iterated sensor seen in Fig. 7-5. In the rest of this section we will discuss sensor location and visual feedback constraints that differ from virtual reality style applications presented in Section 7.2.

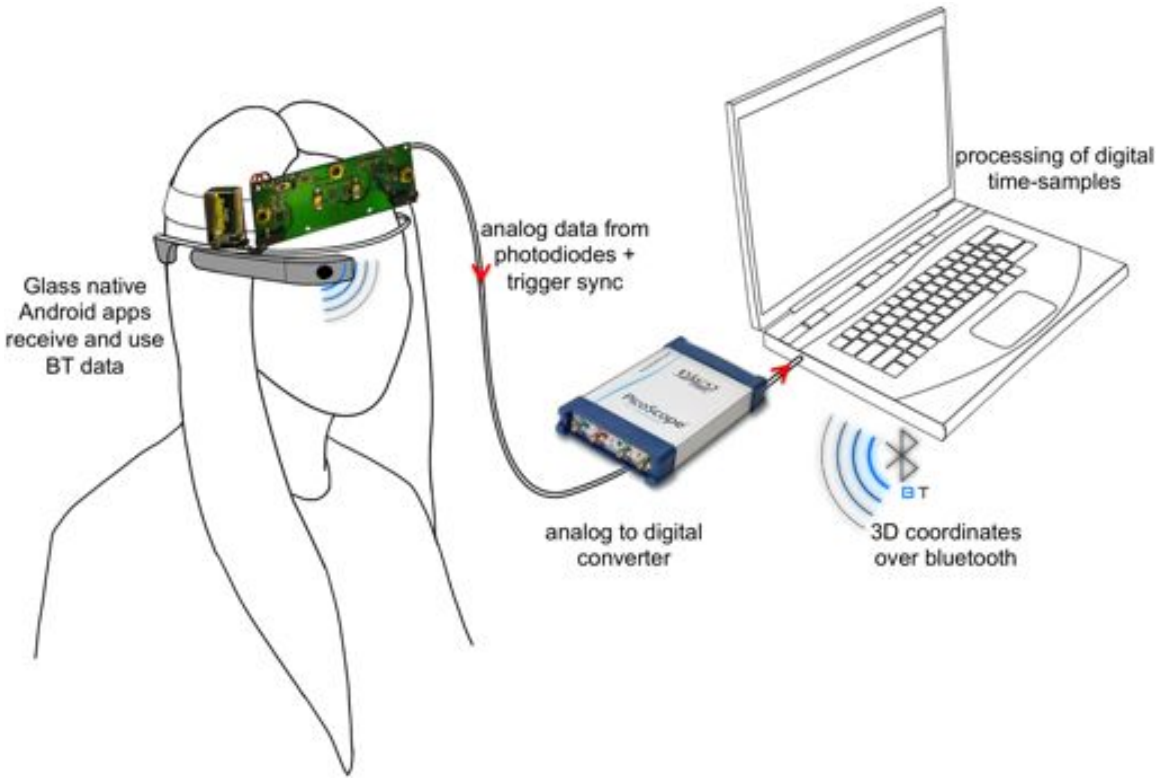


Figure 7-6: System design and data processing flow. Analog data from the sensor are received by a USB oscilloscope which transfers time samples to a laptop computer. The time samples are processed and 3D coordinates are communicated over Bluetooth to the Glass.

7.3.1 System design

For integration with the Google Glass form-factor, we redesigned the hardware behind the Mime sensor to better match the dimensions and mounting location on the Glass. Fig. 7-5 shows an iteration of the sensor that is ideal for both smart glasses and smart phones in shape and dimensions. System design is seen in Fig. 7-6. For the applications described in this section, the Mime sensor is mounted along the rim of the Glass. The analog data from the photodiodes and synchronization trigger are passed to a digital oscilloscope which digitizes the incoming analog data. Time samples from the oscilloscope are read over USB by a processing unit (for our purposes a laptop computer). Data processing is performed on the laptop to produce 3D coordinates of the tracked object. These coordinates are then communicated over Bluetooth to the wearable device. Google Glass in our implementation

runs native Android applications to process the incoming Bluetooth data and use it as required.

7.3.2 Live trace

Our initial exploration of the Google Glass platform as an emerging wearable display revealed some input limitations. Currently, speech and touch pad swipes/taps are the only way to input and navigate the interface. With limited touch pad area, text input is challenging. However, the vantage point of the camera is ideal since it maps to roughly where the user might be looking at. We exploit the position of the camera to harness it as the visual canvas which the user can point to, trace and annotate.

(a) Pulling up menu items. (b) Creating a trace. (c) Region of interest selected.

Figure 7-7: **Live Trace:** This scenario shows a user creating a trace on-the-go. The region of interest is cropped once the trace is made.

In this interactive experience we are interested in enabling quick input actions that do not induce fatigue and are not tedious or repetitive. The application allows the user to interactively select an object or region of interest in their live view using the **pointing gesture**. Millimeter accurate finger tracking is achieved using the 3-pixel Mime sensor. We register the field of view of the Mime sensor with the field of view of the optical camera on Glass. As the user traces an object or region, a trace appears from the point of origin and follows the user's hand. This trace is overlaid on the camera preview seen through the Glass display. The Live Trace app demonstrates the effectiveness of gestural control for head mounted displays. Existing touchpad input to Glass has a very small surface area that precludes such an interaction. The act of tracing provides the user with a quick tool to tag objects of interest while the scene view is fresh and alive. Once the trace is completed by the user, the application selects a rectangular region spanning the maximum horizontal

and vertical coordinates traced. The coordinates of the trace around the object are also stored as meta-data along with the cropped rectangular region to provide easy tagging after images are captured. Fig. 7-7 shows snapshots from this interaction.

