UNIVERSITY OF CALIFORNIA
SANTA CRUZ

Fast Time-of-Flight Phase Unwrapping and Scene Segmentation Using Data Driven Scene Priors

A dissertation submitted in partial satisfaction
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER ENGINEERING

by

Ryan Eugene Crabb

December 2015

The Dissertation of Ryan Crabb is approved:

_____
Professor Roberto Manduchi, chair

_____
Professor James E. Davis

_____
Professor Gabriel Hugh Elkaim

_____
Tyrus Miller
Vice Provost and Dean of Graduate Studies

Table of Contents

List of Figures

List of Symbols

Notation	Term	Description
$\|x\|$	*Absolute value*	Absolute value of x
B	*Active Brightness*	Average intensity of observed signal from illuminator only
ρ	*Albedo*	Reflectivity of surface, [0,1]
α	*Alpha value*	Continuous valued classification assignment to foreground
A	*Ambient Illumination*	Intensity from sources other than illuminator
ω_m	*Angular Frequency*	Radians per second of modulation signal
M	*Classification label*	Binary label to classify between foreground and background
$c(\tau)$	*Correlation Function*	Cross-correlation of correlation signal and reference signal with a phase delay τ
$\mathbf{\Sigma}$	*Covariance Matrix*	Covariance matrix of an array (bolded to distinguish from sum)
$\delta(x)$	*Dirac Delta Function*	Dirac delta function integrates to 1 if root is within limits of integration
D	*Distance*	Distance from sensor to surface
$E(x)$	*Energy Function*	Energy function of a term, $\log(p(x))$
exp(x)	*Exponential function*	Exponential function: e^x
$H(x)$	*Heaviside Step Function*	Function whose value is 1 for positive argument, 0 otherwise
$\boldsymbol{I}$	*Identity Matrix*	Identity matrix (bolded to distinguish from intensity image)
L	*Intensity constant*	Illumination at surface seen by pixel, calibrated at 1m for albedo 1 and reflection angle 0
I	*Intensity Image*	Intensity of light collected during one frame capture
$\|\|x\|\|_p$	$\mathcal{L}_p$ *Norm*	$\mathcal{L}_p$ norm of x
$C_p(K)$	*Labeling Cost*	Cost function for assigning label K to pixel p
$\mathcal{L}(y\|x)$	*Likelihood Function*	Likelihood of parameter values Y, given outcomes X
f_m	*Modulation Frequency*	Modulation frequency of signal
$\mathbb{N}_p$	*Neighborhood*	Set of pixels surround pixel p
$\mathcal{N}(\cdot;\mu,\sigma^2)$	*Normal Distribution*	Normal distribution with mean μ and standard deviation σ

ϕ	*Observed phase*	Phase of signal observed by sensor, equal to true phase modulo 2π
θ	*Phase*	Unwrapped phase, $2\pi \times$ number of periods of signal modulation
τ	*Phase Delay*	Phase delay of reference signal in cross-correlation
π	*Pi*	Ratio of a circle's circumference to diameter
C	*Pixel Color*	Color of pixel
$P(x)$	*Probability*	Probability of discrete outcome $X = x$
$p(x)$	*Probability Density*	Probability density function of continuous random variable X
β	*Reflection Angle*	Angle between surface normal and incident angle
c	*Speed of light*	2.998×10^8 m/s
σ_x	*Standard Deviation*	Standard deviation of random variable X
N	*Surface Normal*	Unit normal vector of a surface
t_{FG}	*Threshold Value*	Threshold for foreground classification
λ	*Weighting Value*	Weighting value for summing terms
K	*Wrap State*	Number of phase wraps

Abstract

Fast Time-of-Flight Phase Unwrapping and Scene Segmentation Using Data Driven Scene Priors

by

Ryan Crabb

This thesis regards the method of full field time-of-flight depth imaging by way of amplitude modulated continuous wave signals correlated with step-shifted reference waveforms using a specialized solid state CMOS sensor, referred to as photonic mixing device. The specific focus deals with the inherent issue of depth ambiguity due to a fundamental property of periodic signals: that they repeat, or wrap, after each period, and any signal shifted by a whole number of wavelengths is indistinguishable from the original. Recovering the full extent of the signal's path is known as phase unwrapping. The common, accepted solution requires the imaging of a series of two or more signals with differing modulation frequencies to resolve the ambiguity, the time delay of which will result in erroneous or invalid measurements for non-static elements of the scene. This work details a physical model of the observable illumination of the scene which provides priors for a novel probabilistic framework to recover the scene geometry by imaging only a single modulated signal. It is demonstrated that this process is able to provide more than adequate results in a majority of representative scenes, and that it can

be accomplished on typical computer hardware at a speed that allows for the range imaging to be utilized in real-time, interactive applications.

One such real-time application is presented: alpha-matting, or foreground segmentation, for background substitution of live video. This is a generalized version of the common technique of green-screening that is utilized, for example, by every local weather reporter. The presented method, however, requires no special background, and is able to perform on high resolution video from a lower resolution depth image.

To my teachers.

Acknowledgements

I am eternally grateful to Roberto Manduchi, for bringing me in out of the academic cold and helping guide my way along this path to where I now find myself. With encouragement all along, he let me explore my own ideas and methods with enough expert guidance to make sure I didn't run myself in circles, or off a cliff. Much thanks to James Davis for helping me maintain momentum and keep an eye on the big picture. And though we parted ways years ago, I would not have gotten started if Hai Tao hadn't seen my potential.

Thanks to my colleagues and co-authors, Feng Tang and Steve Scher, the team at Canesta, particularly the contributors to my background subtraction work, Collin Tracey, Akshaya Puranik, and Chris Rossbach with his mad coding skills. To all the folks in the UCSC Vision Lab, it was great to work with you and I hope our paths continue to cross. Thanks to Filix and Richard, for checking my math.

I could never have maintained such perseverance without the support, love and patience of my family—my mom, dad, and sister, whom have never shown an inch of doubt in me, my role model Granny Pauline "Mike" Crawford Crabb, and all the rest of you, from my greatest supporter, Vishali, to my littlest as well.

Introduction

Of all of the many capabilities born out of the human visual system (and might we dare generalize to the vertebrates as a whole), one of the most fundamental and powerfully advantageous is the ability for an individual to have an immediate and persistent perception of the 3 dimensional environment that one finds oneself in. To most, it is such an integral part to one's conception of the world that it is practically unimaginable what it would be to experience our spatial environment without vision. Certainly, most of the common tasks that are performed by the visual system are directly related to the perception of depth: acts of locomotion, manipulation of the environment, and even other vision tasks such as recognition of objects and scene understanding—which in the computer vision community may be regarded as traditionally 2D tasks—are greatly aided by intuitive segmentation that comes with perception of depth.

Given that the depth is a fundamental part of nature's vision system, it is almost surprising that 3D computer vision has been, to an extent, regarded by the community as its own set of problems, almost a sub-field distinct from the main body of computer vision. This sort of mindset is likely an accident of history, or more specifically the field of computer vision found its shape around problems that were technically feasible. The theory of Computational stereo was already well established 35 years ago [72], though the technological limitations on depth

measurement have long been a bottleneck in the development of the discipline. The determining of distance of an object through triangulation is indeed a straightforward process given that the object location is known from two viewpoints, but the difficulty in creating a full field range image through stereo is determining the matching correspondence for every pixel in the image. Much of the advancement around the stereo problem has been in making the correspondence solution more robust and efficient. Despite the ever-increasing processing power that has matched the predictions of Moore's law, modern machines still require a non-trivial processing time to compute depth (perhaps hindered by the ever increasing resolution of our digital cameras). Another traditional approach to range imaging is scanning LIDAR. This approach has also been traditionally slow, not from the computational load, but that it requires a single point (or in more recent versions a scanning line) to be scanned over the entire scene: a slow process with potentially delicate hardware.

Still, as vision-based systems start to become more ubiquitous, new applications emerge, and new demands for performance and capabilities are made. And as technological solutions prove to be ever more capable, new and bigger uses are dreamed up and the boundaries of what is considered possible are continually expanded. Automated manufacturing has a rich history at this point, and the tasks that are assigned to industrial robots become increasingly complex, involving processes that are less precisely repeatable. Robots are expected in

some cases to have spatial awareness in highly dynamic environments with high stakes for errors: just take self-driving cars for example.

Relatively recent developments, in the last decade and a half, of range imaging techniques of structured light and time-of-flight seemed to have sparked a resurgence in interest. Real-time (at up to 30 frames per second) full field (at VGA resolution) range imaging is affordable at the consumer level. It would seem that the pendulum has swung to the other side, in that the technical feasibility of high speed high resolution range imaging is no longer the bottle neck. There are likely a wealth of computer vision problems which could stand to be revisited in the context of three dimensions rather than two.

Still, the push goes on against the technological limitations. There is always a problem around the corner that could be more easily resolved with a more robust measurement, a faster frame rate or a longer range. In this work, I seek to improve on the current state of time-of-flight range systems. I present in this thesis a solution to the phase unwrapping problem, which is a fundamental issue to continuous wave based time-of-flight systems. A key feature of this solution is that it requires only a single frequency modulated signal, greatly reducing the capture time that is required by multiple frequency based approaches. This can result in more efficient power usage, higher frame rates, and most significantly, a reduction in erroneous measurements of moving objects.

Also presented is an application of this range imaging which demonstrates a need for the short capture time which is offered by the proposed phase

unwrapping technique. I describe an efficient method of alpha-matting (otherwise known as foreground segmentation) which can be performed on streaming color video using the aid of a much lower resolution depth stream—at a level which might require several seconds or longer given color information alone, or even prove to be unattainable for low color contrast foreground/background pairs.

In the following sections, I describe the structure of this thesis and provide a brief summary of each chapter.

1.1. Chapter 2: Background Knowledge

This chapter is meant to provide context in which my work can be understood to fit in. The motivations for this line of research are presented, namely the types of applications and possible uses for range imaging, as well as some of the existing shortcomings in the current technology—with the notion that this work may aid in overcoming some of these such issues.. After noting some of the most popular and relevant applications of range imaging, the subsequent section reviews the major techniques of range imaging, noting on what are the primary advantages and limitations of each of the methods, and thus what applications they might be best suited for.

Featured then is a discussion of different methods of time-of-flight, with a more thorough discussion than that of general range measurements. At a yet finer level of detail is a walkthrough of the principal and implementation of the specific

time-of-flight device that was used in the work described in subsequent chapters, the continuous wave amplitude modulated homodyne system. Following that is an enumeration and description of types of error and sources of measurement uncertainty in time-of-flight systems.

1.2. Chapter 3: Probabilistic Phase Unwrapping for Single-Frequency Time-of-Flight Range Cameras

This chapter describes my investigation into a novel method of phase unwrapping in time-of-flight cameras that use only a single frequency of modulated light. Phase unwrapping is a fundamental problem in indirect time-of-flight methods that rely on a modulated signal to indicate travel time. This solution offers an advantage over the most popular approach—in which the process of measuring phase shift is repeated two or more times while adjusting the modulation frequency for each instance in order to resolve phase ambiguity—in that the proposed method requires as few as 2 capture periods. Thus the total capture time is reduced, which opens the possibility of higher frame rate, lower energy, and less motion artifacts.

While the information provided by a second (or third) modulation signal is indeed a crucial component of the multi-frequency approach, the algorithm is able to make up for some of that lost information by incorporating the

accompanying intensity image into a probabilistic framework (Markov random field) that makes use of a physical model of reflectance and a simple but effective notion of local smoothness when indicated as a possibility by the wrapped phase values of adjacent pixels.

1.3. Chapter 4: Fast Single-Frequency Time-of-Flight Range Imaging

This work is a continuation of the previous chapter, though it offers fundamental improvements that indicate the viability of the proposed method. It makes use of the wrapped phase values to estimate the orientation of surfaces in the scene, and uses this estimate to both bolster the reflectance model, and to enhance the effectiveness of spatial consistency by enriching the similarity metric of neighboring pixels.

The viability of this method as a real-time solution to phase unwrapping is demonstrated by overhauling the energy optimization scheme to rely on a very efficient non-iterating technique called Non-Local Cost Aggregation [119]. Not only does it improve the computational efficiency by well over 2 orders of magnitude, the overall accuracy of the system is greatly improved. Due to the improved illumination model, similarity measurement, and spatial coherence cum optimization method, not only is the speed improved, but the number of incorrectly unwrapped pixels is cut by more than half.

1.4. Chapter 5: Foreground Segmentation Using Range Cues

Contained within this chapter is a novel method for alpha-matting of live video (often called chroma-key or green-screening due to the most popular approach of using a bright solid color screen behind the foreground, which can then be easily substituted digitally—as long as the distinct green color is not present in the foreground). The presented method requires no special background, instead using a time-of-flight range imager that is actually much lower resolution than the RGB video stream of interest. Using depth and color cues at their native resolutions, I apply a bilateral filter framework to determine the alpha-matted values at the higher resolution. Given a dividing depth plane to distinguish foreground from background, the algorithm efficiently generates a trimap to limit the more intensive processing to a very small portion of the field, while the comparably course depth values provide enough cues to enable reliable and realistic alpha matting at the native video resolution. This work was presented at the CVPR 2008 Workshop on Time-of-Flight [24].

Background Knowledge

2.1. Methods of Range Imaging

There are a variety of techniques for measuring depth within a scene or of an object, which are each individually specialized for particular attributes such as accuracy, density and resolution, frame rate, overall range, precision. This section provides a brief survey of the most common and popular techniques used in range imaging. The principles behind each technique are explained, along with some commentary on the strengths or weaknesses that might make some approaches more appropriate for particular tasks.

2.1.1. Geometric Range Measurements

Certainly the oldest form of range measurement is geometric-based: the depth that can be inferred through stereoscopic vision. Commonly referred to as triangulation, undoubtedly due to its basis in the geometry of the triangle, the principle behind this technique is demonstrated in Figure 2.1. This strategy for range imaging is a far more matured field than the time-of-flight technique which is at the basis of this thesis, and the body of literature is vast, so this section will only touch on some of the major points. Some of the surveys on stereo techniques alone include [5] [14] [31].

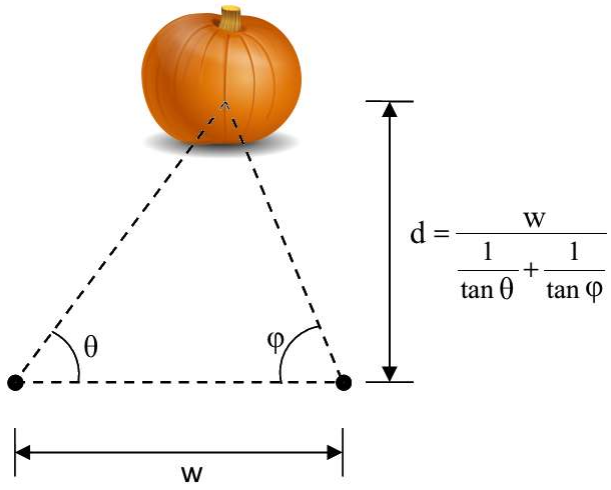


Figure 2.1. Geometry of distance through triangulation. Distance is determined by the direction (angle) of the surface from two viewpoints of a known width apart.

There are a very wide range of schemes that recover depth using geometric reasoning, some utilizing very precise, specialized equipment and others relying only on off-the-shelf cameras. Most all of these techniques involve some form of the following three types: stereo-vision, that is based on a device with a pair of two cameras, carefully calibrated with respect to each other, structure-from-motion, which relies on multiple views of the same scene captured from a single camera, or the structured light approach, in which a known pattern is projected from one viewpoint and this pattern is observed from a different viewpoint (imagine that one of the cameras from a stereo pair is replaced with an illumination source). While stereo and structure-from-motion are both cases of multiple-view-geometry, they are distinct enough from each other, each with their own body of specific techniques, that they warrant separate discussion.

1. Stereo

The basic premise of stereoscopy relies on a pair of cameras that are carefully arranged with respect to each other and can capture a pair of images of the scene simultaneously from their respective viewpoints. The features in one image are paired with the corresponding features in the other image, and the disparity between the positions of the features is an indication of their distance from the cameras. The cameras' intrinsic parameters must be known so that the geometry of the scene can be identified by each camera, up to a homography, but the pairing of different views can resolve the ambiguity. In typical stereo camera rigs, the cameras are carefully placed so that the sensor's image planes coincide and their horizontal axes are aligned. This greatly simplifies the geometry and also limits the task of pixel correspondence to be constrained to a single pixel row at a time.

In this simplified case, the chief goal then becomes to determine the corresponding pixels along each row of the left and right image pairs, still a major task [97] [40]. Much work has been devoted to finding optimal and efficient means of determining this correspondence, as the surveys reveal. However, some consistent issues that will be faced are that textured surfaces are necessary to perform this matching, so this approach is generally ill-suited to handle scenes with low-surface texture. Some approaches that try to globally optimize while maintaining local consistency, such as by use of a Markov random field, have found some success in dealing with low texture.