Optical flow tracking: There are two challenges with visually overlaying the trace on the Glass display. First, the object may move while tracing or once the trace is complete. Second, the user's head may move during tracing or after tracing. Both these cases would result in a mismatch with the overlaid trace and the object's position in the field of view. To resolve the above two issues, we implemented optical flow tracking with the RGB camera on the Glass. Optical flow tracking is enabled with the RGB camera to maintain the position of the trace relative to user head motion. Once the user starts tracing we use the Lucas-Kanade method for flow tracking available in OpenCV [1]. The method first performs pyramidal feature tracking to identify and select the dominant features close to the trace. The height of pyramid determines maximum displacement that can be tracked. The iterative Lucas-Kanade method is used to solve the optical flow equation for all pixels in the neighborhood of the selected features by the least squares criterion. The output is a translation vector in x, y (see Fig. 7-8 a, b). If translation is detected, we correspondingly translate the trace to map to its new, correct position. The pyramidal approach cannot account for head movement in the third (z) dimension as seen in Fig. 7-8 c.

(a) Initial location of trace. (b) Trace translated. (c) Movement in z misaligns trace.

Figure 7-8: **Optical flow tracking:** These are consecutive images where the object moves relative to the user's head position and is tracked. Notice part (c) where movement in z cannot be rectified.

Burst mode For selecting multiple regions of interest, we present a burst mode. In the example we implemented (see Fig. 7-9), the interaction is reminiscent of creating digital cut-outs of real world magazines. The user traces either multiple objects in the same field

[1] OpenCV. www.opencv.org

of view or creates multiple traces in succession. The regions of interest from these traces are then aggregated as a set. This style of interaction could be a useful tool in scenarios involving sequences (of steps or objects) or in collecting a set of related items. Examples include creating instructions for cooking and building a real-world gift registry.

Figure 7-9: **Burst mode:** The user selects three objects of interest in succession. The traces around the object create a collection of related fashion items that can be added to the user's catalog.

Obtaining a region of interest could be useful in many other scenarios as well. Next, we describe a few such scenarios that we implemented with the Mime sensor. These use cases are designed to be used for a short duration and easily integrated with other modalities.

7.3.3 Live filters

We use the trace as the foundation for allowing the user to indicate interest in a visual region. In this section we introduce a tool to manipulate the traced region while it is being captured. While capturing pictures from a mobile or wearable device camera, the user typically cannot manipulate it on the display screen itself because of the smaller degrees of freedom of control available and the inherently limited display size. Our approach allows the user to manipulate visual content during the capture pipeline through pointing, tracing and selecting. First, the user traces out an object and the app then snaps to a rectangular hull enclosing the region of interest (ROI). The user is then presented with filtering options that can be applied to the ROI. We provide the following filter options – blur, mosaic, sharpen, color filter (sepia) and character recognition. The Mime sensor tracks the user's hand while they select a filtering option. In our implementation, selection is signaled by hovering over a filter until it is applied. Fig. 7-10 shows filtering options and their effects on the ROI.

(a) Color filter. (b) Blur. (c) Mosaic.

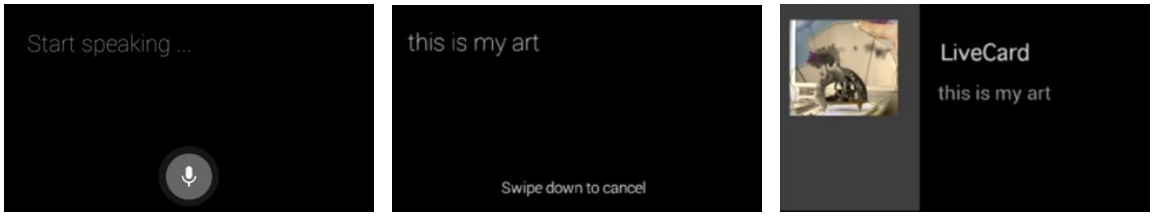
Figure 7-10: **Live filters:** Visual effects applied to the region of interest selected by tracing could be useful for obscuring or highlighting details.

Often filtering is desired for either highlighting or obscuring details from the user's visual field. Using the blur filter on the ROI, it is also possible to obscure the details of certain objects in the final recomposited image. This is potentially useful in preserving privacy while sharing images that are captured on-the-go. Likewise color filters may be applied for visual aesthetic effects. We discuss character recognition in greater detail in the next section. We envision customizable filtering options populated by the user based on their specific use cases or filtering options could also be listed by frequency of use. For example, when capturing live events or public spaces journalists may frequently desire to obscure faces in their images while architects may require highlighting and zooming in to details.

7.3.4 Text annotations

Even though manual text input to the Glass is challenging, speech input is readily available. We use this additional modality as an indirect way to provide annotations to traced out regions. Once a trace has been captured and cropped to fit the rectangular region around it, the user is prompted to provide a tag or annotation using speech (see Fig. 7-11). The region along with the annotation are then composited into a Livecard that is placed on the Glass timeline.

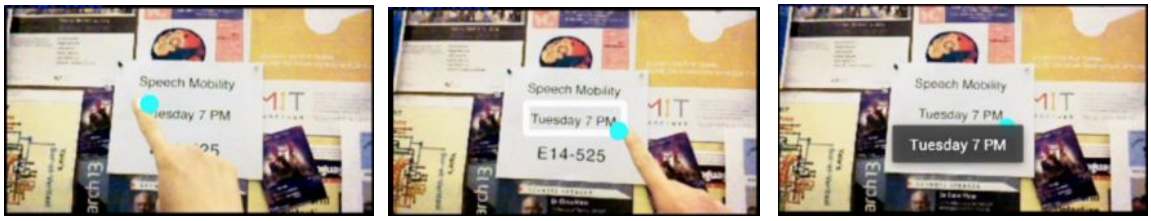
This pipeline suffers from failure cases associated with speech recognition engines. Input is limited to annotations that are accurately transcribed to text by the voice recognition engine. For our application, we used native Android speech recognition to process voice annotations.



(a) Speech input prompt. (b) Recognized speech-to-text. (c) Composited Livecard.

Figure 7-11: Text annotations through speech input on Glass.

Optical character recognition While experimenting with speech input to Glass, we observed that if we started with our initial premise of using the world itself as an input canvas, we could opportunistically use already existing text in our environment. In order to achieve this we integrated optical character recognition into our pipeline. Again, in environments with dense amounts of text, any character recognition engine would be too slow or too failure prone. We use the advantage of **finger pointing gesture** and tracing to select only those regions of text that are relevant. We experimented with a quicker input style. Since text typically occurs in lines we provide the user a rectangular region to start out with. Instead of tracing around the text region, the user simply starts at the upper-left corner of interest and moves his/her hand along the diagonal of the rectangle to pick the bottom-right corner. Once selected the pixels within the rectangular region are processed by our OCR engine to output the recognized text. The above selection and output steps are seen in Fig. 7-12 In our implementation we used the Tesseract library [87].



(a) Upper-left corner. (b) Diagonal selection. (c) Recognized text.

Figure 7-12: Screenshots of optical character recognition within rectangular selection implemented on Glass.

This selection and processing tool could naturally integrate with quick note-taking applications, such as Evernote[2] on emerging wearable platforms. Text extracted either from voice

[2]Evernote. www.evernote.com

recognition or character recognition could be parsed to then added to calendars and lists of reminders.

7.4 Back to the desktop

Here, we propose constructing a virtual desktop centered around the smartphone display with the surface around the display opportunistically used for input. The Mime sensor provides sensing opportunities around the display through hand gesture and motion sensing. The Mime sensor on the phone allows the table surface next to the phone to be mapped to conventional desktop windows, and the phone's display is a small viewport onto this desktop. Moving the hand is like moving the mouse, and as the user shifts into another part of the desktop, the phone viewport display moves with it. Instead of writing new applications to use smart surfaces, existing applications can be readily controlled with the hands.

Sensing hand motion on a surface next to a mobile device could be compared to conventional mouse input because of the similarity of hand pose and movement during input. The wrist is rested on a surface during input; relative movement on the surface results in cursor manipulation (as opposed to absolute position). This is an advantage compared with touch displays which only anchor a single finger to a surface during interaction. Further, touch displays provide input resolution that is determined by user finger dimensions. Thus, our design constraints for the style of interaction presented in this section are outlined below.

- Input on-surface should be precise within the target region of interest.
- The user should not be required to perform large hand movements, rather fine movements on the surface should facilitate desired cursor movement.
- Switching between whole hand and finger movement should be seamless for most effective input bandwidth.

Opportunistic trackpad: In Fig. 7-13 we see the user's hand movements being tracked by Mime and displayed like a mouse pointer on the smart phone screen. This makes an surface interactive without requiring to instrument the surface itself. The device display shows a circular pointer that maps hand movement to the screen.

Figure 7-13: Capturing hand movements on the side of a smart phone transforms the surface immediately next to the device into a larger input space.

For mobile applications that present high density of visual information, such as maps, keeping the display unoccluded is useful. It allows the user to focus on the content as it changes, i.e., while panning and zooming to different regions or details. We integrated the Mime sensor with a native map application on a smart phone to demonstrate the effectiveness of using the space immediately at the side of the device to interact, Fig. 7-14.

Figure 7-14: Mobile application scenario with Mime: User navigates within a map by moving their hand left to right.

Document annotation: As mobile devices support word processing use cases, the main input limitations are granularity of touch input and display occlusion. We investigate

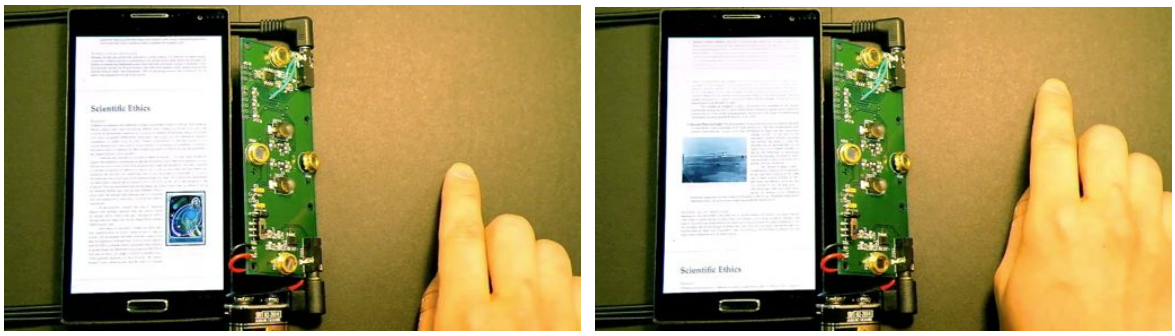


Figure 7-15: Scrolling a document by moving the hand vertically.