2. Multi-Image & Structure-from-Motion

There are several other similar methods which share the fundamental principles of this passive stereo approach. One such is called shape-from-motion and it involves using a set of frame stills from a video sequence or otherwise captures 2D images of a scene from multiple viewpoints, perhaps best encapsulated by Hartley and Zisserman [52]. Instead of starting with the assumption of two viewpoints with a known relation, a single camera is used and the change in viewpoints comes from manipulating the camera position. There exist a wealth of variations on this approach as well [63]. One distinction between some of the families of techniques here are that in some cases the motion of the camera is estimated by use of special hardware. In other cases, the motion is determined jointly with the scene geometry [120]. A common use case for this approach is to extract depth information from existing 2D video imagery for the purposes of rendering a stereoscopic view [62] [73]. While a more ambitious goal is to more fully recreate the 3D scene geometry [38].

3. Structured Light

Yet another range imaging approach that relies on triangulation is structured light. There is much similarity to the stereo approach here in that the two "viewpoints" are of a known relation to each other, however only a single camera is used, and instead of a second camera a light source is used. This light source, unlike in time-of-flight, does not illuminate the entire field of view

uniformly, but instead displays a very specific pattern. As an example, in the Kinect by PrimeSense [121] [106], a structured pattern is used that consists of simple points in a particular arrangement, demonstrated in Figure 2.2.

The task then becomes to identify the each of the points uniquely. Given that the pattern is known a priori, this problem ends up being much simpler than that of the correspondence problem in classical stereo. Further, it is no longer required that the surface be textured. This technique still has the same problems of shadowing, such that if a feature point is hidden from view of the camera, there will remain some ambiguity as to which point it is. Different decisions of which

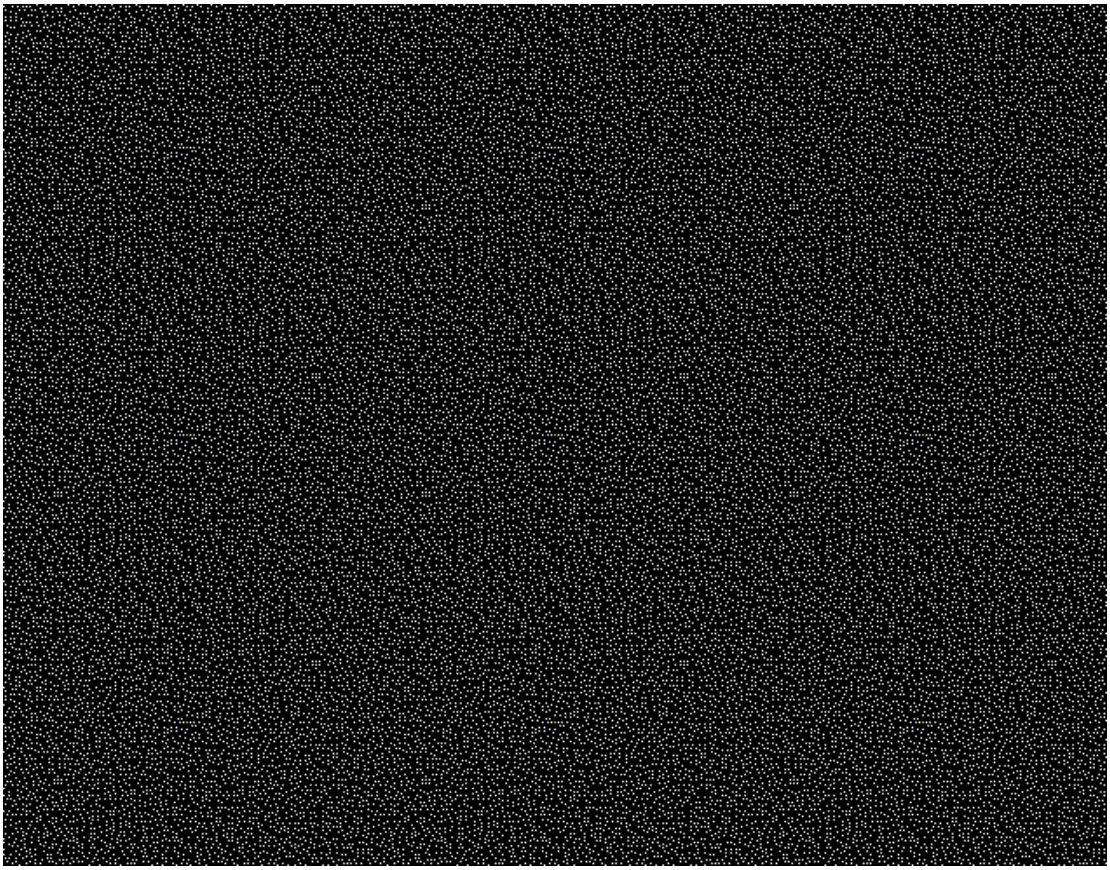


Figure 2.2. Structured light pattern from Kinect. Reprinted from [92].

feature points are hidden will result in different estimations of depth, and several may be valid. Further constraints (such as smoothness priors) may still be needed to determine the optimal solution.

2.1.2. Time-of-Flight

Time of flight refers to the time it takes for a signal to be transmitted, reflected (backscattered) from the target surface, and returned. In the context of sonar, that signal would be acoustical. In this context, however, the signal is electromagnetic radiation: light. As the speed of light in a vacuum is 299,792,458 meters per second, the time of this trip is extremely short—on the order of 10 nanoseconds for common distances—it requires very fast and precisely timed electronics.

The time of the signal may be measured directly (à la a stop watch) or indirectly, where some other property of the signal measurement indicates the time of flight. While the specific mechanisms of direct measurement devices may vary, the underlying concept is basically the same. There are a few flavors of approaches for the indirect method, however, which rely on their own unique tricks.

1. Pulse time-of-flight

This is the most straightforward (in concept) approach to time-of-flight. It is probably the method that might first assumed when the name “time-of-flight”

is explained. In this technique, a single pulse of light is flashed over the entire scene, and each pixel mechanisms marks the time of the return signal.

The most challenging part about this approach (with current technology) is the precise timing mechanism needed for each pixel. In [2] a small 6x6 array was constructed using a delicate bundle of optical fibers to connect the pixel plane to the electronic sensors. While the technique can provide reasonable results, much development is needed on the hardware side. Another drawback is in determining exactly when to mark the time of the return pulse. It is not instantaneous, rather the pulse will appear Gaussian in nature, so if the mechanism is set to trigger at a particular threshold, this value may appear at different parts of the curve based on the intensity of reflected light.

2. Shuttered Light Pulse

A variation of the pulse is the shuttered pulse. In this case the pulsed light signal is not intended as a delta, but rather a short square wave. The optical shutter is timed to close at a particular moment so that different amounts of light will have been accumulated on the shutter, based on how far that light had to travel. Of course, the absolute magnitude of this signal will also be determined by factors such as the albedo and orientation of the surface, so reference pulses need to be sent and received as well.

3. Continuous Wave Homodyne

The principle of continuous wave homodyne measurements is that a periodic signal is transmitted into the scene such that the signal received at the sensor at any given moment corresponds to the phase sometime in the past (specifically, the time it took for the signal to travel). By correlating this signal with a reference signal that is synced to the illumination signal (with some known offset), the phase shift of the received signal (up to the distance of one wavelength).

This is the method employed by the sensor used in my work. This is explained in more detail in section 2.2.

4. Continuous Wave Heterodyne

This approach is similar in nature to the Homodyne system, however in this case the demodulating reference signal is slightly out of sync with the transmitted signal. This creates a beat signal, of a much lower frequency, and is more robustly detected with standard hardware, and can achieve remarkable levels of precision in the sub-millimeter range [33]. This system is described in more detail in [75].

2.1.3. Interferometry

This is another active technique that requires transmitting a signal and observing the reflected signal. In this case, the signal is a monochromatic coherent light source (a laser) which is beam split into a measurement signal and a reference signal. The reference signal is directed towards a mirror with

adjustable, known distance and the measurement signal is directed towards the surface being measured. The two beams are then superimposed and the constructive or destructive interference of the phase shift signal will be observable [32].

By adjusting the path length of the reference signal and observing the effects on the interference, the distance of the measurement signal can be determine to within fractions of the laser's wavelength, on the order of nanometers. On the flip side to this fantastic precision is a terrifically small unambiguous range, which is half the wavelength, still only hundreds of nanometers. One solution with some attention, reminiscent of the multiple frequency solution to time-of-flight, is to use multiple wavelengths of light to extend this unambiguous range [16] [27].

2.2. Phase-Stepped Amplitude Modulated Continuous Wave Homodyne ToF

This is the system for which the phase unwrapping problem is investigated in the following chapters, so a more complete description of this system is described here. Included is an explanation of the principle behind this method of measurement, the process illustrated by a series of tractable equations that model the physical system and explain how a depth measurement is extracted from a series of intensity images of a systematically illuminated scene. I go on to

enumerate some of the shortcomings and difficulties presently encountered by these types of devices.

2.2.1. Amplitude modulated signal phase shift principle

The entire scene, that is the field of view of the camera, is illuminated by either an array of light emitting diodes or laser diodes. The illumination does not need to precisely uniform, as the intensity level does not impose a (significant) bias on the measurement, but in order to make best use of the full dynamic range of the pixel array, the field of view should be as evenly illuminated as is easily feasible.

The illuminator is cycled on and off at a frequency of f_m in order to produce a signal that approximates a sine wave (systematic errors will occur as a result of the signal not being a precise sine wave and are discussed in the Harmonics section of 2.2.3). The transmitted signal is modeled as an ideal sinusoidal function

$$g(t) = B_{\mathrm{L}}[1 + \sin(\omega_m t)] \tag{2-1}$$

where $\omega_m = 2\pi f_m$ and B_{L} is the intensity of the illumination source. The returned signal can be modeled as

$$s(t) = A_0 + B_0 \sin(\omega_m t - \theta) \tag{2-2}$$

with the first term corresponding to the DC component of the measured signal: $A_0 = B_0 + A_{amb}$, where A_{amb} is from ambient light sources—such as indoor lamps or the sun—and B_0 is amplitude of the signal reflected from the surfaces of the scene. The second term is the time-shifted and attenuated signal. The strength

of the signal will be diminished from B_L to B_0 in accordance with the travel distance, surface albedo, surface orientation,—a model of which is described in section 3.2.2— and θ is an unknown phase shift that corresponds to the distance the light has traveled. The distance d, ultimately the value that is intended to be measured, is directly related to this phase shift θ as

$$\theta = \frac{2d \cdot f_m \cdot 2\pi}{c} \tag{2-3}$$

Where c is the speed of light, approximately $3 \times 10^8 m/s$. Note that when the signal has traveled a distance of $c/2f_m$, the phase will have shifted an entire 2π. The phase of the signal having traveled any further will be indistinguishable from a signal that had traveled less by any integer multiple of the wavelength. Half the length of this distance is referred to as the unambiguous range (half because it travels both directions). In order to measure any further than the unambiguous range, the extended distance may be recovered in a process call phase unwrapping. In my work, I propose some novel solutions to this problem.

The value of θ is determined through a process of demodulating the received signal, termed the correlation signal, with a reference signal, which is synchronized with the illuminator. This reference signal can be delayed by a controllable phase shift τ, and the hardware system, as described in section 2.2.2, will effectively sample the cross-correlation function of the reference and correlation signals at different angular offsets τ. The reference signal is defined as

$$\gamma(t) = \mathrm{H}(\sin(\omega_m t)) \tag{2-4}$$

where H is the Heaviside step function, such that the value is 1 when the operand is positive and 0 otherwise. For the demodulation, the cross-correlation function of the signals with a delay τ over a period of T is defined as

$$c(\tau) = \frac{1}{T} \int_{-T/2}^{T/2} s(t)\gamma(t + \frac{\tau}{\omega_m})dt \tag{2-5}$$

where τ is the imposed delay from the transmitted signal and T is the time of one phase period, such that $T = 1/f_m$. The waveforms are correlated over a series of n periods. This function can be simplified as described by Martin Schmidt [99]

$$\begin{aligned} c(\tau) &= \frac{1}{nT} \int_{-\frac{nT}{2}}^{\frac{nT}{2}} (A_0 + B_0 \sin(\omega_m t - \theta))\, \mathrm{H}(\sin(\omega_m t + \tau))dt \\ &= \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} (A_0 + B_0 \sin(\omega_m t - \theta - \tau))\, \mathrm{H}(\sin(\omega_m t))dt \\ &= \frac{1}{T} \int_{0}^{\frac{T}{2}} (A_0 + B_0 \sin(\omega_m t - \theta - \tau))dt \\ &= A + B\cos(\theta - \tau) \end{aligned} \tag{2-6}$$

with definitions $A = A_0/2$ and $B = B_0/\pi$. As seen here, there are 3 unknowns: the phase shift θ, the intensity of the reflected signal B, and the ambient illumination A (shifting the phase delay τ is key in recovering the other variables). It may be noted that the number of periods n does not contribute to the final term, but it is

included to better reflect the actual process, in which the modulated signal is transmitted continuously for some time and signals are correlated over many periods. An optimal solution is found if the phase delays are equally spaced to have N samples over 2π radians [115] [87] [42]. In this case, Monson et al. [79] then show that the unknown variables can be calculated from the DC component of the discrete Fourier transform of the function c(τ). Given that the only the DC component is needed, the entire Fourier spectrum need not be calculated, and the unknowns can be computed efficiently as:

$$A = \frac{1}{N}\sum_{1}^{N} c(\tau_i) \tag{2-7}$$

$$B = \frac{1}{2}\sqrt{\left(\sum_{1}^{N} c(\tau_i)\cos(\tau_i)\right)^2 + \left(\sum_{1}^{N} c(\tau_i)\sin(\tau_i)\right)^2} \tag{2-8}$$

$$\theta = \tan^{-1}\left(\frac{\sum_{1}^{N} c(\tau_i)\cos(\tau_i)}{\sum_{1}^{N} c(\tau_i)\sin(\tau_i)}\right) \tag{2-9}$$

For the particular example described in the following section, phase shifts used for τ are spaced π/2 radians apart, requiring four samples to be taken.

2.2.2. Photonic Mixer Device

Phase measurement of a modulating signal on the order of 10s of megahertz can be accomplished through the use of specially engineered pixels

referred to as photonic mixer device, or PMD (note that while there does exist a company called PMD Technologies which manufactures time-of-flight cameras; to avoid confusion I will refer to these pixels as *multi-tap*, a more broad term that includes photonic mixing devices). These chips rely on standard image sensor techniques, most often complimentary metal-oxide-semiconductor (CMOS) but sometimes charged-coupled devices (CCD), to convert the incoming light into voltage. The trick is in how the charge is accumulated in each pixel.

Some significant reasons for the success and popularity of this method of time-of-flight range imaging is due to the design of the chip itself. Because these CMOS chips can be manufactured using industry-common methods, they are far cheaper to produce than competing methods such as the shuttered light pulse and range gating segmentation. Although they are not strictly "off the shelf" technology, they can be easily manufactured to order with existing factory equipment by semiconductor fabricators such as TSMC (Taiwan Semiconductor Manufacturing Company) [17]. Further, this solid state design is mechanically stable and relatively durable, as opposed to the delicate construction of pulse-based devices such as by Ailisto et al. [2]

Consider that the purpose of a pixel is to convert an analog sample of light into a digital quantization. That is, during the "shutter" period when the sensor is on, the pixel bin accumulates value as more light hits the sensor (shutter is in quotations as it makes reference to the physical device of analog film cameras in which a mechanic shutter expose the film briefly—here there is no physical

shutter, but rather the sensor is only activated for a limited time). Due to the inner photoelectric effect, the incoming photos generate a charge, which during a read-out cycle is converted into a voltage which is amplified and digitally quantized. Essentially the signal is integrated over the time period so the output value is proportional to the number of photons that hit the photodiode.

For the photonic mixer device, each pixel has two such accumulating bins, which can be alternately activated with great precision. The reference signal for this activation is synced with the light source, such that for half the time period of a modulation phase one bin is activated and accumulating, while for the other half of the period, the other bin is. For each of these bins, the accumulation for one half of the phase period is actually a reasonable approximation for the cross correlation of the transmitted signal with the Heaviside step function of the received signal [98], as described in equation (2-5) of section 2.2.1. And note that if one bin is representing the value for $c(\tau)$ then the other bin is the function offset by half a period: $c(\tau+\pi)$.

The hardware is thus well suited to take an even number of cross correlation samples. So for example, in order to sample $c(\tau)$ for 4 evenly spaced intervals within 2π, only 2 separate sampling intervals need to be performed: for $(0,\pi)$, and $(\pi/2, 3\pi/2)$. Thus, equations 7-9 are realized as:

$$A = \frac{1}{4}\sum_{1}^{4} c(\tau_i) \qquad (2\text{-}10)$$

$$B = \frac{1}{2}\sqrt{(c(\tau_1) - c(\tau_3))^2 + (c(\tau_2) - c(\tau_4))^2} \tag{2-11}$$

$$\theta = \tan^{-1}\left(\frac{c(\tau_1) - c(\tau_3)}{c(\tau_2) - c(\tau_4)}\right) \tag{2-12}$$

2.2.3. Noise, Artifacts, and Other Problems

At its most basic level, the primary purpose a time-of-flight imaging device is that it is a measuring instrument. And of fundamental importance to the science of measurement is the recognition and understanding of uncertainties. By understanding the nature and cause of errors, they might be mitigated, and even when they cannot be corrected, if they are at least known then the data can be properly interpreted. Our purpose as researchers and engineers of technology is not to tout and enjoy the remarkable capabilities of our tools, but to delve into their limitations in an effort to surpass them. This section provides a brief description of the major sources of uncertainty and error for time-of-flight range imaging.

Though I have attempted to provide as comprehensive (while concise) a catalogue as possible, certainly some issues are underrepresented, others overrepresented, and still others perhaps completely omitted. For example, several of the errors described below are variations on the same principle of "mixed pixels." Still, they result from different root causes and typically present

themselves distinctly. The fundamental basis of the error is that the signal received in one pixel comes from more than one reflected surface, whether because that pixel spans more than one object along their edge, an object is in motion, there is scattering inside the camera or from the lens, or that surfaces in the scene are transparent or semi-specular.

1. Intrinsic Random Noise

Certain processes in the camera electronics are bound to introduce noise at some level [74] .The photodiodes introduce electronic shot noise. As the photon detection is itself a random process following a Poisson distribution, the standard deviation of the photon noise is proportional to the square root of the number of photons [67]. The amplifier will introduce errors as well. And the process of digitization will inherently cause quantization errors.

2. Harmonics

One major source of error that is *not* caused by mixed pixels is an effect due to the imprecise modeling of the signals. Rapp calls this effect the wiggling error [91] due to the fact that a plot of the error against distance has a clear look of a sinusoid. It is also referred to as the harmonics error, as it results directly from the aliasing of higher order harmonics that are not correctly modeled [113]. As described in the previous sections, the transmitted signal is modeled as a sinusoid, and the reference signal is modeled as a square wave. Due to the

imprecision of the illuminator and photoreceptors (recall they are operating at speeds of 10s of nanoseconds) the waveforms will not follow the ideal models.

3. Temperature Drift

As the sensor warms up with use, it causes a bias in the phase measurement. This is, however, a reasonably consistent bias and can be corrected (more or less) by an initial temperature calibration of the hardware, then monitoring the temperature of the sensor during use [98].

4. Intensity Bias

In principle, the absolute intensity of the pixel should not influence the distance calculated from the measurements, as defined by the theory and equations in the previous section. However, this tends not to be the case in practice. As described by Falie and Buzuloiu [39], the observed intensity of a pixel is correlated with a predictable distance error that is also a function of the true distance. This effect is quite apparent in a straightforward example such as a checkerboard pattern on a flat surface.

5. Flying Pixels

A particular kind of error in phase-based time-of-flight sensors occurs for pixels that lie on a surface discontinuity, that is, along the edge of an object in the image foreground. Edge pixels produce an error that is generally more severe and far less predictable than the simple blurring of a regular 2D image. Recall that the intensity measured during each interval represents a sample of a phase shift

sinusoidal function. So mixing the values from the foreground and background portions of the pixel (some unknown linear combination), will result in a very unpredictable output from the arctangent function (see equation (2-12) from section 2.2.2). Even in cases of a static scene, the value of an edge pixel will vary greatly from one frame to the next. Because this type of error occurs specifically along the boundaries, they tend to be sparse and relatively easy to recover. In a 3D reconstruction, these pixels will generally appear by themselves in the space, unattached to any objects, and are thus referred to as *flying pixels*.

6. Motion Artifacts

A much related kind of error is that of motion artifacts. This refers to the pixels that cover multiple surfaces (similarly to the edge pixels described above) due to the motion of a foreground object (or camera motion). These errors do tend to occur near surface discontinuities, however unlike the case of edge pixels in static scenes, the motion artifacts will typically be several pixels wide, depending on the speed of the object. Also, this error tends to be more systematic in nature than the standalone edge pixels. For pixels at the edge of a motion artifact, their value will be derived from *mostly* one surface, and will tend to result in a depth value near that one surface, though as the pixels become more mixed, those derived depths become less predictable.

For implementations of time-of-flight that employ multiple frequencies as a solution to the phase ambiguity problem, there can be additional artifacts if the

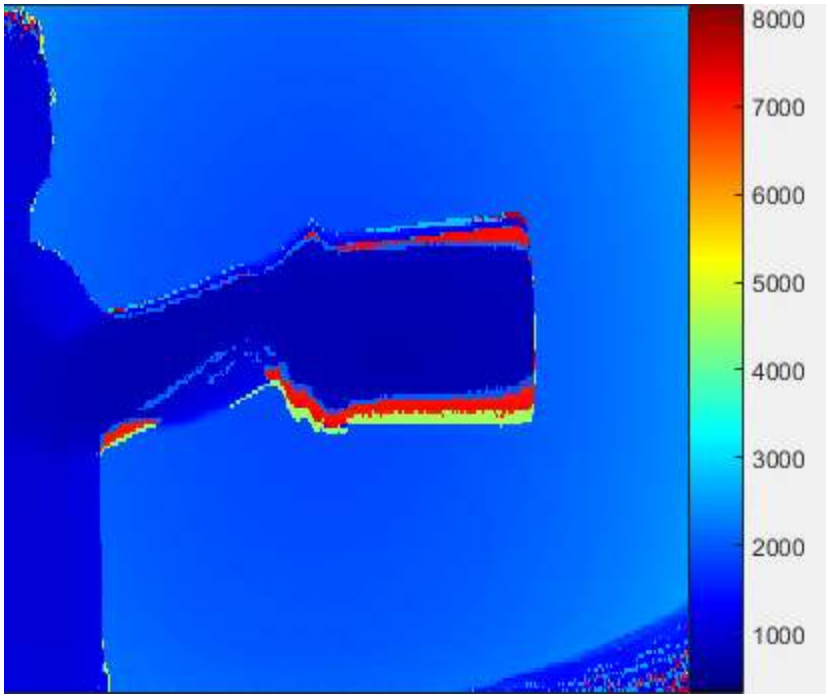


Figure 2.3. Motion artifact. As this object is being lifted, the sensor sees different combinations of foreground and background at different parts of the collection sequence. This leads to several different bands of differing depths.

individual frequency frames are captured in sequence [69]. Much of the time it takes to produce a single depth frame is not for the light capture itself, but rather the readout phase as the information captured by the sensor is digitized, so reducing the number of raw captures required to produce a depth frame is a key element to ameliorating motion artifacts. Mitigating this kind of error is one of the motivations for the work in Chapters 3 and 4.

7. Blooming

Another flavor of the mixed pixel effect is that of blooming. This is a case where the light that should be imaged by one particular pixel ends up cast onto other pixels due to unavoidable scattering by the lens [39]. It is related to the

artifact of lens flare in 2D imaging. It is most apparent in situations in which a foreground object is much much brighter than the nearby background pixels. This effect can be easily observed by moving one's hand in closely towards the lens; while the intensity image may look almost normal, in the range image the entire field quickly takes on the depth value of the hand as it moves closer, and this false reading appears to "bloom" from the hand.

8. Multipath Error

One of the toughest problems facing time-of-flight systems is that caused by multiple reflections with in a scene, in which case the light is reflected off of more than one surface before reaching the sensor. Because the light travels along more than one path to reach the sensor, the problem is sometimes called *multipath error.* The problem of multipath is fundamental to the principle of time-of-flight and there is currently no generally accepted solution [51].

In practice, this error is functionally similar to that of edge error and motion artifacts in that it is the result of mixed pixels. However, motion artifacts tend to be limited in nature, near object edges, are reasonably detectable, and are bordered by more stable pixels that can contribute to their correction. This applies more so to stationary edge pixels. The effects of multipath are not as easily identified or corrected.

Multipath errors tend to be most severe when imaging objects with concave geometry, which readily allows for surfaces to reflect onto each other at

close distances, and specular surfaces, in which very little of the light will be reflected back directly, and instead light coming back from further objects will easily be reflected back towards the camera. A common and prime example of this case is that of a shiny floor where it meets the wall. Light that reflects from the wall onto the floor and back to the camera will overpower the direct reflection, and due to the longer path, the floor will appear to drop down.

9. Phase Unwrapping

Phase unwrapping is fundamental problem to this method of indirect time-of-flight measurement, as well as many other phase-based measurement systems. It is unique from the other issues afore described in that it is not actually introducing error in the measurement, but it is an inherent limitation in the nature of the system that requires a solution in order for the measurement to extend past it. And the term limitation is used here quite literally, in that measurements can be made only up to the *unambiguous range* of $c/2f_m$ before some additional process is required to recover the true depth with some amount of confidence. The problem is addressed in much detail starting in section 3.2.1, and a solution to this problem is the impetus of the following two chapters.

Probabilistic Phase Unwrapping for Single-Frequency Time-of-Flight Range Cameras

Time-of-flight depth sensing technology (ToF) has matured in the last decade into one of the more common methods of range measurement, among the ranks of LIDAR, stereo depth, and structured light. These range sensing devices have become commercially available by manufacturers such as Mesa Imaging [81], Canesta [46], 3DV [57], and most notably Microsoft [106] with its second generation Kinect sensor for Xbox One. Because it is relatively inexpensive to produce and provides high capture rates, ToF range imaging is well suited for several applications despite a comparatively low accuracy and resolution.

Compared to other popular methods of range measurement, ToF offers its own advantages and disadvantages. Although it may be considered a subclass of scannerless lidar, ToF is distinct from typical scanning laser systems in that it has no moving parts and captures an entire 2D field simultaneously rather than scanning through the scene point by point. By capturing the entire scene at once, the ToF system is able to record depth at high frame rates. Though because of the need to illuminate the entire field of view it does so at a sacrifice of maximum range, depth precision, spatial resolution, and power consumption. This makes

ToF more suited towards close-range active applications (such as gesture-based user interface or interactive gaming) rather than precise 3D measurements of large static scenes. In comparison to stereo-based range imaging it requires much less computation, is not dependent upon a heavily textured surface, and not effected by ambient light. However, it is again limited in range and resolution as well as requiring more power for the active illumination. On the other hand, in comparison to structured light methods, ToF offers other advantages: because the setup is confocal it suffers no occlusion, and its resolution of measurement is independent of the hardware's geometry.

The term "time-of-flight" is in reference to the time it takes light to travel from the illumination source to the scene surface and back to the sensor. The illumination source is typically a near-infrared wavelength LED array or laser, placed very close (ideally, confocal) to the sensor. The returned light is imaged by a 2D CMOS array, making the hardware relatively inexpensive. Rather than measure the flight time of the light directly, the signal is modulated at a frequency f_m, and the phase shift θ of the recovered signal is an indication of the distance traveled. Specifically, the phase shift θ is related to the distance traveled by this equation:

$$2D = \frac{c\theta}{2\pi f_m} \tag{3-1}$$

where D is the distance from the camera to the surface and c is the speed of light.

There are multiple ways in which the phase of the signal can be measured; the hardware used in my measurements uses the method described in [46]. It is assumed that the intensity (*brightness*) image is available, in addition to phase measurement. This is usually the case in most ToF systems (e.g., the Canesta sensor).

Due to its cyclic nature, phase will wrap when it exceeds 2π, causing phase shift measurements to be ambiguous. For example, a measurement of 0.5π radian shift might signify an actual phase shift of 0.5π, 2.5π, 4.5π, etc. Formally, the phase θ in (3-1) can be decomposed as follows:

$$\theta = \phi + 2\pi K \tag{3-2}$$

where ϕ is the observable wrapped phase and integer K represents the number of phase wraps. If the target surface is within the *maximally unambiguous range* of $c/2f_m$, then phase (and thus depth) recovery is straightforward. However, for scenes exceeding this depth, it is necessary to determine the phase wrap number K at each pixel in order to recover the correct depth of the scene. The task of determining the value of K for the collection of pixels in the (wrapped) phase image is known as the *2D phase unwrapping problem*, for which I present a new solution here.

A solution to the 2D phase unwrapping problem is to image the scene with two (or more) different modulation frequencies f_m [46]. However, this approach requires the scene to be consistent between the exposures, and thus is potentially ill suited to situations in which the scene is changing rapidly, or where the camera

is moving. The method presented here uses just one frequency f_m, and so requires as few as capture periods. It relies on both the wrapped phase measurements and the brightness image to estimate the true phase, and thus the depth, of surfaces in the scene. The key innovation in this paper is in the derivation of the probability density function of the measured brightness from a surface illuminated by the sensor's light source, conditioned on the distance of the surface. The resulting expression is integrated in a Markov Random Field model defined on the wrapping numbers. Maximization of a suitable global objective function that includes a smoothness term defined on the estimated depth is performed via loopy belief propagation. Experiments with depth computation from phase and intensity images obtained from a single frequency ToF camera prototype show the effectiveness of our method when compared against the state of the art algorithm for single-frequency 2D phase unwrapping [21].

3.1. Related Work

The phase unwrapping problem is not unique to time-of-flight depth imaging. The problem is encountered in technologies such as synthetic aperture radar (SAR) and magnetic resonance imaging (MRI), acoustic imaging, and microwave interferometry [45]. The phase unwrapping task may be in one, two or three dimensions (magnetic resonance imaging, as an example, images a volume and thus is a 3-dimensional phase unwrapping problem). Depth sensing

time-of-flight images occurs on a 2-D CMOS sensor and requires two-dimensional phase unwrapping.

Standard solutions to phase unwrapping [45] recover the unwrapped phase via path integration: the phase jump between adjacent pixels is determined, and the surface is recovered by first choosing an initial unwrapped phase value for the first pixel, and then spreading out over the rest of the image by adding the appropriate phase jump for each adjacent pixel. Using this approach, however, the phase is recoverable only up to a constant, which needs to be decided for the initially chosen pixel. Moreover, the solution may be path dependent, meaning that if the phase jumps between pixels are chosen independently, there is no guarantee that they will all be consistent (different paths leading to the same pixel may imply different changes in phase). This leads to an artifact called a *residue*: the case where the sum of phase jumps around a square of four neighboring pixels does not equal zero.

Two families of algorithms for path integration are the *path following* methods and *minimum norm* methods. Path-following methods [47] [89] tend to either explicitly label discontinuities and choose paths that avoid them, or to grow the path (specifically, a tree) one step at a time without introducing residues [55]. Minimum norm methods assign an $\mathcal{L}_p$-norm penalty to phase jumps, and minimizes the sum of penalties over the entire grid [90]. This and other methods are described in detail in [45]. While minimum norm methods can be very efficient to calculate, they do not guarantee that residues will be completely

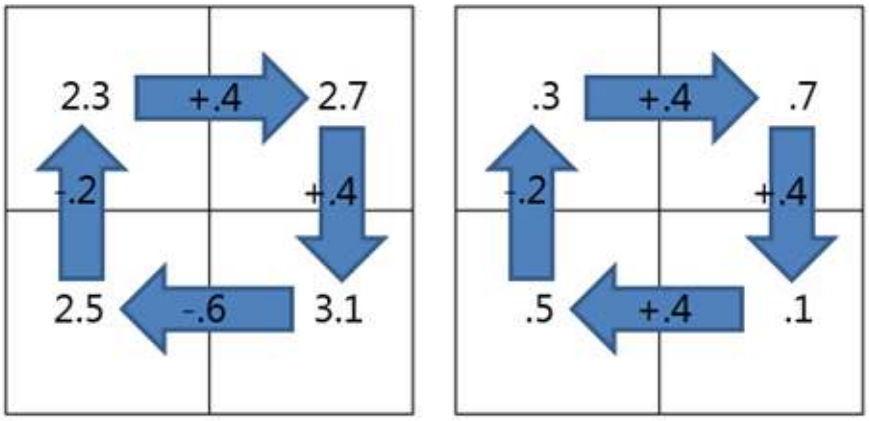


Figure 3.1. Phase jump residue. On the left is a sample of four pixels with the actual phase measurements and phase jumps between adjacent pixels. Note the sum as you proceed clockwise is zero. (b) On the right is the wrapped phase as would be measured, and the implied phase jump using a minimum norm solution. Here, the sum is not zero, creating a residue at this junction.

removed. Frey et al. address the residue issue by including a "zero-curl constraint" node in a Markov random field in [43], and this method was applied to time-of-flight imagery by Droeschel et al. [35]. In the solution proposed in this paper, the wrap number at each pixel is estimated directly, avoiding the possibility of residues entirely.