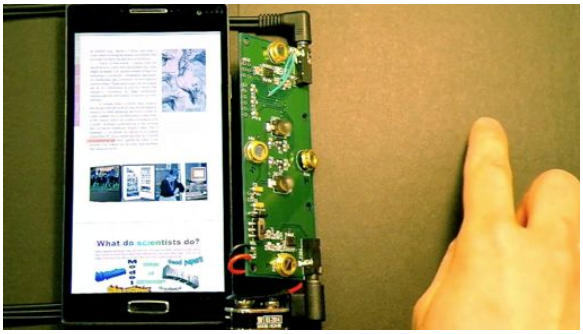


Figure 7-16: The side facing configuration of the Mime sensor is used to highlight text in a word processing task on a mobile device.

and present use cases in which trackpad-like input on the side of the device mitigates these challenges associated with conventional touch input. First, we mapped scrolling of documents to vertical movement of the hand as a pointer on the side of the device as shown in Fig. 7-15. Next, we experimented with precise pointing to lines of text and locations within the document (see Fig. 7-16). Note that on a touch display it is hard to obtain precise location within text due to occlusion and lack of very fine input granularity. We further experimented with two-handed input. As seen in Fig. 7-17 the user can select a function by touching it (using the left-hand in this example) and control the value of the specific function with hand movement to the side of the device. Since the screen is not occluded while controling the changing value (in these examples hue, size and opacity), the user can easily view the desired output while precisely controling how much it changes (see Figures 7-17, 7-18, 7-19). In the example we present such control could be useful in selecting an appropriate highlighting tool for annotating the document.

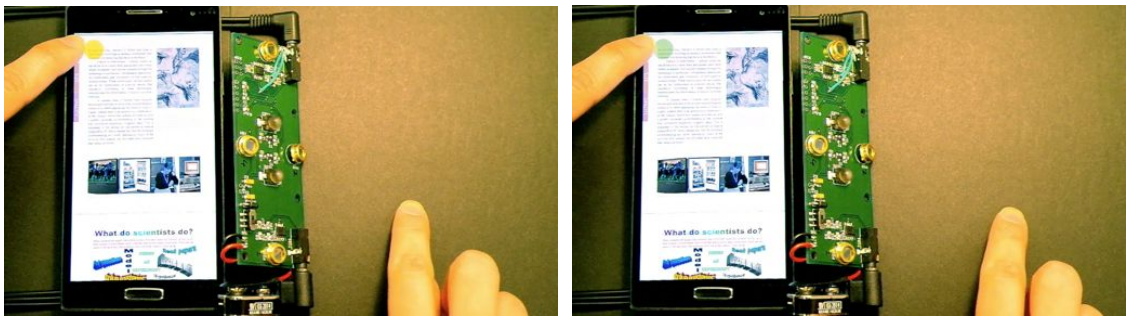


Figure 7-17: Two-handed input with touch used to select a function and side movement of the hand used to precisely control color.

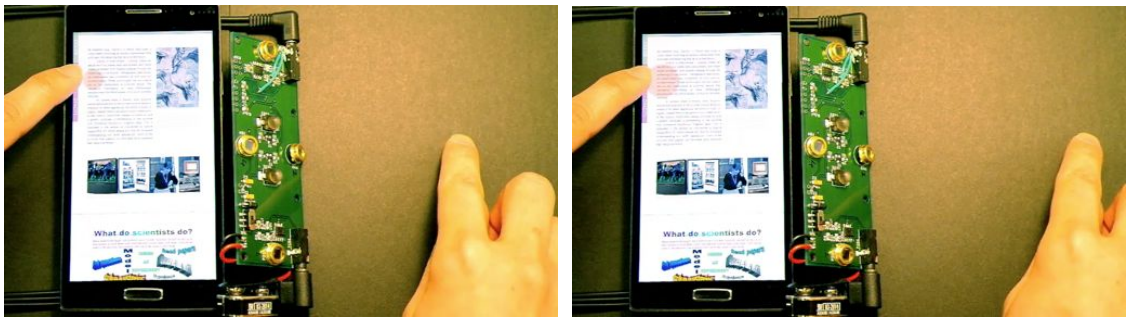


Figure 7-18: Two-handed hybrid input with touch used to select a **size** function and hand movement used to precisely control brush thickness.

Figure 7-19: Opacity decreased by moving the hand vertically downward.

This chapter presented applications of precise 3D sensing enabled by the Mime sensor. We first defined commonly occurring sets of gestures that would be useful in mobile scenarios. Despite input sensing limited to a single hand, we demonstrated that the universal finger pointing gesture provides adequate information to capture what the user is interested in while using wearable head mounted displays. We developed use cases for soliciting additional input through other modalities like speech and automated text recognition. Finally, we introduced the use of the Mime sensor in a smart phone configuration where hand movement is sensed on a surface on which the device is placed. This application shows the utility of precise mouse-like input to replace touch in use cases where the user is interested in the contents on the display while manipulating them (for example, navigating maps and word processing tasks).