In theory, the need for phase unwrapping could be removed simply by increasing the unambiguous range. One method could be to decrease the modulation frequency; unfortunately, the measurement uncertainty increases proportionally with the unambiguous range [46]. Another way to increase the unambiguous range is through the use of multiple frequencies for the amplitude modulation of the active signal. This technique is adapted from INSAR technology [116] [109] and is a popular solution for ToF [34] [39] [20] [76]. If two frequencies are used, in successive or intermixed frame captures, then this pair can act as a

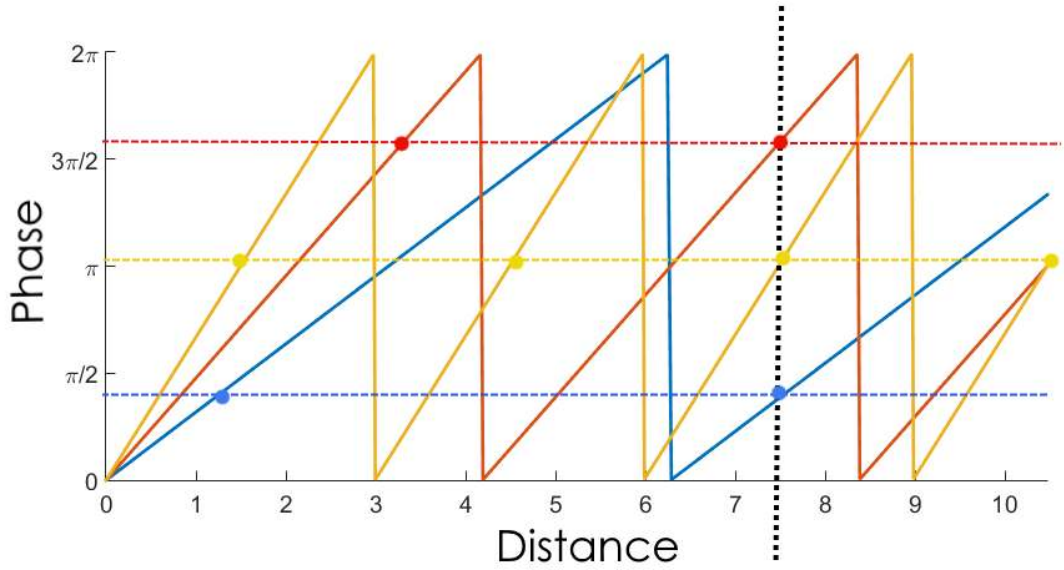


Figure 3.2. Multi-frequency phase unwrapping. For each of the frequencies, a wrapped phase is measured (denoted by the colored dot). Assume the measurements are not too noisy and the frequencies are chosen appropriate with respect to each other [32], there will only be one distance at which the unwrapped phases align.

single lower frequency. This will increase the unambiguous range by acting as an *effective frequency,* which is equivalent to the difference between the frequencies: $f_{eff} = |f_1 - f_2|$. This is alternatively expressed as the effective unambiguous range and is the least common multiple of the unambiguous ranges of the two frequencies. A 3-frequency approach is described in [10] that is very efficient by way of a look-up table while remaining robust to noisy measurements. A visualization of the principle behind the multi-frequency approach is provided in Figure 3.2. Multi-frequency phase unwrapping.Figure 3.2, inspired by [4]. One disadvantage of this approach is that it requires more power per frame to determine the distance. The key drawback to this approach, though, is that it requires more time to perform the multiple captures, rendering it more

vulnerable to motion artifacts. This is especially troubling for tasks such as gesture detection, where the target of interest is likely to be in motion. Some more recent work from the University of Waikato [86] uses multiple frequencies simultaneously, removing the need for some redundant samples. The overall capture time is reduced from consecutive multi-frequency, but still requires more time than a single capture.

Other solutions use just a single frequency, while incorporating additional information beyond wrapped phase. One example is given by technology that combines a ToF camera with a traditional stereo camera pair. Gudmundsson et al. [49] use the ToF data to bootstrap the stereo pair. Beder et al. [9] combine the data to optimize local "patchlets" of the surface. Several others [123] [28] [18] fuse the data under a probabilistic framework using a Markov random field. Choi and Lee [19] exploit stereo principles using a pair of ToF cameras.

Some other recent work takes advantage of the intensity image measured by ToF cameras, along with the phase data. My work in this chapter fits squarely within this line of research. Böhme et al. use a shading constraint (à la shape from shading) in order to smooth noisy ToF depth images [11], however they assume the phase is already correctly unwrapped. In [75] the scene is first segmented by the depth edges, then the average intensity of each segment is analyzed to determine whether it falls into the unambiguous range, which is based on a manually set threshold. Jongenelen et al. [59] use the intensity image along with a 2-frequency process to increase the confidence of their phase unwrapping.

Most comparable to my own work, though is that by Ouk Choi et al. [21], which employs a Markov random field framework whose data term is informed by the intensity image and smoothness term encourages adjacent pixels to be assigned wrapping numbers that result in comparable unwrapped phase values. Their algorithm first applies an EM optimization on the corrected intensity image (amplifies the intensity by a factor proportional to the square of the wrapped phase) [83] to determine the intensity threshold that will classify each pixel as being within or outside the unambiguous range. The resulting segmentation is smoothed using Grab Cut [94], and the data term penalizes a label which differs from the Grab Cut classification, increasingly so the further the corrected intensity is from the classification boundary. The MRF is optimized by belief propagation.

3.2. Method

I approach the phase unwrapping problem in a probabilistic framework, representing the number of phase wraps at each pixel as a discrete random variable. The novelty of this method is in how I exploit the available brightness information. Rather than looking at the residue in a closed path [43], I model the difference in depth between two adjacent pixels as a normal random variable with zero mean and fixed variance. Also utilized is the partial information about depth provided by the observed brightness. Brightness is a function of surface depth, slant, and albedo. Intuitively, bright pixels must correspond to nearby surfaces, whereas dark pixels may result from far away surfaces, or from surfaces that are

close to the camera but have low albedo and/or large slant angle. This phenomenon was already observed by Choi et al., [21]. However, in contrast to the early-commitment segmentation approach of [21], I define a probabilistic model for the relation between brightness and distance, and use it as "data term" in a global optimization approach that includes the depth smoothness constraint mentioned above.

3.2.1. Definitions and Problem Statement

For each frame contains of observation $\mathbf{O} = \{\boldsymbol{\phi}, \mathbf{B}\}$ consisting of a wrapped phase measurements[1] $\boldsymbol{\phi}$ and intensity (brightness) measurements $\boldsymbol{B}$. The goal is to recover the unwrapped phase θ at each pixel, given the observed wrapped phase ϕ, by estimating the phase wrap number K (then $\theta = \phi + 2\pi K$, see equation (3-2)). K is modeled as a uniform discrete random variable taking values between 0 and K_{max}. The maximum allowed number of wraps K_{max} is determined by the maximum distance of surfaces expected in the scene, or by the maximum distance at which a surface is visible (which depends on the power of the light source, which is typically known in advance.) The phase wrap number assignment ("labeling") is performed using a maximum-a-posteriori (MAP) criterion[2]:

---

[1] **Bolded** letters are used to indicate the collection of values (field) over the whole image. To denote the general case of a single pixel it is left un-bolded or with a subscript notation to denote the specific pixel, e.g. B_q, especially when referring to interactions between pixels.

[2] I will use the symbol $P(\cdot)$ to indicate probability distributions, and $p(\cdot)$ to indicate probability density functions.

$$\hat{\boldsymbol{K}} = \arg\max_K P(\boldsymbol{K}|\boldsymbol{B}, \boldsymbol{\phi}) = \arg\max_K p(\boldsymbol{B}|\boldsymbol{K}, \boldsymbol{\phi}) P(\boldsymbol{K}|\boldsymbol{\phi}) \quad (3\text{-}3)$$

The first term in (3-3), $p(\boldsymbol{B}|\boldsymbol{K}, \boldsymbol{\phi})$, represents the conditional likelihood of the brightness $\boldsymbol{B}$ given the wrapped phase $\boldsymbol{\phi}$. A model for this quantity is derived in the following section. The second term is the posterior distribution on the label configuration $\boldsymbol{K}$ given the wrapped phase observation $\boldsymbol{\phi}$. This term is derived from a smoothness prior on the depth field, as described in Sec. 3.2.3.

3.2.2. Intensity Model

The light intensity measured at each pixel can be attributed to four sources: light from the illumination source cast directly onto and reflected by the observed surface (direct reflection), light from the illumination source reflected indirectly off of neighboring surfaces (multipath reflection [60]), light from ambient sources other than the illuminator reflected off of the surface (ambient), and light from surfaces other than that directly observed, caused by lens defects or other unintended sources (stray light). As described in 2.2.1, the ambient light is easily removed from the image. Here, I consider only the direct reflection component of the measured brightness, which is usually the dominating component.

The model assumes that illumination is from a point source, co-located with the camera. This means that the line of sight through a pixel coincides with the line of propagation of light illuminating the surface element imaged by that pixel. The illumination power, though, is typically not homogeneous, meaning that

the transmitted power depends on the direction of propagation. The power distribution across pixels can be calibrated for a specific camera hardware, for example by measuring the brightness reflected off a planar surface of constant albedo, and compensating for the distance and angle of incidence of the surface element imaged by each pixel. After this calibration procedure, the light power along a propagation line through a pixel q can be represented by the brightness L_q measured at q from reflection against a surface element of albedo 1, at distance of 1 meter from the camera, orthogonal to the line of sight. If assumed that the visible surfaces are Lambertian, we can thus model the observed intensity B_q at a pixel as

$$B = \frac{L \cdot \rho \cdot \cos\beta}{D^2} \tag{3-4}$$

where ρ is the surface albedo, β is the angle of the surface normal with respect to the line of sight (slant), and D is the distance from the camera to the surface element imaged by the pixel.

Hence, the brightness measured at a pixel depends on the (unknown) distance, slant, and albedo of the surface element. (Note that the measured brightness will be affected by Poisson noise; however, for the sake of simplicity, it is assumed to be noise-free.) Thus in the ideal case, if each of these values are given, the conditional probability density of the brightness can be modeled as a Dirac delta function:

$$p(B|D,\rho,\beta) = \delta\left(B - \frac{L \cdot \rho \cdot \cos\beta}{D^2}\right). \tag{3-5}$$

I will make the following simplifying hypotheses: (a) the orientation of surfaces in the scene is modeled as a uniformly distributed random variable, which can be shown (see Appendix A) to result in a probability density function for the slant angle β equal to:

$$p(\beta) = 2\,\sin\beta\,\cos\beta \tag{3-6}$$

(b) the surface albedo ρ is a random variable uniformly distributed between 0 and 1; (c) the surface normal and albedo at a pixel are independent of the normal and albedo at other pixels. We thus obtain:

$$p(\boldsymbol{B}|\boldsymbol{D}) = \prod_q p(B_q|D_q) \tag{3-7}$$

with

$$\begin{aligned} p(B_q|D_q) &= \int_0^1 \int_0^{\frac{\pi}{2}} p(B_q|D_q,\rho_q,\beta_q)\,p(\rho_q|\beta_q)\,p(\beta_q)\,d\beta_q\,d\rho_q \\ &= 2\int_0^1 \int_0^{\frac{\pi}{2}} \delta\left(B_q - \frac{L_q \cdot \rho_q \cdot \cos\beta_q}{D_q^2}\right)\sin\beta_q\cos\beta_q\,d\beta_q\,d\rho_q \qquad (3\text{-}8) \\ &= \begin{cases} \dfrac{2D_q^2}{L_q}\left[1 - \dfrac{B_q \cdot D_q^2}{L_q}\right], & \text{if } 0 \le B_q \le L_q/D_q^2 \\ 0, & \text{otherwise} \end{cases} \end{aligned}$$

A more detailed derivation can be found in Appendix B, as well as for the posterior probability density:

$$p(D_q|B_q) = \begin{cases} \frac{4B_q \cdot D_q}{L_q}\left[1 - \frac{B_q \cdot D_q^2}{L_q}\right], & \text{if } 0 \le D_q \le \sqrt{L_q/B_q} \\ 0, & \text{otherwise} \end{cases} \quad (3\text{-}9)$$

A comparison between the distributions can be found in Figure 3.3. While the shape of the distributions are not wildly dissimilar, the likelihood function $p(B_q|D_q)$ clearly favors further distances.

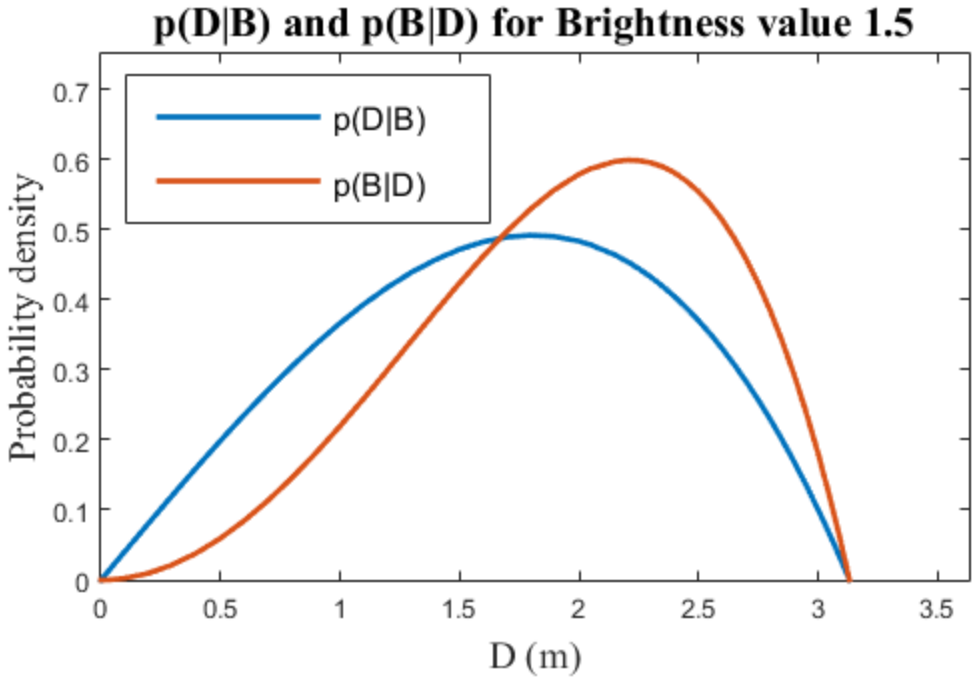


Figure 3.3. Conditional probability density and conditional likelihood for distance given measured brightness. Note that as the likelihood distribution, as a function of distance is not a proper probability distribution and was normalized to integrate to 1 for this comparison.

It is instructive to look at the posterior probability density function $p(D_q|B_q)$ as it changes with different observations of brightness. As seen in Figure 3.4 and Figure 3.5, the spread of this density tightens with increasing values of B_q.

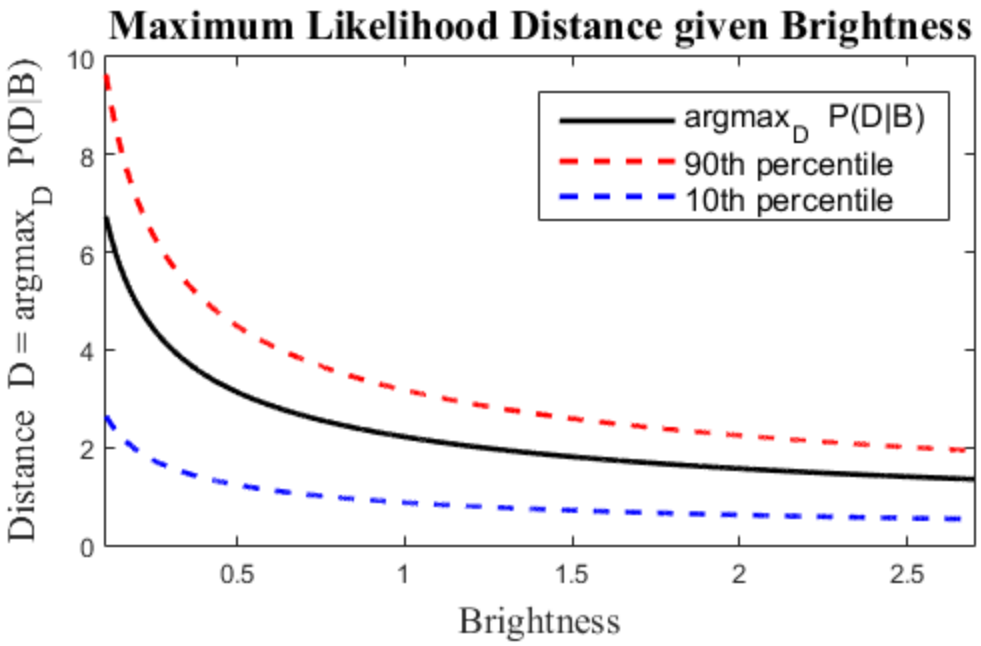


Figure 3.4. Maximum likelihood of distance, given brightness. Outlining probability density of D for values of B. Black line shows most likely distance given measured brightness. Colored lines indicate the 10th and 90th percentiles. Curve is plotted over the observed range of Brightness and distance values, computed using a representative value for I.

This can be explained by the fact that a small brightness value at a pixel can be due to a surface being far away, but also to a surface having low albedo or being at a large slant angle. Hence, a dim pixel cannot provide much information about the surface distance alone, resulting in a large spread of $p(D_q|B_q)$. Conversely, *bright pixels can only be generated by a surface close to the camera*[3], with large albedo and small slant angle, resulting in a narrow spread of $p(D_q|B_q)$. This means that darker pixels have a high classification error rate, consistent with the increasing spread of the posterior density $p(D_q|B_q)$ with decreasing brightness. However,

[3] An exception to this is for specular reflections, which are not being considered in this Lambertian model

the misclassification rate is much smaller for brighter pixels; these pixels function as "anchor points" in the belief propagation step described in Sec. 3.2.4.