Chapter 8

Conclusion

The use of computing systems is defined and limited by the set of input actions available to the user. Desktop systems limit the interaction space to the keyboard and mouse. The use of touch devices confines the user to the boundaries of the display itself. Emerging wearable displays like Google Glass attempt to dissolve the boundary between the display and our physical environment. While output to user is achieved through information overlaid on displays, designing responsive input to the system is essential for an effective on-the-go computing experience.

In this thesis, we investigated the limitations and advantages of mobile devices with small touch displays. We discussed input opportunities for emerging form-factors of wearables like smart glasses. We primarily focused on free-form hand gesture input to mobile and wearable devices. In order to enable sensing of unencumbered gestural input, we presented an analysis of existing techniques and their limitations. The main constraints when implementing hand gesture control on mobile devices are power, sensor size, performance in a variety of environment conditions, and computation requirements. We addressed these constraints individually with new image acquisition frameworks and experiments. Our approaches included a compressive depth acquisition camera using only a single time-resolved photodiode and a single photon counting detector based low power depth acquisition framework. Since these two techniques do not alleviate computation requirements, we presented

an application-specific signal processing framework, called parametric optical signal processing to identify features of interest – hands. This framework is intended to meet all mobile constraints together, but trades off generality of sensing to only acquire pre-defined types of features like moving hands. We built a real-time implementation of this framework, called the Mime sensor which is one such effort to provide a compact, low-power 3D gesture sensor which is amenable to mobile size and power constraints. Finally, we presented integration of this sensor with various applications designed for full-display virtual reality smart glasses, augmented reality displays like the Google Glass and traditional mobile devices like smart phones.

Here, we outline steps towards practical integration of the Mime sensor prototype with HMDs or other mobile devices.

- For a standalone implementation of the sensor, sampling, analog-to-digital conversion and processing needs to be integrated with a micro-controller unit. Due to the sensor's low power requirements, battery operation is trivially extended.
- Demonstrate multi-finger detection accuracy and tracking. This will require sophisticated computer vision processing within the ROI to mitigate the effect of extraneous objects that may lie within the ROI.
- Introduce multiple-hand tracking and expand the set of gestures supported. We present preliminary theoretical extensions of multiple hand tracking with the Mime TOF module in Chapter 6.
- Placement of the Mime sensor on the mobile or wearable device governs the set of supported gestures. The baseline requirement of the Mime sensor makes it ideal for placement along the HMD frame, facing the world. This configuration naturally elicits user interaction in front of their body, and may be useful in creating an immersive experience. In the case of a handheld mobile device, this configuration will be use case dependent. For example, the world-facing configuration is ideal for 3D augmented reality interaction while a lateral placement (side-facing) may be better suited to applications that replace stylus input with hand or finger-based input.

- **Close-to-body discreet interaction** using subtle actions is highly desirable. This configuration is possible via a side-facing or downward-looking Mime sensor. Mime's high precision tracking and localization makes it possible to accurately detect subtle gestural movements. Such Mime sensor configurations would be an ideal subject for future work.

Bibliography

[1] M. Weiser, "The computer for the 21st century," *Scientific American*, vol. 265, no. 3, pp. 94–104, 1991.

[2] Leap-Motion. (2012) 3d sensor for sub-millimeter accurate finger tracking. [Online]. Available: https://leapmotion.com/

[3] R. Lange and P. Seitz, "Solid-state time-of-flight range camera," *IEEE J. Quant. Electron.*, vol. 37, no. 3, pp. 390–397, 2001.

[4] PMD. (2012) Camboard nano depth camera. [Online]. Available: http://www.pmdtec.com/

[5] B. Jones, R. Sodhi, D. Forsyth, B. Bailey, and G. Maciocci, "Around device interaction for multiscale navigation," in *Proc. 14th Int. Conf. Human-Computer Interaction with Mobile Devices and Services*. ACM, 2012, pp. 83–92.

[6] T. Starner, J. Auxier, D. Ashbrook, and M. Gandy, "The gesture pendant: A self-illuminating, wearable, infrared computer vision system for home automation control and medical monitoring," in *Fourth IEEE Int. Symp. Wearable Comput.*, 2000, pp. 87–94.

[7] C. Harrison, H. Benko, and A. D. Wilson, "OmniTouch: Wearable multitouch interaction everywhere," in *Proc. 24th Ann. ACM Symp. User Interface Softw. Tech.*, 2011, pp. 441–450.

[8] G. Bailly, J. Müller, M. Rohs, D. Wigdor, and S. Kratz, "ShoeSense: A new perspective on gestural interaction and wearable applications," in *Proc. ACM Ann. Conf. Human Factors in Comput. Syst.*, 2012, pp. 1239–1248.

[9] P. Mistry, P. Maes, and L. Chang, "WUW – wear ur world: A wearable gestural interface," in *Ext. Abs. Human Factors in Comput. Syst.* ACM, 2009, pp. 4111–4116.

[10] R. Sodhi, H. Benko, and A. Wilson, "Lightguide: projected visualizations for hand movement guidance," in *Proc. ACM Ann. Conf. Human Factors in Comput. Syst.*, 2012, pp. 179–188.

[11] C. Harrison, D. Tan, and D. Morris, "Skinput: appropriating the body as an input surface," in *Proc. ACM Ann. Conf. Human Factors in Comput. Syst.*, 2010, pp. 453–462.

[12] D. Miaw and R. Raskar, "Second skin: Motion capture with actuated feedback for motor learning," in *IEEE Virtual Reality Conf.*, 2010, pp. 289–290.