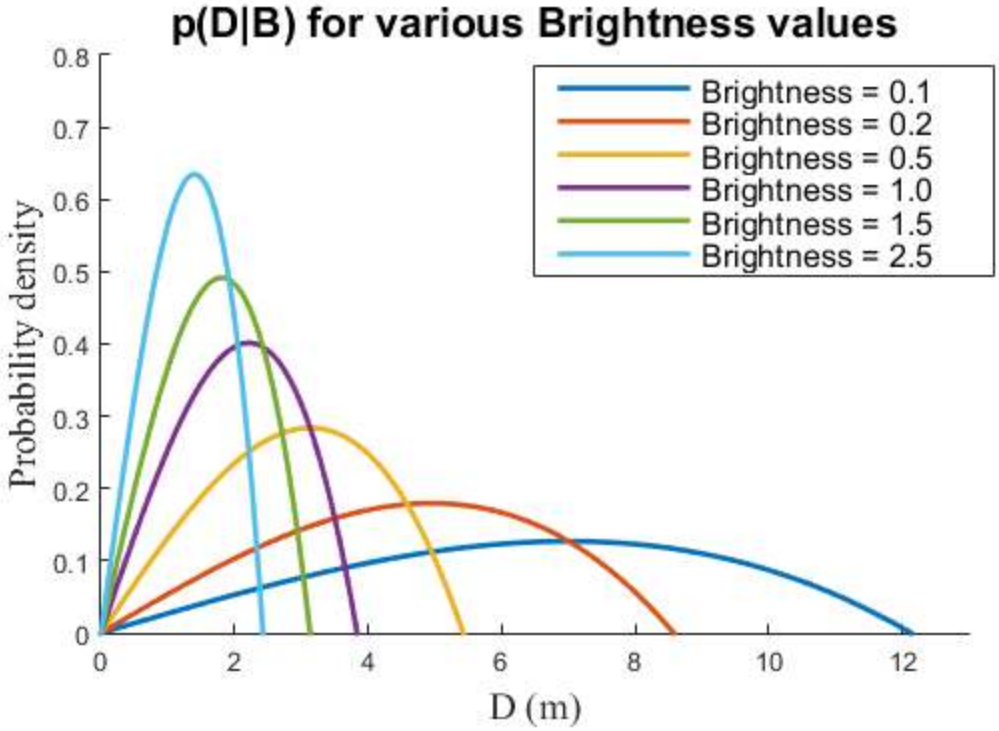


Figure 3.5. Conditional probability density for distance given measured brightness. For very bright values, surface must be nearby. For dim, it is more spread over the range.

3.2.3. Smoothness Model

The "smoothness term" $P(\boldsymbol{K}|\boldsymbol{\phi})$ in (3-3) formalizes the notion that neighboring pixels are expected to have similar depths, and thus similar value of the unwrapped phase θ. The phase difference between neighboring pixels is modeled as a Gaussian with mean 0 and variance σ^2:

$$p(\theta_q - \theta_p) = \mathcal{N}(\theta_q - \theta_p; 0, \sigma^2) \tag{3-10}$$

where $\mathcal{N}(\cdot;\mu,\sigma^2)$ represents the Gaussian probability density. Note that $\theta_q - \theta_p = 2\pi(K_q - K_p) + (\phi_q - \phi_p)$. It will be assumed that: (a) the random variables ϕ_q and ϕ_p have uniform joint distribution; (b) the "dimension reduction" $\phi_q - \phi_p$ is sufficient, meaning that $P(K_q - K_p|\phi_q,\phi_p) = P(K_q - K_p|\phi_q - \phi_p)$; (c) the joint statistical description of the variables $(\phi_q - \phi_p)$ and $(K_q - K_p)$ is fully represented by their sum: $p(\phi_q - \phi_p, K_q - K_p) = p(\theta_q - \theta_p)$. Under these simplifying conditions, it is easy to prove that

$$P(K_q - K_p|\phi_q,\phi_p) \propto \mathcal{N}(K_q + \phi_q - K_p - \phi_p; 0, \sigma^2) \qquad (3\text{-}11)$$

This expression can be used to establish a "factor potential" between neighboring pixels in a MRF labeling, as discussed in the next section.

The wrapping number could be inferred based on this smoothness term alone, using the global optimization framework described in the next section but neglecting the brightness term. Note that in this case the solution can be computed only up to a constant wrapping number (since adding any constant wrapping number does not change the smoothness term).

In closing this section, note that a more robust version of the smoothness term in (3-10) could be used to account for depth discontinuities. In my experiments, however, this did not lead to any noticeable advantage.

3.2.4. Global Optimization

I estimate the label configuration $\widehat{K}$ that maximizes $P(\boldsymbol{K}|\boldsymbol{B},\boldsymbol{\phi})$ in (3-3) by defining a global objective function $E(\boldsymbol{K})$ defined as:

$$E(\boldsymbol{K}) = \lambda \sum_{q} E_{B,\phi}(K_q) + \sum_{(p,q)} E_{\phi}(K_q, K_p) \tag{3-12}$$

where λ is a weighting value, $E_{B,\phi}(K_q) = -\log p(B_q|D_q)$, from (3-8) $E_{\phi}(K_q, K_q) = \log P(K_q - K_p|\phi_q, \phi_p)$, from (3-11); and (p,q) represent pairs of neighboring pixels. Inference is performed by loopy belief propagation as described in [41] [105] using a simultaneous message passing schedule, with an 8-connected neighborhood. We cease iterations when one of the convergence criteria are met: either the change in total sum log marginalized probability is below a threshold, or there is no change in labels for a set number of iterations.

3.3. Experiments

The algorithm was tested on a set of 45 images (consisting of wrapped phase $\boldsymbol{\phi}$ and brightness $\mathbf{B}$) collected from 3 different locations: a home, an office setting, and a computer lab. Only indoor scenes—the primary setting for ToF sensors—were chosen to be included as a known issue for active illumination sensors is excessive ambient light. We attempted to capture scenes with a variety depths, so to include a range of difficulties for which to test the algorithm. Data was captured using a Canesta XR670 camera with a resolution of 320×200 pixels.

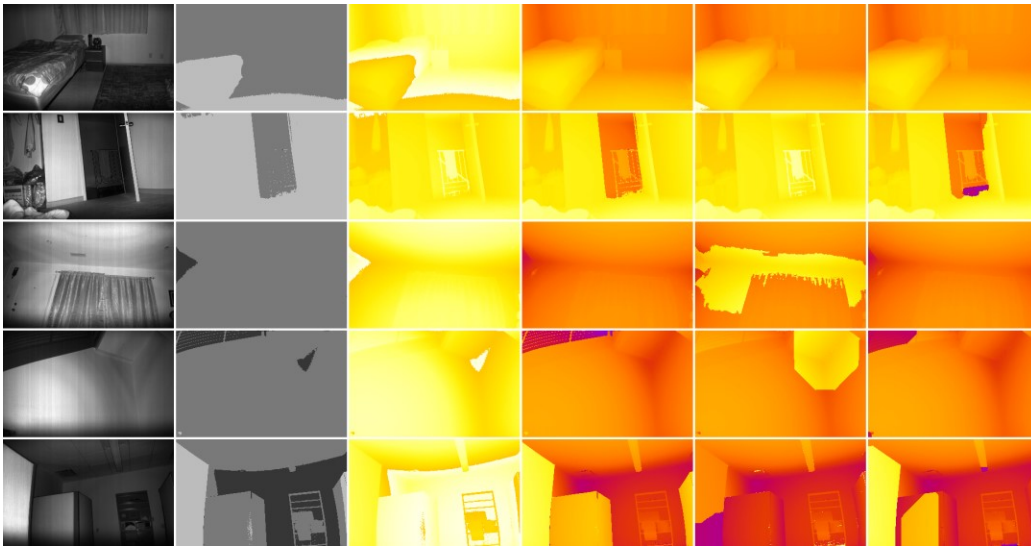


Figure 3.6. Selected scenes from Experiment 1. The top row is taken from the *easy* group, second and third rows are from the medium and the bottom two from *hard*. The first column shows the intensity image from the active illumination. The second column is the ground truth number of phase wraps, from 0 to 3 wraps with darker greys being more wraps. The 3rd through 5th columns show phase, increasing from hot white to cool purple. The 3rd column shows the measured phase (wrapped at 2π). The fourth column is representation of the unwrapped phase or depth, as measured by a multi-frequency ToF camera. The final two columns compare the best solutions from the method by [21] to my own algorithm at the end.

Each static scene was captured at two frequencies, 51.4MHz and 68.6MHz, which determined unambiguous ranges of 2.2m and 2.9m respectively. Ground truth was determined by combining these pairs of captures using the approach of [46], which uses both frequencies to obtain an unambiguous phase measurement. I ran the phase unwrapping algorithm on both frequencies individually. In addition, the algorithm was tested on an extended range of modulation frequencies by synthesizing wrapped phase data from the data at the two original modulation frequencies. This was obtained starting from the "ground truth" depth

data (obtained as described above), converting the depth into phase according desired modulation frequency, and wrapping at 2π radians. Synthetic phase images were created with frequencies ranging from 50Mhz to 150Mhz (unambiguous range of 1 - 3m).

Results of the algorithm were compared against the method of Choi et al. [21], which may be considered the state of the art for *single frequency* solutions. This is a natural comparison, as both methods incorporate the intensity image. The key difference is that in [21] each pixel is classified as bright or dim based on the distribution of intensity in that image, with no consideration for the albedo or surface normal. Thus the contribution from the intensity image is binary in nature, supporting each pixel as near (0 wraps) or far (1 or more wraps), which is reasonable in the case of at most 1 wrap but becomes less effective with more wraps (as is demonstrated clearly in my latter experiments with high modulation frequencies) . Since the original implementation of Choi's algorithm was not available (due to proprietary restrictions), nor were the original data, I implemented the algorithm in Matlab following the algorithm's description in [21] in order to make a comparison. As the precise method of belief propagation was not described, I used my own implementation of simultaneously scheduled loopy belief propagation—and whether it is more or less effective than the optimization employed in the original work, it at least puts the comparison of the energy terms on equal ground.

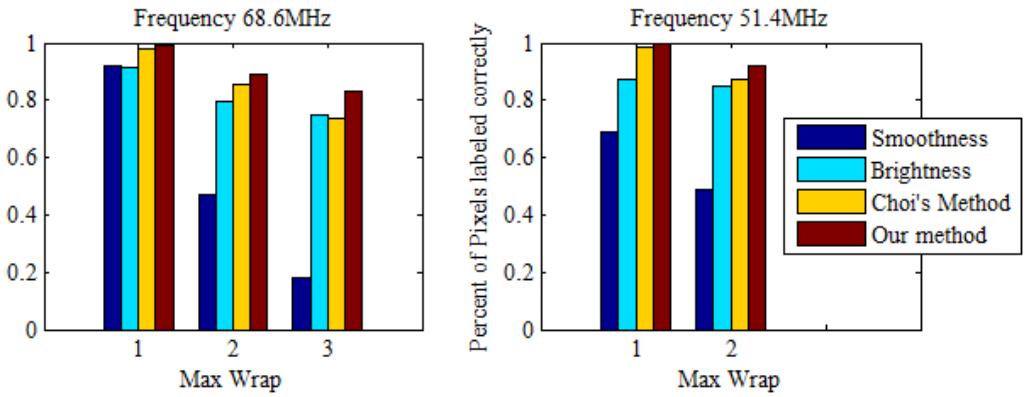


Figure 3.7. Results on real data. These bar graphs compare, for each frequency, the performance of the individual terms of the proposed solution, Smoothness (blue) and Brightness (cyan), alongside the method proposed by Choi et al. (yellow) and my complete proposed solution (red). The data is divided by the maximum number of wraps, as the phase unwrapping problem becomes more difficult with a larger range of phase. Note that for 51.4MHz, there were never more than 2 phase wraps.

The data was categorized by the maximum number of phase wraps observed in each scene, either 1, 2, or 3, (that is up to 4× the unambiguous range). The data is further subdivided by frequency in Figure 3.7, where I summarize the performance of the algorithms. Alongside I present the performance of our smoothness and brightness terms alone. For the smoothness term alone, I set $\lambda = 0$ and ran the loopy belief propagation. For the brightness term, rather than using belief propagation, I simply choose the value of $\mathbf{K}$ that maximizes $P(\boldsymbol{B}|\boldsymbol{K},\boldsymbol{\phi})$, as defined in (3-8).

In the 14 *easy* test cases (up to 1 phase wrap), the proposed method outperformed Choi's method with average of 99.4% correctly labeled pixels compared to 98.3%. A selection of results is displayed in Figure 3.6 (first three rows). For example, the first row shows a simple bedroom scene. Note that my algorithm is able to recover the section of floor in the bottom left corner beneath

the bed. In the second row of Figure 3.6, my algorithm is able to correctly identify the small area through the door as being farther away.

For the 45 scenes with 2 phase wraps, both algorithms have a tougher time, but the advantages over Choi start to become more apparent: successfully labelling still 91.4% of the pixels versus 87.1%. Examples of these scenes are shown in Figure 3.6, last two rows. Careful analysis shows that the key cause for this bad performance was due to a misclassification of lighter and darker pixels as nearer and farther in the early stage of Choi's algorithm. For example, in the case shown in the last row of Figure 3.6, the low albedo of the television set results in dark pixels that are incorrectly classified as far. Similar pitfalls are avoided with my algorithm, as dark pixels are considered uninformative under my image brightness model (see Figure 3.5) and can be supported through the smoothness term by brighter pixels in the foreground.

In the hardest of cases, with 3 wraps, the gap continues to widen, with 83.3% against Choi's 73.8%. This trend goes on as is demonstrated by increasing the maximum number of wraps by synthesizing cases using the same depth and brightness values, but computing new values for ϕ as they would appear for higher frequencies.

The results of experiments with synthetically produced phase images over varying modulation frequency are shown in Figure 3.8. I tested at 7 different simulated frequencies for all 45 test scenes at half-resolution. As the frequency

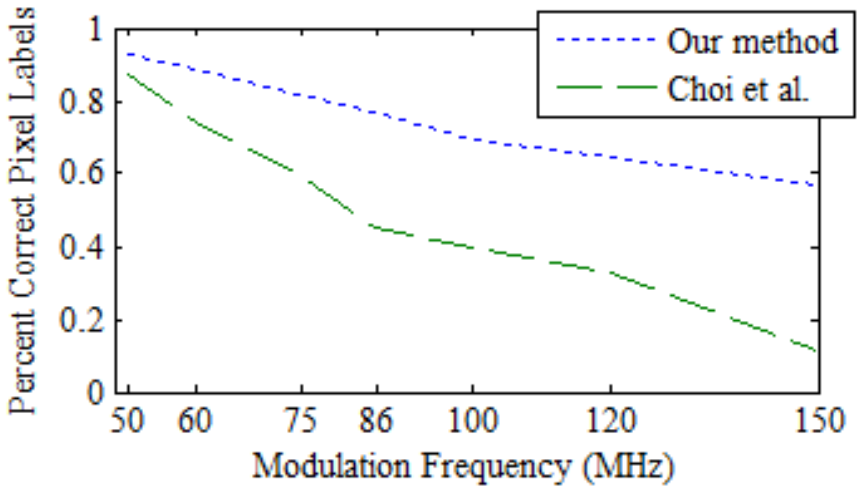


Figure 3.8. Results on semi-synthetic data. Averaged over all 45 scenes at each frequency, regardless of the max wrap value. The maximum wrap value reached (at 150MHz) was 8.

increases (and the unambiguous range decreases), the phase unwrapping problem becomes more challenging, given the larger number of possible wrap values for the same maximum distance. Both methods show a decrease in performance for increasing modulation frequency, although this decrease is relatively minor with my method. Although the method is presented as applying to multiple wraps, Choi's algorithm clearly fails in the case of high modulation frequency.

I implemented the loopy belief propagation algorithm in Matlab. For the 90 scenes worth of real data, a grid search was performed over the parameters λ and σ to find the best performance in terms of correctly labeled pixels. The algorithm's convergent criterea was chosen to be a change in negative log marginal probability of less than 1e-10 or if no change in label after 4 consecutive iterations (it was not uncommon that underlying marginal probabilities would be

updated and propagated for a handful of frames before the next change in labels resulted). Each scene converged after an average of 330 iterations, with an average time of about .5 seconds per iteration.

3.4. Conclusions

In this chapter I presented an approach for phase unwrapping of time-of-flight systems with the potential to improve frame rates and avoid motion artifacts. The wrap number is estimated directly, which guarantees no residues and provides an exact solution, as opposed to a solution "up to a constant." This method makes use of valuable information from the signal intensity, without resorting to early-commitment, segmentation-based procedure as with other previous approaches. The experimental results show that the proposed algorithm outperforms existing methods that also use brightness information. Future work, such as additional priors such as local smoothness of albedo and surface normal (as presented in the following chapter), could be easily be integrated in the MRF framework.

This work leaves open the possibilities of including additional priors, such as smoothness of albedo and surface normal in a localized area, or an estimate of the surface normal from wrapped phase measurements. Literature on the statistics of natural images, or specifically albedo, tends investigate the spatial variation of albedo across a surface rather than looking at an overall distribution of albedo in images [56]. So while further investigation may lead to a more

descriptive distribution, it is hypothesized in this work that simply including this basic range of values for albedo adds significant information to the model. An explicit treatment of the albedo and surface normal, if included in the smoothness term, would be expected to greatly enhance the precision of this model. Barron and Malik demonstrate in [7] the power of applying local smoothness constraints to these terms, even under unknown lighting conditions. However, the task of simultaneously optimizing these additional terms increases the complexity so as to likely render it unfeasible for real-time applications.

Fast Single-Frequency Time-of-Flight Range Imaging

4.1. Introduction

Range imaging is becoming an essential component in applications such as automotive, augmented reality, natural user interface, biometrics, computational photography, and more. There are currently three main methods for range imaging: stereo or multi-camera triangulation; triangulation from projected patterns (structured light); and time of flight (ToF) measurements. This contribution focuses on ToF technology, which has already been implemented in multiple products such as Microsoft Kinect for Xbox One, PMD, Intel RealSense.