[13] S. Gustafson, D. Bierwirth, and P. Baudisch, "Imaginary interfaces: Spatial interaction with empty hands and without visual feedback," in *Proc. 23nd Ann. ACM Symp. User Interface Softw. Tech.*, 2010, pp. 3–12.

[14] X. A. Chen, N. Marquardt, A. Tang, S. Boring, and S. Greenberg, "Extending a mobile device's interaction space through body-centric interaction," in *Proc. 14th Int. Conf. Human-Computer Interaction with Mobile Devices and Services.* ACM, 2012, pp. 151–160.

[15] L. G. Cowan and K. A. Li, "Shadowpuppets: supporting collocated interaction with mobile projector phones using hand shadows," in *Proc. ACM Ann. Conf. Human Factors in Comput. Syst.*, 2011, pp. 2707–2716.

[16] K. D. Willis, I. Poupyrev, S. E. Hudson, and M. Mahler, "Sidebyside: ad-hoc multi-user interaction with handheld projectors," in *Proc. 24th Ann. ACM Symp. User Interface Softw. Tech.*, 2011, pp. 431–440.

[17] A. Butler, S. Izadi, and S. Hodges, "Sidesight: multi-touch interaction around small devices," in *Proc. 21st Ann. ACM Symp. User Interface Softw. Tech.*, 2008, pp. 201–204.

[18] S. Kratz and M. Rohs, "Hoverflow: expanding the design space of around-device interaction," in *Proc. 11th Int. Conf. Human-Computer Interaction with Mobile Devices and Services*. ACM, 2009, p. 4.

[19] R. Bainbridge and J. A. Paradiso, "Wireless hand gesture capture through wearable passive tag sensing," in *IEEE Int. Conf. Body Sensor Networks*, 2011, pp. 200–204.

[20] D. Ashbrook, P. Baudisch, and S. White, "Nenya: subtle and eyes-free mobile input with a magnetically-tracked finger ring," in *Proc. ACM Ann. Conf. Human Factors in Comput. Syst.*, 2011, pp. 2043–2046.

[21] K.-Y. Chen, K. Lyons, S. White, and S. Patel, "utrack: 3d input using two magnetic sensors," in *Proc. 26th Ann. ACM Symp. User Interface Softw. Tech.*, 2013, pp. 237–244.

[22] D. Kim, O. Hilliges, S. Izadi, A. Butler, J. Chen, I. Oikonomidis, and P. Olivier, "Digits: Freehand 3D interactions anywhere using a wrist-worn gloveless sensor," in *Proc. 25th Ann. ACM Symp. User Interface Softw. Tech.*, 2012, pp. 167–176.

[23] D. Way and J. A. Paradiso, "A usability user study concerning free-hand microgesture and wrist-worn sensors," in *IEEE Int. Conf. Body Sensor Networks*, 2014.

[24] R. Smith, W. Piekarski, and G. Wigley, "Hand tracking for low powered mobile AR user interfaces," in *Proc. Sixth Australasian Conf. User Interface*, vol. 40. Australian Computer Society, Inc., 2005, pp. 7–16.

[25] B. H. Thomas and W. Piekarski, "Glove based user interaction techniques for augmented reality in an outdoor environment," *Virtual Reality*, vol. 6, no. 3, pp. 167–180, 2002.

[26] A. Kirmani, A. Colaço, F. N. C. Wong, and V. K. Goyal, "Exploiting sparsity in time-of-flight range acquisition using a single time-resolved sensor," *Opt. Expr.*, vol. 19, no. 22, pp. 21 485–21 507, Oct. 2011.

[27] A. Colaço, A. Kirmani, G. A. Howland, J. C. Howell, and V. K. Goyal, "Compressive depth map acquisition using a single photon counting detector: Parametric signal processing meets sparsity," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recog.*, 2012, pp. 96–102.

[28] A. Colaço, A. Kirmani, H. S. Yang, N.-W. Gong, C. Schmandt, and V. K. Goyal, "Mime: compact, low power 3d gesture sensing for interaction with head mounted displays," in *Proc. 26th Ann. ACM Symp. User Interface Softw. Tech.* ACM, 2013, pp. 227–236.

[29] J. Mei, A. Colaço, A. Kirmani, and V. K. Goyal, "Compact low-power 3d imaging of simple planar scenes using parametric signal processing," in *Proc. 47th Ann. Asilomar Conf. on Signals, Syst. & Computers.* IEEE, 2013.

[30] J. Sharpe, U. Ahlgren, P. Perry, B. Hill, A. Ross, J. Hecksher-Sørensen, R. Baldock, and D. Davidson, "Optical projection tomography as a tool for 3d microscopy and gene expression studies," *Science*, vol. 296, no. 5567, pp. 541–545, Apr. 2002.

[31] A. Wehr and U. Lohr, "Airborne laser scanning—an introduction and overview," *ISPRC J. Photogrammetry & Remote Sensing*, vol. 54, no. 2–3, pp. 68–82, Jul. 1999.

[32] D. A. Forsyth and J. Ponce, *Computer vision: a modern approach.* Prentice Hall Professional Tech. Ref., 2002.

[33] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, New York, NY, Jun. 2006, pp. 519–528.

[34] S. Hussmann, T. Ringbeck, and B. Hagebeuker, "A performance review of 3D TOF vision systems in comparison to stereo vision systems," in *Stereo Vision*, A. Bhatti, Ed. InTech, 2008, pp. 103–120.