ToF cameras are comprised of an illuminator producing modulated infrared light, and an imaging sensor that is synchronized with the illuminator. The imaging sensor computes the phase difference (or shift) θ between the transmitted and received wavefronts, along with the intensity (magnitude) B of the received light (irradiance) [29]. Light reflected by a surface at distance D has a phase measurable shift equal to

$$\phi = (4\pi f_m D/c) \bmod 2\pi \qquad (4\text{-}1)$$

where f_m is the modulation frequency of the illuminator, and c is the speed of light. The distance (range) to the surface can thus be recovered from ϕ, but only up to

a multiple of the *wrapping distance* $D_m = c/2f_m$, or unambiguous range. This is the well-known *phase unwrapping* problem of ToF sensing. To deal with this problem, most commercial cameras use the so-called multi-frequency approach: two sets of phase measurements (or even three to be more robust to noisy measurements) are taken of the same scene, where the illuminator is modulated at different frequencies f_m in the two sets of phase measurements. By analyzing the two (or more) phase shift images, the distance D to each visible surface element can be uniquely recovered.

Multi-frequency phase unwrapping assumes that the scene has not changed between the two images. If this is not the case (e.g., if the camera is moving), the result is not accurate. While this is the most widely accepted solution for phase unwrapping in time-of-flight devices, it has been shown that multi-frequency is *not* necessary for phase unwrapping. Other techniques have been demonstrated that leverage knowledge of the measured intensity B to recover the "unwrapped" distance D from a *single* phase image. These techniques rely on the observation that the intensity B of light reflected by a Lambertian (opaque) surface is inversely proportional to the square of the surface distance D – suggesting that some information about D could be inferred from B. Unfortunately, the intensity B is also affected by the (unknown) albedo ρ and slant angle β of the surface.

To reduce the uncertainty of inference, it is thus necessary to impose additional constraints, such as spatial smoothness priors, typically expressed in

the form of a Markov Random Field (MRF). While this approach has produced impressive results [23], it requires use of techniques such as belief propagation or graph cuts, which are computational demanding and preclude frame-rate processing. Indeed, Markov random fields are a very popular paradigm for computer vision problems (perhaps in part because the image lattice is so easily described in such a way), and so the optimization of such models is a longstanding open problem. Proposed solutions are abundant, and even surveys [30] [105] of the variations are far from exhaustive. In this work, I depart from the formal MRF framework in order to pursue a much more efficient (and greedy) approach that still attempts to satisfy the same goal of balancing the *data cost* imposed by a labeling with the spatial smoothness constraints we expect from the scene whose depth is to be measured.

In this chapter, I present the development of a computational single-frequency ToF camera that solves the phase unwrapping problem with accuracy comparable to or exceeding the state of the art for a single frequency, and speed that, at up to 0.3 seconds per frame, is orders of magnitude faster than previous approaches. Specifically, this work presents three main contributions. First, I show that the surface slant angle β can be computed to a good approximation even *before* unwrapping the phase shift. Knowledge of β reduces the uncertainty in the determination of D. Second, I demonstrate that this surface normal estimate and intensity information can be gainfully incorporated into the similarity metric used to enforce spatial coherence, and finally I impose my smoothness prior using the

fast Non-Local Cost Aggregation algorithm, which was recently proposed for stereo matching. This non-iterative algorithm is extremely efficient, requiring only a few operations per pixel per wrap number.

4.2. Method

In time-of-flight imaging, each observation $O = \{\phi, B\}$ is initially composed of four distinct intensity images $\{I_0, I_{\pi/2}, I_{\pi}, I_{3\pi/2}\}$ captured at specific time intervals with respect to the illumination modulation cycle, each staggered by $\pi/2$ radians with respect to the modulation phase. As described in more complete detail in section 2.2.2, using the difference of image pairs offset by π, $I_{0^\circ} = I_{\pi} - I_0$, and $I_{90^\circ} = I_{3\pi/2} - I_{\pi/2}$, and with simple computation can recover the phase offset from the illumination modulation $\phi = \arctan(-I_{90^\circ}/I_{0^\circ})$, and the active scene illumination $I = \sqrt{I_{0^\circ}{}^2 + I_{90^\circ}{}^2}/2$. As mentioned in the Introduction, the sensor measures the "wrapped" phase shift ϕ_p at each pixel, where the subscript indicates the pixel index. The goal is to recover θ_p, the actual ("unwrapped") phase difference, from which the distance to the surface D_p is obtained as by $D_p = c \cdot \theta_p / 4\pi f_m$. The terms θ_p and ϕ_p are related as by $\theta_p = \phi_p + K_p 2\pi$, where K_p is the (unobservable) "wrap number". Thus the goal is achieved by determining the wrap number K at each pixel, using the observed data: wrapped phase shift ϕ and intensity B.

Drawing from my work in Chapter 3, the assignment discrete values to the set $\boldsymbol{K}$ of wrapping values over the image is based on maximizing a probabilistic term. Here I explore two possible approaches: conditional posterior probability of the distance D_p given B_p and an estimate of the slant $\hat{\beta}_p$: $P\left(D_p|B_p,\hat{\beta}_p\right)$, and the conditional likelihood $P\left(B_p|D_p,\hat{\beta}_p\right)$. While the terms are closely related and have similar forms, they are distinct. Neither term can be decidedly stated as the "right" term to use, though as I found through experimentation, the conditional likelihood does perform better in this task.

A key motivation of this single-frequency approach to time-of-flight phase unwrapping is to reduce the time required to produce a single depth frame. The results from my initial attempts demonstrated that the information contained in a single-frequency phase measurement may be sufficient to perform phase unwrapping, but the use of loopy belief propagation to optimize the solution requires too much computation to be considered a viable technique for real-time range imaging using standard hardware. In this work I enforce consistency between neighboring pixels using an efficient approximation of a globally optimized solution, borrowing from a recent approach to stereo disparity matching.

4.2.1. Intensity Model

The imaging sensor can be receive light from at least four sources: direct reflection of the source illumination off of observable surfaces, indirect reflection

from the source (multi-path), ambient illumination sources, and stray light from unintended sources (e.g. lens defects). Additionally, the reflected light may be specular or diffuse. As in the previous chapter, I only consider the direct reflection, and in that only diffuse reflection off of a Lambertian surface. Recall the ambient light is more or less eliminating due to the differencing of raw intensity image pairs, offset by a half period, as is the design of the 2-tap pixels. Stray light is typically the result of hardware defects or conditions with extreme ambient light, and can be reasonably set aside as a rare or extreme case.

Multipath and specular reflectivity, on the other hand, are both not-uncommon in typical use cases, and in fact, specular surfaces are a common cause of severe multipath effects.

I include the assumption that the visible surfaces are Lambertian (which also means that incorrect results are expected in areas with high specular reflection). The illuminator is mounted on the camera itself, as close to the lens as possible. It is modeled as a point light source, located at the camera's optical center. If the illuminator were an ideal isotropic point source, then the irradiance B_p at a pixel would be related to the (constant) radiant intensity L from the point source as by $B_p = L \cdot \rho_p \cdot \cos\beta_p / D_p^2$, where ρ_p is the albedo of the surface element imaged by pixel p, and β_p is its slant angle (the angle between the surface normal and the line of sight). In practice, the radiant intensity (that is, the light power emitted per solid angle) is not uniform, nor is the pixel sensitivity to light arriving from multiple directions (due to multiple reasons, including the effect of optical

elements). This non-uniformity can be calibrated *a priori*, resulting in a distribution L_p of equivalent radiant intensity (or *light profile*, shown in Figure 4.5). This allows us to specify the general model of observed intensity B_p at pixel p as

$$B_p = \frac{L_p \cdot \rho_p \cdot \cos\beta_p}{D_p^2}. \tag{4-2}$$

This expression for the measured intensity was used in [23] (3.2.2) to derive the conditional likelihood of B_p given the distance D_p under the assumption that the albedo and the surface orientation are uniformly distributed random variables. In fact, we observe that the surface normal could, to some approximation, be computed from the wrapped data ϕ. More specifically, suppose one reconstructs a surface patch from distances computed from the measured wrapped phase ϕ, assuming a constant wrap number K. A generic 3-D surface point $P_p^{(K)}$ in this surface has distance $D_p^{(K)}=c\cdot(\phi_p + 2K\pi)/4\pi f_m$. The normal $N_p^{(K)}$ to the so computed surface patch at $P_p^{(K)}$ can be estimated at each wrapping value K by projecting all of the pixels in the neighborhood $\mathbb{N}_p$ according to the wrap value K and the wrapped phase measurements $\boldsymbol{\phi}_{\mathbb{N}_p}$. Hence, one can approximate the slant angle β_p of the actual surface patch imaged by p by the slant angle of the reconstructed patch for a fixed K (e.g., K=0). An example of reconstructed surface orientation is shown in Figure 4.5.

Note that this approximation fails catastrophically if the actual surface patch is at a distance that is a multiple of c/f_m, that is, exactly where the phase shift undergoes a wrap, as some pixels in the neighborhood $\mathbb{N}_p$ will be on either side of the phase wrap. This problem can be avoided by choosing the value of K at each pixel that causes it to be nearest to $P_p^{(K)}$ rather than simply assigning the same wrap value to all pixels. This significantly increases the computation necessary to estimating the normals, not just in the choosing the values of K for each neighborhood, but in that it precludes the use of the integral trick to estimate normals for the entire image collectively. However, there is another technique that can still allow for this efficient computation. For each wrap value K two projections are computed: one in which all pixels are assigned the same wrap value, and one in which pixels with a wrapped phase measurement $\phi_p > \pi$ are assigned a wrap value one less. This effectively moves the wrapping boundary from 2π radians to π. Then, the choice of the normal $N_p^{(K)}$is decided based on whether or not $\pi/2 < \phi_p < 3\pi/2$.

Based on the estimation $\hat{\beta}_p$ of the slant angle β_p, and neglecting sensor noise, one can use (4-2) to compute the conditional likelihood $\mathcal{L}(D_p|B_p,\beta_p) = P\big(B_p|D_p,\beta_p\big)$ with a proper prior distribution of the albedo ρ_p. The derivation is detailed in Appendix C. One can similarly compute the conditional posterior probability of the distance D_p given B_p and β_p: $P\big(D_p|B_p,\beta_p\big)$. Note that this expression for the posterior distribution of D_p does *not* take into account the

measured phase shift ϕ_p. In fact, we can combine the two to obtain a (marginal) posterior probability distribution of the wrap number K_p as by:

$$P\left(K_p|\phi_p, B_p, \beta_p\right) = P\left(D_p|\phi_p, B_p, \beta_p\right) \tag{4-3}$$

with $D_p \in \left\{c \cdot \left(\phi_p + K_p 2\pi\right)/4\pi f_m \middle| K_p = 0, \dots, K_{max}\right\}$.

If the slant angle β_p is assumed known, and assuming a prior uniform distribution for the unknown albedo ρ_p, it is easy to see that $P\left(B_p|D_p, \beta_p\right)$ is also uniformly distributed between 0 and $L_p \cdot \cos\beta_p / D_p^2$. However, it was noted that the estimation of β_p is often inaccurate, in part because, as mentioned earlier, the actual surface normal depends on the (unknown) wrap number K_p, though more so because surface normal estimation is notoriously noisy. Hence, rather than assuming a fixed value for β_p, I model the slant angle by means of a normal distribution centered at the estimated value $\hat{\beta}_p$ with standard deviation σ_β. The resulting form for $p\left(B_p|D_p, \beta_p\right)$ becomes more complex and does not lend itself to a closed form expression. It can, however, be pre-computed and stored in a reasonably sized two-dimensional look-up table. A description of how this term is derived can be found in Appendix C:

$$p(B|D, \beta) = \frac{D^2}{L \cdot \sigma_\beta \sqrt{2\pi}} \int_{\rho=\frac{BD^2}{L}}^{1} \frac{e^{\frac{-\left(\cos^{-1}\left(\frac{BD^2}{\rho L}\right) - \hat{\beta}_q\right)^2}{2\sigma_\beta^2}}}{\sqrt{\rho^2 - \left(\frac{BD^2}{L}\right)^2}} d\rho. \tag{4-4}$$

And similarly one can derive the posterior probability distribution for the distance given an observed brightness:

$$p(D|B,\beta) = \frac{2B \cdot D}{L \cdot \sigma_\beta \sqrt{2\pi}} \int_{\rho=\frac{BD^2}{L}}^{1} \frac{e^{\frac{-\left(\cos^{-1}\left(\frac{BD^2}{\rho L}\right)-\hat{\beta}_q\right)^2}{2\sigma_\beta^2}}}{\sqrt{\rho^2-\left(\frac{BD^2}{L}\right)^2}} d\rho. \qquad (4\text{-}5)$$

It can be informative to view the distribution under different conditions to help get a sense of how strongly the various parameters influence the overall distribution, and to confirm that it matches with our intuition.

Plots of the distributions of both the posterior conditional probability of the distance given brightness and slant as well as the likelihood distribution are compared in Figure 4.2 and Figure 4.3, using set values for the estimate of the slant $\hat{\beta} = \pi/4$ and its standard deviation $\sigma_\beta = .3$. The posterior probability $p(D|B,\beta)$ is plotted in Figure 4.1 (with varying values for the estimate of the slant, keeping a fixed value for the uncertainty) and in Figure 4.4 (this time varying the uncertainty).

Consider how the orientation of the surface should effect our expectation of the distance for a given brightness: if the surface is facing directly $(\beta = 0)$ then we should expect it to be further than a surface of the same apparent brightness that is tilted away, as is confirmed in Figure 4.1 If we vary the amount of uncertainty σ_β in the estimate of the slant, as in Figure 4.4, we find some interesting changes to the shape of the distribution. As the uncertainty increases

($\sigma_\beta = 10$) then the shape begins to resemble that of Figure 3.3, in which we assume the orientation of the surface to be equally likely in any direction. As the certainty increases ($\sigma_\beta = 0.01$) then the shape begins to look like the CDF of a uniform distribution, which is just what would be expected if the slant β were given as a constant and we had simply marginalized out the uniform random variable ρ.

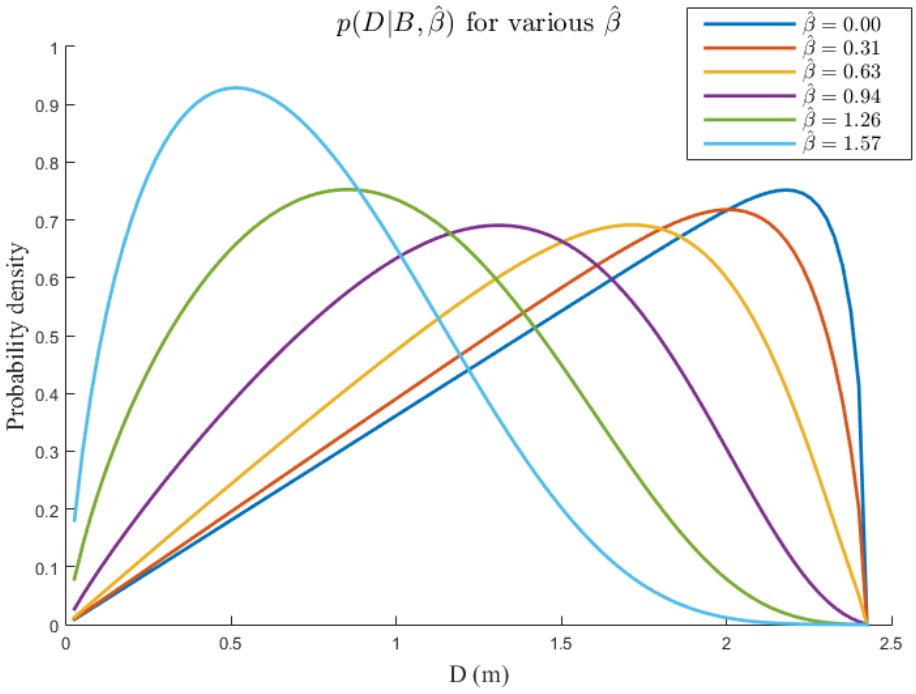


Figure 4.1 Conditional probability density, varying Slant for the distribution $p(D|B,\beta)$ of distance given measured brightness and estimated surface slant. Using Brightness B = 0.2, and $\sigma_\beta = 0.3$.

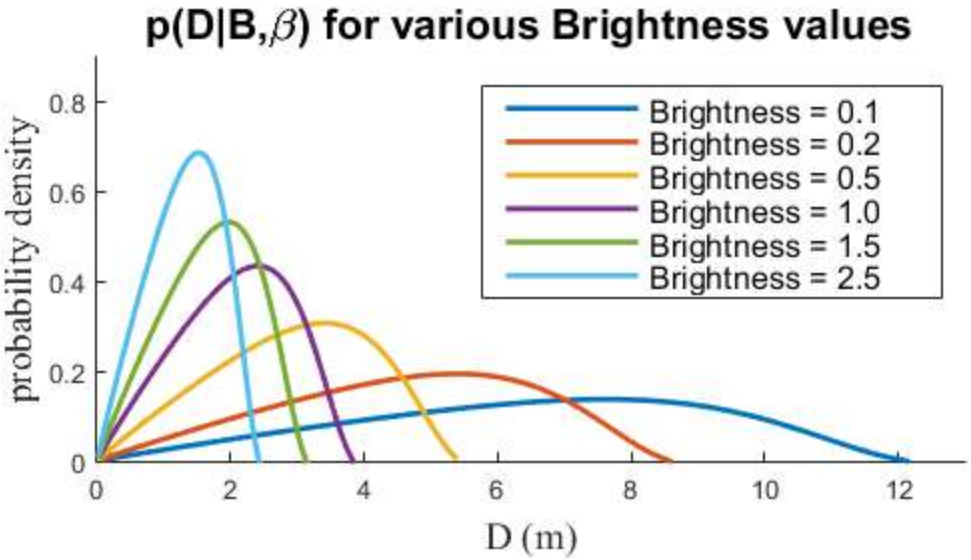


Figure 4.2 Conditional probability density, varying Brightness for distance given measured brightness and estimated surface slant.

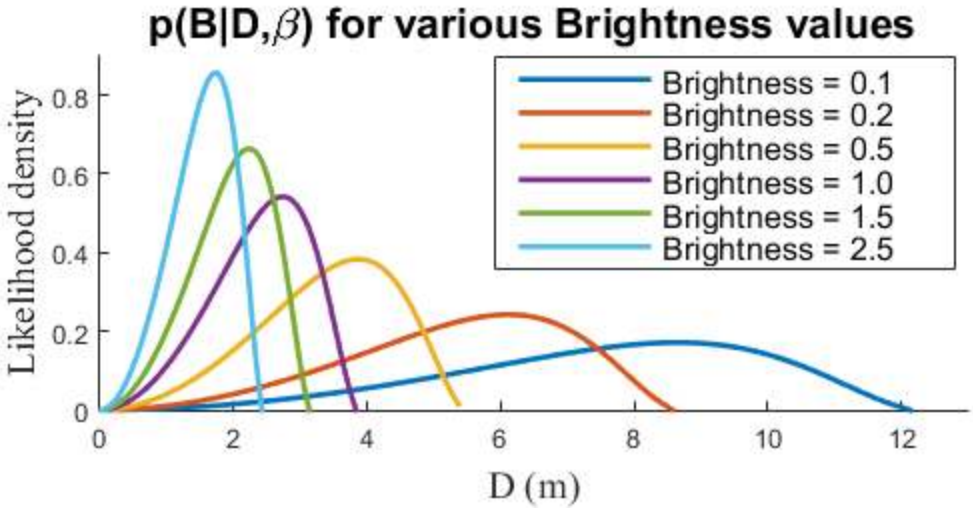


Figure 4.3 Conditional likelihood distribution, varying Brightness for brightness given distance and estimated surface slant. Normalized to integrate to 1 and plotted using $\beta = \pi/4$ and $\sigma_\beta = .3$.

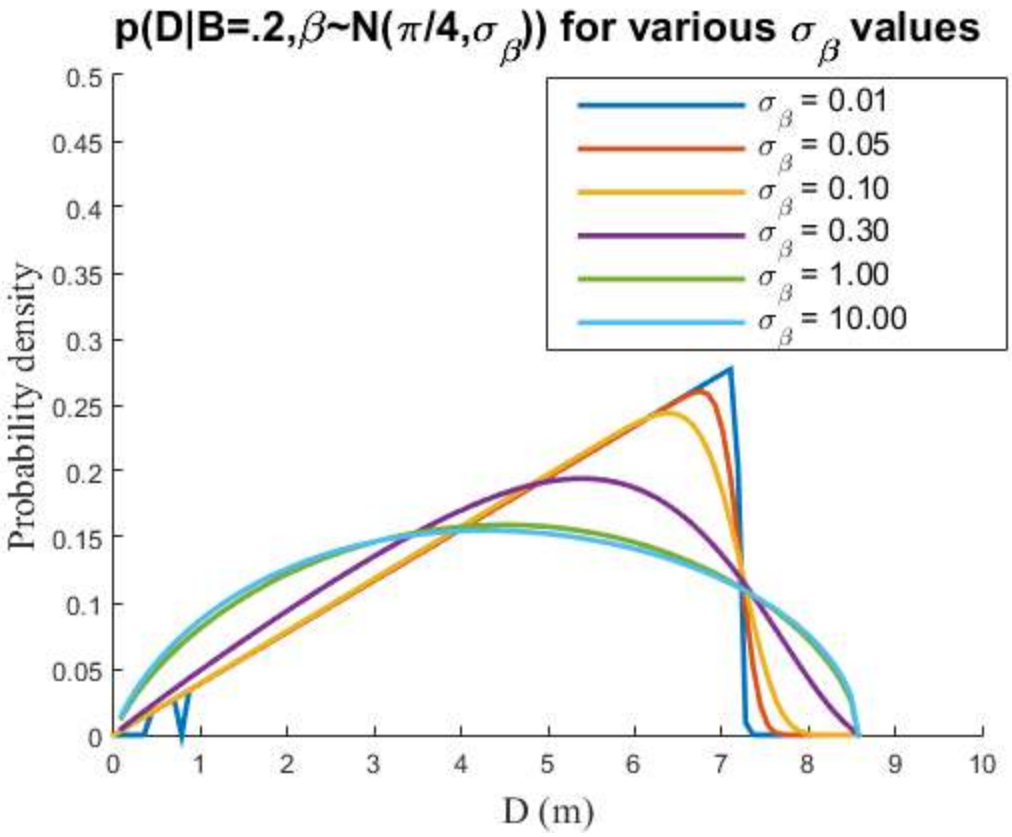


Figure 4.4 Conditional probability density, varying Slant uncertainty σ_β for distance given measured brightness and estimated surface slant.

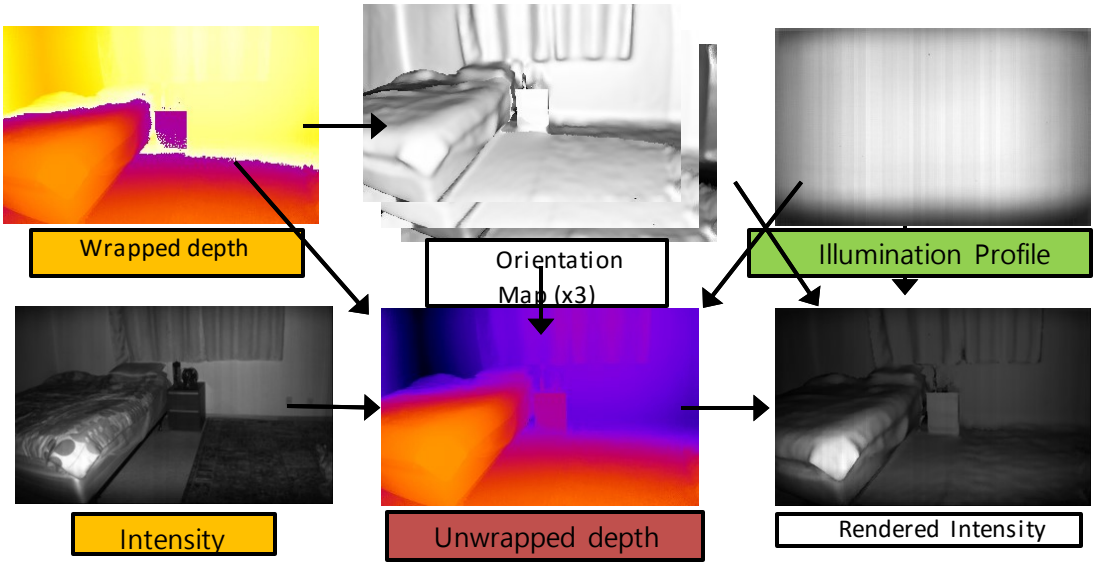


Figure 4.5. Visualization of the intensity model.. Gold labels represent the observations: the wrapped phase and the intensity image. Green is measured a priori: the illumination profile is calibrated by making measurements of intensity against a surface with known albedo at known distances; embedded are electronic system artifacts (visible as stripes). White are rendered images: the surface normal is estimated from the wrapped phase, and this is done for each wrap value. Rendered here is the $\cos(\beta)$, the slant with respect to the camera. The rendered intensity is the image that would be predicted by my model. The simplifying assumption of uniform albedo is a chief cause in the differences between the predicted intensity image and the observed image: note the missing texture form the bed and carpet, and how the dark wooden dresser appears light. Orange is the final phase predicted by the algorithm

4.2.2. Enforcing Spatial Coherence

In the previous section, I derived an expression for the marginal probability of the wrap number K_p at each pixel. I now discuss how this knowledge can be used in a framework that also leverages spatial coherence priors.

Spatial coherence is traditionally modeled by means of the Markov Random Field (MRF) formalism. MRF and related techniques attempt to find an image labeling that maximizes the joint posterior probability of label assignment given the observables. In practice, this translates to defining a cost function that is the sum of *data cost*, that penalizes label assignment inconsistent with the observation, and *discontinuity cost*, that penalizes changes of label assignment across nearby pixels [41]. Unfortunately, closed form expressions for these cost functions are available only for simple 1-D cases, whereas cost minimization for generic 2-D images requires computationally expensive operations such as simulated annealing, belief propagation, or graph cut [13].

In this contribution, I define a different cost function, one that, while enforcing spatial coherence, can be minimized very efficiently. The approach is inspired by the Non-Local Cost Aggregation (NLCA) algorithm, originally proposed by Yang for stereo matching [119]. In order to make this contribution self-contained, I begin by providing a short introduction to NLCA, then show how NLCA can be applied to this problem.

1. The NLCA Algorithm

Let us first review the notation from the previous section, and introduce the symbols for the new terms. O_p represents the observation at pixel p. (In this case, O_p comprises ϕ_p, B_p, and β_p.). $C_p(K)$ is the *marginal data cost* of assigning label K to pixel p based on the observation O_p. For example, in Yang's paper

regarding stereo disparity, the marginal data cost is defined by $C_p(K) = \left|I_p^l - I_{p-K}^r\right|$; it represents the "matching cost" between the left and the right image (I^l, I^r) if the disparity value K is assigned to pixel p in the left image.

The *aggregated cost* $C_p^A(K)$ is defined as follows:

$$C_p^A(K) = \sum_q C_q(K) S_{p,q} \tag{4-6}$$

where the *similarity function* $S_{p,q}$ represents the belief, based on the observations, that pixels p and q should be assigned the same label. A small value of $C_p^A(k)$ (the aggregated cost at p for a certain label K) signifies that the set of *supporting pixels* (pixels that are *similar* to p) "agree" on this label. The NLCA algorithm simply assigns to each pixel the minimizer of its associated aggregated cost.

Similarity functions have been used extensively in computer vision. For example, the bilateral filter [107] uses a similarity function to define adaptive filter kernels, where two pixels are "similar" if they are at close distance *and* their colors are close in color space. Specifically, the bilateral filter defines the following:

$$S_{p,q} = \exp\left(-d\left(O_p, O_q\right)^2 / \sigma_O^2\right) \cdot \exp(-\|p - q\|^2 / \sigma_D^2) \tag{4-7}$$

where $d(O_p, O_q)$ is a suitable distance between observables, and σ_O, σ_D are balancing constants[4]. The normalized cut algorithm [101] defines a similar metric for the edges of the graph to be clustered.

The NLCA algorithm defines the similarity function $S_{p,q}$ in a way that preserves the character of (4-7), while allowing for very fast computation. The algorithm first defines a planar (4- or 8-connected) graph on the image pixel grid, with edge cost between two neighboring pixels (r, s) equal to $d(O_r, O_s)$. Then, the minimum spanning tree of this graph is computed. The spanning tree, coupled with the edge costs, defines a *tree metric* on the image pixels, induced by the distance in the tree between the nodes representing any two pixels (where the tree distance is equal to the sum of the edge costs in the unique path between the two nodes). Let us define the tree distance between p and q as $d^T(p, q)$. One easily sees that two pixels have a small tree distance only if they have similar appearance *and* they are close in the tree (and thus close in the image grid). The NLCA algorithm defines the similarity function $S_{p,q}$ in (4-6) simply as:

$$S_{p,q} = \exp(-d^T(p,q)/\sigma) \tag{4-8}$$

A very useful characteristic of this similarity function is that, if pixel r is in the path in the tree between p and q, then

$$S_{p,q} = S_{p,r} \cdot S_{r,q} \tag{4-9}$$

[4] Note that the similarity function $S_{pq}(K)$ must be normalized for use as a kernel in the bilateral filter. Normalization is not necessary for NLCA.

Yang uses this property to cleverly derive an extremely efficient algorithm for minimization of the aggregated cost $C_p^A(K)$ at each pixel. The computational cost of producing a labeling (in addition to the computation of the minimum spanning tree) is of 2 additions/subtractions and 3 multiplications for each label K in the set of labels. The maximum number of wraps in a given scene is a function of the maximum range of that scene and the modulation frequency of the illumination source. Though in practice, the ability to measure phase shift from the returning signal is dependent on a strong enough signal, so the illumination power should be chosen to sufficiently light the desired range. In my experiments, I use a maximum wrap value of 3, in other words, 4 times the unambiguous range.

2. Phase Unwrapping Via NLCA

The NLCA algorithm is a generic labeling technique that can be easily extended to our wrap number estimation problem. Specifically, the marginal data cost is defined as follows:

$$C_p(K) = -P\left(K_p|\phi_p, B_p, \beta_p\right) \tag{4-10}$$

The cost is used to populate a cost volume, with each slice representing the wrap label. The volume is reweighted by the aggregated cost from (4-6), efficiently computed as described in [119]. To encourage spatial coherence only between coherent areas, I choose a distance function that can reflect similarity of pixels beyond having merely similar phase measurements. The natural extensions involve using the other observable features: intensity B and surface normal N. I

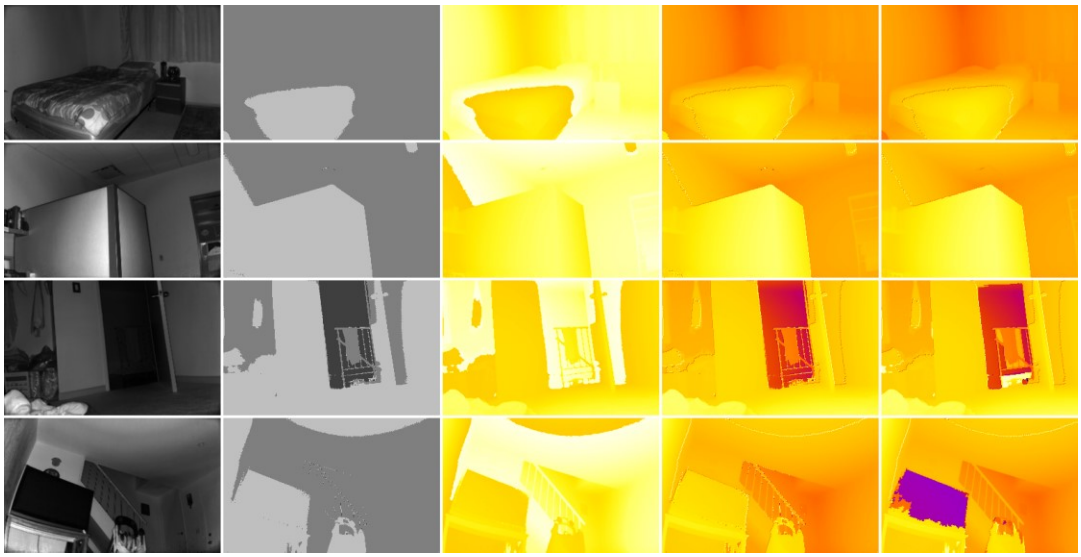
define distance functions for each feature type as: $d_\phi(O_p, O_q) = |\phi_p - \phi_q|/2\pi$, $d_I(O_p, O_q) = |B_p - B_q|/\max(I)$, and $d_N(O_p, O_q) = 1 - \text{dot}(N_p, N_q)$, where $\max(I)$ is the maximum intensity value over the image, and $\text{dot}(\cdot,\cdot)$ is the dot product of normalized vectors. The functions are defined to each be within the interval of [0,1] and can be assembled into multi-feature distance functions, as described in 4.3.4.

4.3. Experiments

For these experiments the same set of 45 indoor scenes from 3.3 (consisting of wrapped phase $\boldsymbol{\phi}$ and intensity $\boldsymbol{B}$) were used to test the proposed algorithm. Additionally, surface normal offset angles $\boldsymbol{\beta}$ were computed from the wrapped phase values. Surface normals were estimated using the Point Cloud Library [95]. These indoor scenes consisted of a home, an office setting, and a computer lab. The Canesta XR670 3D Camera was used to capture data at a resolution of 320×200 pixels, running at two frequencies: 51.4MHz and 68.6MHz. Ground truth was determined by using the dual-frequency approach of [2] to combine these pairs of captures. The phase unwrapping algorithm was tested using data from each of the two frequencies individually.

Tests were run under the assumption that the maximum range of the scene is known ahead of time, thus limiting the number of phase wraps the algorithm can expect to encounter. The difficulty of the problem is compounded by a larger

number of wraps, and therefore the test scenes were classified by difficulty in the maximum value of wrap label K, either 1 (14 cases), 2 (45 cases), or 3 (31 cases).