[35] E. Stoykova, A. A. Alatan, P. Benzie, N. Grammalidis, S. Malassiotis, J. Ostermann, S. Piekh, V. Sainov, C. Theobalt, T. Thevar, and X. Zabulis, "3-D time-varying scene capture technologies—A survey," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 11, pp. 1568–1586, Nov. 2007.

[36] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int. J. Computer Vision*, vol. 47, no. 1–3, pp. 7–42, 2002.

[37] B. Schwarz, "LIDAR: Mapping the world in 3D," *Nature Photon.*, vol. 4, no. 7, pp. 429–430, Jul. 2010.

[38] S. B. Gokturk, H. Yalcin, and C. Bamji, "A time-of-flight depth sensor-system description, issues and solutions," in *IEEE Conf. Comput. Vis. Pattern Recog. Workshop*, 2004, p. 35.

[39] S. Foix, G. Alenyà, and C. Torras, "Lock-in time-of-flight (ToF) cameras: A survey," *IEEE Sensors J.*, vol. 11, no. 9, pp. 1917–1926, Sep. 2011.

[40] A. P. Cracknell and L. W. B. Hayes, *Introduction to Remote Sensing.* London, UK: Taylor & Francis, 1991.

[41] F. Blais, "Review of 20 years of range sensor development," *J. Electron. Imaging*, vol. 13, no. 1, pp. 231–240, Jan. 2004.

[42] A. Medina, F. Gayá, and F. del Pozo, "Compact laser radar and three-dimensional camera," *J. Opt. Soc. Amer. A.*, vol. 23, no. 4, pp. 800–805, Apr. 2006.

[43] J. Hightower and G. Borriello, "A survey and taxonomy of location systems for ubiquitous computing," *IEEE Computer*, vol. 34, no. 8, pp. 57–66, 2001.

[44] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing.* New York, NY: Springer, 2010.

[45] M. B. Wakin, J. N. Laska, M. F. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. F. Kelly, and R. G. Baraniuk, "An architecture for compressive imaging," in *Proc. IEEE Int. Conf. Image Process.*, Atlanta, GA, Oct. 2006, pp. 1273–1276.

[46] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. Kelly, and R. G. Baraniuk, "Single-pixel imaging via compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 83–91, Mar. 2008.

[47] G. A. Howland, P. B. Dixon, and J. C. Howell, "Photon-counting compressive sensing laser radar for 3d imaging," *Appl. Optics*, vol. 50, no. 31, pp. 5917–5920, Nov. 2011.

[48] P. L. Dragotti, M. Vetterli, and T. Blu, "Sampling moments and reconstructing signals of finite rate of innovation: Shannon meets Strang-Fix," *IEEE Trans. Signal Process.*, vol. 55, no. 5, pp. 1741–1757, May 2007.

[49] C. Beder, B. Bartczak, and R. Koch, "A comparison of PMD-cameras and stereo-vision for the task of surface reconstruction using patchlets," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2007.

[50] G. Sansoni, M. Trebeschi, and F. Docchio, "State-of-the-art and applications of 3d imaging sensors in industry, cultural heritage, medicine, and criminal investigation," *Sensors*, 2009.

[51] S. Mitra and T. Acharya, "Gesture recognition: A survey," *IEEE Trans. Syst., Man, Cybernetics, Part C: Appl. Rev.*, vol. 37, no. 3, pp. 311–324, 2007.

[52] S. B. Kang, J. A. Webb, L. C. Zitnick, and T. Kanade, "A multibaseline stereo system with active illumination and real-time image acquisition," in *Proc. 5th Int. Conf. Comput. Vis.*, 1995, pp. 88–93.

[53] J. García, Z. Zalevsky, P. García-Martínez, C. Ferreira, M. Teicher, and Y. Beiderman, "Three-dimensional mapping and range measurement by means of projected speckle patterns," *OSA Applied Optics*, 2008.

[54] D. Um, D. Ryu, and M. Kal, "Multiple intensity differentiation for 3-d surface reconstruction with mono-vision infrared proximity array sensor," *IEEE Sensors J.*, 2011.

[55] G. J. Iddan and G. Yahav, "3d imaging in the studio (and elsewhere)," in *Proc. SPIE*, vol. 4298, 2001, pp. 48–55.

[56] Y.-K. Ahn, Y.-C. Park, K.-S. Choi, W.-C. Park, H.-M. Seo, and K.-M. Jung, "3d spatial touch system based on time-of-flight camera," *WSEAS Trans. Inform. Sci. & Appl.*, vol. 6, no. 9, pp. 1433–1442, Sep. 2009.

[57] J. J. Leonard and H. F. Durrant-Whyte, *Directed sonar sensing for mobile robot navigation.* Kluwer Acad. Pub., 1992.

[58] G. Ogris, T. Stiefmeier, H. Junker, P. Lukowicz, and G. Troster, "Using ultrasonic hand tracking to augment motion analysis based recognition of manipulative gestures," in *IEEE Symp. Wearable Computers*, 2005.

[59] Ellipticlabs. (2012) Ultrasound gesture sensing. [Online]. Available: http://www.ellipticlabs.com/

[60] S. Gupta, D. Morris, S. Patel, and D. Tan, "Soundwave: using the doppler effect to sense gestures," in *Proc. ACM Ann. Conf. Human Factors in Comput. Syst.*, 2012, pp. 1911–1914.

[61] C. Harrison and S. E. Hudson, "Abracadabra: wireless, high-precision, and unpowered finger input for very small mobile devices," in *Proc. 22nd Ann. ACM Symp. User Interface Softw. Tech.*, 2009, pp. 121–124.

[62] H. Ketabdar, M. Roshandel, and K. A. Yüksel, "Towards using embedded magnetic field sensor for around mobile device 3d interaction," in *Proc. 12th Int. Conf. Human-Computer Interaction with Mobile Devices and Services.* ACM, 2010, pp. 153–156.

[63] J. Smith, T. White, C. Dodge, J. Paradiso, N. Gershenfeld, and D. Allport, "Electric field sensing for graphical interfaces," *IEEE Computer Graphics & Applications*, vol. 18, no. 3, pp. 54–60, 1998.

[64] R. Wimmer, P. Holleis, M. Kranz, and A. Schmidt, "Thracker-using capacitive sensing for gesture recognition," in *IEEE 26th Int. Conf. Dist. Computing Systems Workshops*, 2006, pp. 64–64.

[65] Q. Pu, S. Gupta, S. Gollakota, and S. Patel, "Whole-home gesture recognition using wireless signals," in *Proc. 19th Ann. Int. Conf. on Mobile Computing & Networking*. ACM, 2013, pp. 27–38.

[66] F. Adib and D. Katabi, "See through walls with wifi!" in *Proc. Ann. ACM Conf. on SIGCOMM*, 2013, pp. 75–86.

[67] M. Vetterli, P. Marziliano, and T. Blu, "Sampling signals with finite rate of innovation," *IEEE Trans. Signal Process.*, vol. 50, no. 6, pp. 1417–1428, 2002.

[68] T. Blu, P.-L. Dragotti, M. Vetterli, P. Marziliano, and L. Coulot, "Sparse sampling of signal innovations," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 31–40, Mar. 2008.

[69] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, 3rd ed. Upper Saddle River, NJ: Prentice-Hall, 2009.

[70] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 1.21," http://cvxr.com/cvx, Apr. 2011.

[71] G. C. M. R. de Prony, "Essai éxperimental et analytique: Sur les lois de la dilatabilité de fluides élastique et sur celles de la force expansive de la vapeur de l'alkool, à différentes températures," *J. de l'École Polytechnique*, vol. 1, no. 22, pp. 24–76, 1795.

[72] A. Colaço, A. Kirmani, G. A. Howland, J. C. Howell, and V. K. Goyal, "Compressive depth map acquisition using a single photon-counting detector: Parametric signal processing meets sparsity," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Providence, RI, Jun. 2012, pp. 96–102.

[73] S. A. Nelson, J. C. Lee, and M. A. Helgeson, "Handheld computer apparatus," US Patent 6,911,969, 2005.

[74] K. Lyons, T. Starner, D. Plaisted, J. Fusia, A. Lyons, A. Drew, and E. W. Looney, "Twiddler typing: One-handed chording text entry for mobile phones," in *SIGCHI Human Factors in Comput. Syst.* ACM, 2004, pp. 671–678.

[75] M. Billinghurst and H. Kato, "Collaborative mixed reality," in *Proc. Int. Symp. Mixed Reality*, 1999, pp. 261–284.

[76] C. Fredembach, N. Barbuscia, and S. Süsstrunk, "Combining visible and near-infrared images for realistic skin smoothing," in *Proc. IS&T/SID 17th Color Imaging Conf.*, 2009, pp. 242–247.

[77] K. C. Ho, X. Lu, and L. Kovavisaruch, "Source localization using TDOA and FDOA measurements in the presence of receiver location errors: Analysis and solution," *IEEE Trans. Signal Process.*, 2007.

[78] R. Szeliski, *Computer Vision: Algorithms and Applications*. Springer, 2010.

[79] A. Kirmani, H. Jeelani, V. Montazerhodjat, and V. K. Goyal, "Diffuse imaging: Creating optical images with unfocused time-resolved illumination and sensing," *IEEE Signal Process. Lett.*, vol. 19, no. 1, pp. 31–34, Jan. 2012.

[80] J. Mei, "Algorithms for 3d time-of-flight imaging," Master's thesis, Massachusetts Institute of Technology, 2013.

[81] J. Rekimoto and K. Nagao, "The world through the computer: Computer augmented interaction with real world environments," in *Proc. 8th Ann. ACM Symp. User Interface Softw. Tech.*, 1995, pp. 29–36.

[82] J. Rekimoto, "Gesturewrist and gesturepad: Unobtrusive wearable interaction devices," in *IEEE Symp. Wearable Computers*, 2001, pp. 21–27.

[83] G. D. Kessler, L. F. Hodges, and N. Walker, "Evaluation of the cyberglove as a whole-hand input device," *ACM Trans. Computer-Human Interaction*, vol. 2, no. 4, pp. 263–283, 1995.

[84] C. Goodwin, "The semiotic body in its environment," *Discourses of the body*, pp. 19–42, 2003.

[85] R. A. Bolt, "Put-that-there: Voice and gesture at the graphics interface," in *Proc. 7th Ann. Conf. Comp. Graphics & Interactive Techniques*. ACM, 1980.

[86] W. Piekarski and B. Thomas, "ARQuake: The outdoor augmented reality gaming system," *Commun. ACM*, vol. 45, no. 1, pp. 36–38, 2002.

[87] R. Smith, "An overview of the tesseract ocr engine," in *Proc. Ninth Int. Conf. Doc. Analysis and Recog.*, vol. 7. IEEE, 2007, pp. 629–633.