(a) (b) (c) (d) (e)

Figure 4.6. Selected scenes demonstrating performance. Column (a) shows the intensity image from the active illumination. Column (b) is the ground truth number of phase wraps, from 0 to 2 wraps with darker greys being more wraps. Columns (d) and (e) show phase, increasing from hot white to cool purple, while (c) shows the measured phase (wrapped at 2π). Column (d) is representation of the unwrapped phase or depth, as measured by a multifrequency ToF camera. The final column shows the reconstructed phase from my method. The top two rows taken from the *easy* group and the bottom two from the *medium*, chosen to highlight some specific difficulties. In the 3rd row, column (b) contains a thin railing challenging the spatial coherence assumption. While the very low albedo of the TV screen in the 4th row may lead the brightness model to assume it is distant.

4.3.1. Comparison to Previous Methods

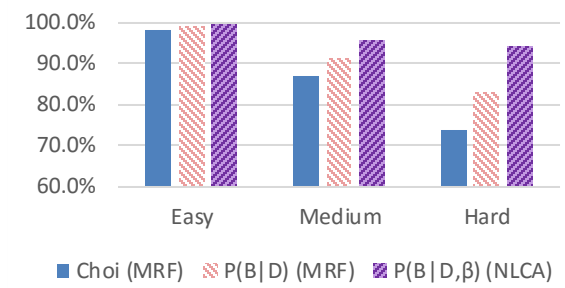


Figure 4.7. Comparison of the proposed method against prior methods of Choi and MRF without normal estimates. Broken down into easy, medium, and hard cases, we can see that in the easy cases, all method perform nearly perfectly, but the advantages of the proposed method stand out in harder cases.

I tested the proposed solution in full, intensity model complete with surface normal estimation, and spatial coherence enforced by nonlocal cost aggregation, using a distance metric utilizing the wrapped phase ϕ and surface normal N (see 4.3.4 for details). I compared the results of my algorithm to that of Choi [21] and my earlier approach [23]. Over the entire data set, I observed an average of 95.9% pixels labeled correctly from this method, compared to 84.3% for [21] and 89.7% for the method of the previous chapter. However, the performance was very much dependent on the difficulty. In the easiest cases of a single phase wrap, each method performs nearly perfectly, though the proposed method slightly edges out with 99.8% over 99.4%, from the method described in the previous chapter. In the hardest of cases I found that the proposed method still makes huge leaps, achieving rates of up to 94.3% of pixels labeled correctly, more than 11% above the previous method.

4.3.2. Intensity Model Comparison

The intensity model, as described in 4.2.1 relies on three sets of measurements: the brightness, wrapped phase, and surface slant. While the intensity B and wrapped phase ϕ are observed directly, the surface slant β, being a function of the surface normal N, must be estimated from the measured phase. The normal is estimated for every pixel at each wrapping label. That is, for each pixel, a point is projected along the pixel's directional ray at a distance of $\frac{c(\phi_p+2\pi K)}{4\pi f_m}$ for each value of K. The set of points is passed into the Point Cloud Library's surface normal estimation function, with a parameter specifying the number of nearest points to use in estimating the normal, and the method of estimation. The Point Cloud Library offers several methods of normal estimation, with varying degrees of accuracy and efficiency. The most efficient methods use integral images to avoid a lot of redundant calculation. There are three methods which use the integral image trick: one that estimates the covariance of a points local neighborhood using 9 integral images, one which creates a smoothed version of the neighborhood with 6 integral images to estimate the 3D gradient (two per dimension), and the quickest method which simply estimates the average depth change with a single integral image (depth only). The estimation using a single integral image proved completely unreliable, but the covariance and 3D gradient methods showed reasonable performance. Additionally, PCL offers more robust methods that use nearest neighbors in 3D space rather than relying on all

points in the neighborhood defined by the grid. This method is much slower, taking up to several seconds per frame, depending on the size of the neighborhood.

In visualizing the surface normal, this parameter makes a clear difference in the consistency in the normal from one pixel to the next. And while the choice in this parameter effects the labeling accuracy by less than 3%, it makes the difference in causing this wrap labeling to perform better or worse than using the intensity model without a surface normal estimation (though in conjunction with spatial coherence, even poor normal estimates outperform the model without normals). I found that choosing a neighborhood of 49 points using the nearest neighbor setting produced good results, and the reported results use that parameter setting.

The first set of tests looked exclusively at the ability of the intensity models, using the conditional likelihood formulation based on $p(B|D,\rho,\beta)$ and $p(B|D,\rho)$, to estimate the correct wrapping label. Simply, the likelihood of each wrapping

	P(K|B,ϕ)	P(K|B,ϕ,β)
Full set	81.4%	82.7%
Easy	88.7%	90.0%
Medium	83.6%	84.8%
Hard	75.0%	76.4%

Table 1. A comparison of the illumination models with and without normal estimation. No spatial support was used; simply the most likely wrapping number from the brightness model was chosen per pixel.

	MRF	NLCA
Full set	89.7%	91.8%
Easy	**99.4%**	96.1%
Medium	91.2%	93.7%
Hard	83.2%	87.2%

Table 2:. A comparison between the MRF and NLCA methods of enforcing spatial coherence, using the data term as defined in 3.2.2.

	P(D|B,ϕ)	P(B|D,ϕ)	P(D|B,ϕ,β)	P(B|D,ϕ,β)
Full set	91.50%	92.08%	92.25%	95.92%
Easy	99.10%	97.13%	98.42%	99.86%
Medium	95.31%	95.43%	96.76%	97.64%
Hard	88.66%	90.21%	90.34%	94.33%

Table 3. Comparison of distribution terms in the intensity model. This table compares the performance of the four basic versions of the data term's objective function, either as a likelihood or posterior probability. Spatial coherence was enforced by NLCA using a distance term formed by the $\mathcal{L}_1$-norm in a (B, ϕ)-space.

label was computed either as in (4-3) or (3-8), and the mostly likely option is selected. Surprisingly, the inclusion of an estimate of the surface normal had only a small impact on the ability to choose a wrap label from intensity alone. Compiling all scenes, I found a labeling accuracy of 81.4% for the intensity model without normal, and 82.7% with. Breaking the data up by difficulty showed a similar spread of just over 1%, as shown in Table 2.

The second set of tests was focused on comparing the alternate probabilistic objectives: the conditional likelihood $p(B|D, \rho, \beta)$ or the posterior conditional probability $p(D|B, \rho, \beta)$. Both seems like reasonable approaches, and prior to testing I had no intuition as to which might be more appropriate. I also included the earlier formulations from 3.2.2 that did not include an estimate of the surface model. In all cases, nonlocal cost aggregation was applied using the distance function which included the normal estimation and the wrapped phase, even when the normal was not included in the probabilistic formulation.

4.3.3. Comparison of Spatial Coherence Methods

A common goal in both the proposed cost function with NLCA and that of the MRF is to inherently identify pixels belonging to the same physical surface and promote that the depth change between neighboring pixels on the same surface is gradual and smooth. Belief propagation enforces this strictly through adjacent pixels, applying a penalty for mismatched labels across a pixel pair. Spatial coherence over larger areas emerges as support or "belief" in a particular labeling is spread through the image over successive iterations of message passing. Pixel support in non-local cost aggregation comes somewhat similarly by way of a series of adjacent pixels. However, unlike the cyclic lattice graph of a Markov random field, support between any two pixels comes via a unique path in the minimum spanning tree. And instead of support disseminating over multiple iterations, it is aggregated wholly in two passes over the entire tree.

Spatial coherence is enforced within the cost function of the MRF by way of a discontinuity cost. This assigned a penalty, for a pair of adjacent pixels, based on the phase jump implied by a choice of wrap labels, defined as $\mathcal{N}\left(\theta_q - \theta_p; 0, \sigma_\theta^2\right)$, (truncated at some maximum value) where $N(\cdot; \mu, \sigma^2)$ represents the normal probability density function, σ_θ^2 is the variance of differences of adjacent phase measurements. Where the difference in measured phase $\phi_q - \phi_p$ is small, there will be a large penalty for mismatched labels K (otherwise, there will be a

large penalty regardless, so the choice of labeling is not significant between this pair).

The similarity function defined in 4.2.2.1 plays the same role as the discontinuity cost, though in the paradigm of NLCA, instead of explicitly assigning a high cost for labels which induce discontinuities, a pixel p receives support from pixel q for a label K that is proportional to the similarity $S_{p,q}$ between p and q. In this way, pixels which are not similar have little influence on each other's labels.

The spatial coherence methods can be compared directly by using the same values from the 'data term' of the Markov random field to populate the cost volume described in 4.2.2. To make this comparison as similar as possible, I defined the distance function for NLCA simply as $d_\phi(O_p, O_q) = |\phi_p - \phi_q|$.

Looking at the tests as a whole, there is a small but significant advantage to the NLCA approach, with 91.8% of pixels labeled correctly, over 89.7%. However, when we separate the cases by difficulty we find the advantage is not so clear cut. In the easiest cases, involving only 1 phase wrap, the MRF approach performs excellently, mislabeling only 0.6% of the pixels, while NLCA misses almost 4%. However, as the difficulty increases, we find NLCA demonstrates an advantage, seen in table 1.

4.3.4. Exploring the Distance Function of Minimum Spanning Trees

A distinct contribution of this work was in defining a similarity function for spatial coherence that went beyond solely the measured phase. In much of the

previous work, including my own, the smoothness term imposes a penalty whose magnitude is based on how similar the unwrapped phase might be if the optimal labeling were *chosen for that pair*. That is, if the difference in wrapped phase between a pair of pixels is, for example, just under 2π, then a large penalty will be imposed if the assigned wrapping labels don't result in a difference of just under 0; while if the wrapped phase were near π, there will be little penalty for any assignment, as there is likely a discontinuity. This is reasonable effective in most cases, however, cases in which the measurement is noisy the penalty might be too low, or if there is a true discontinuity close to a multiple of 2π the penalty might be too high. Using additional cues such as the image intensity or an estimate of the surface normal may provide extra guidance when the phase alone is not enough.

With the approach of NLCA, each pixel will receive some support from every other pixel, by way of the unique path along the minimum spanning tree, and typically, as long as each node along the path is similar to the next the support will remain strong. How much the strength is decreased is determined by the choice of σ. Examples of the difference in pixel support are visualized in Figure 4.8 and Figure 4.9. Notice that when using the intensity, the support stays mostly limited to pixels on the carpet, as they are of a similar intensity. However, using the normal estimate N, support is spread over the floor, as it all shares the same orientation.

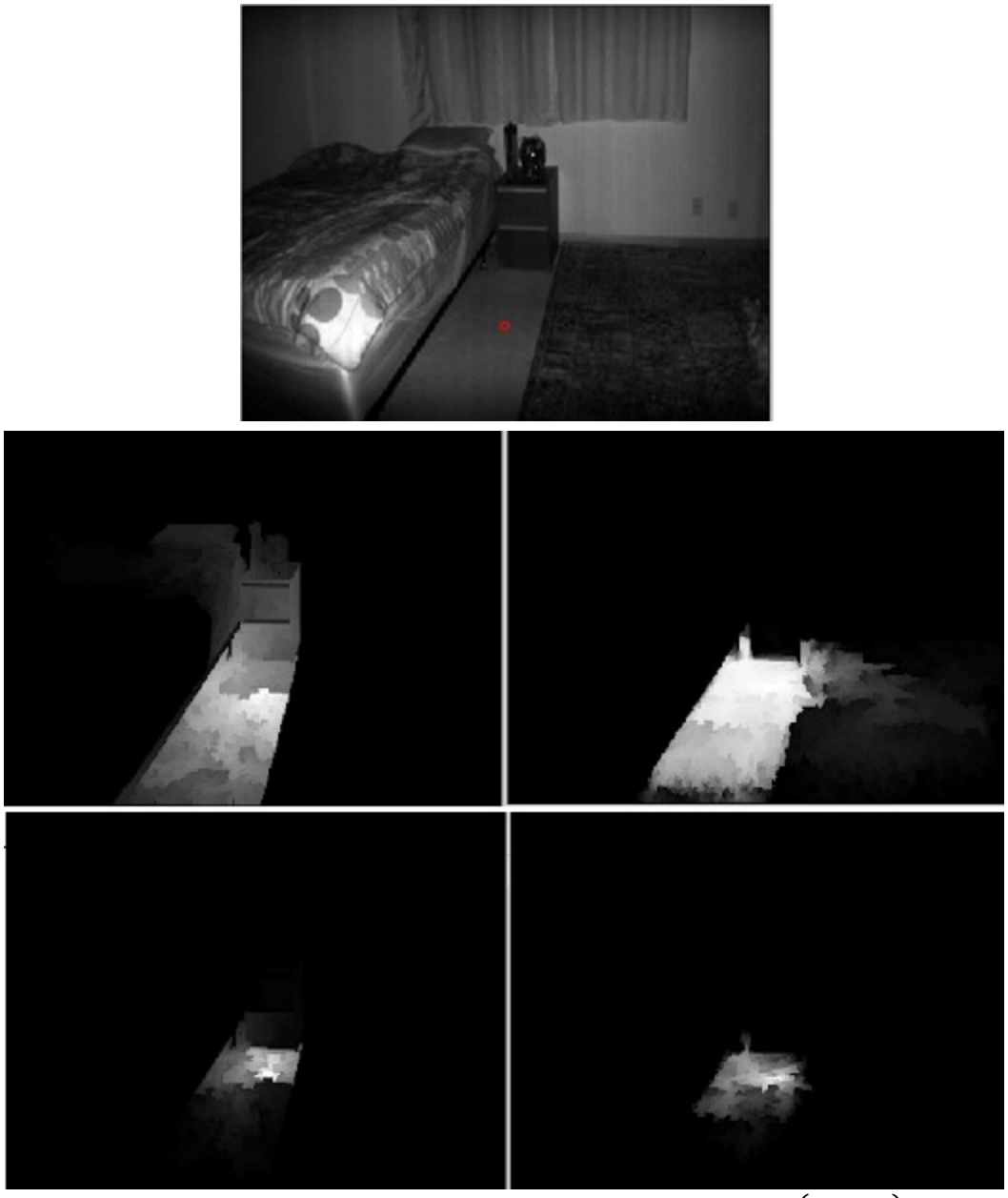
	B	ϕ	N	B,ϕ	B,N	ϕ,N	B,ϕ,N
Full set	88.50%	94.74%	85.46%	95.44%	89.50%	**95.92%**	95.90%
Easy	94.10%	99.14%	92.58%	99.71%	94.89%	**99.75%**	99.68%
Medium	88.18%	94.58%	84.49%	95.30%	89.14%	**95.81%**	95.80%
Hard	86.44%	92.99%	83.65%	93.71%	87.61%	94.33%	**94.35%**

Table 4. Comparison of distance terms. This table compares the performance of the various combinations of distance measurements use to form the similarity term of the NLCA algorithm. For terms which used 2 or 3 distance measures, they were combined using the $\mathcal{L}_1$-norm. In the table headings, B and ϕ represent the brightness and phase differences, respectively, while N is 1 minus the cosine of the angle between normals.

I experimented with these different distance functions by themselves, and combined with each other in a number of ways, such as the maximum, l1- and l2-norms: $\max(\lambda_B d_B, \lambda_\phi d_\phi, \lambda_N d_N)$, $\left\|\lambda_B d_B, \lambda_\phi d_\phi, \lambda_N d_N\right\|_1$, and $\left\|\lambda_B d_B, \lambda_\phi d_\phi, \lambda_N d_N\right\|_2$, with weighting coefficients λ where $\sum_i \lambda_i = 1$ (and to shorten the notation, the observations O_p are not included). I tested a handful of distance metrics composed of combinations of observed phase ϕ, intensity B, and estimated normal N. For pairs, I used the ratios: 1:9, 3:7, and 1:1. For triplets I tried: 1:1:1, 1:1:2, and 1:1:8 (in all permutations). For image intensity B, divided by the calibration constant L to normalize with respect to illumination irregularities.

I found that using the intensity or surface normal alone, or even in combination with each other, as a distance metric produced quite poor results. This makes sense as it is important that the support for labels is not shared across phase wrap boundaries, in which the difference in ϕ will be quite high. However, using B or N jointly with ϕ produces superior results, as seen Table 2. The best

performance was produced with the distance function defined as $d_{\phi,N}(O_p, O_q) = \lambda_\phi |\phi_p - \phi_q| / 2\pi + \lambda_N |1 - \text{dot}(N_p, N_q)|$ with $\alpha_\phi = .7, \alpha_N = .3$.



$d = |B_p - B_q|$ $\mathrm{d} = 1 - \mathrm{dot}(N_p, N_q)$

Figure 4.8. Visualization of NLCA support for one pixelmarked with a red circle. The magnitude of support corresponds to intensity. The top image shows the intensity of a bedroom scene. The left column shows the support provided to a single pixel using the absolute difference of intensity as the distance metric, while the right shows the support using the 1-cos(β), the angle between the estimated surface normal. The upper pair of images uses a high value for σ, which allows the support to come from large regions, while the σ value was set low for the bottom images, restricting support to a more local region.

$\mathrm{d} = |B_p - B_q|$ $d = 1 - \mathrm{dot}(N_p, N_q)$

Figure 4.9. Additional visualization of NLCA supportof the pixel circled in red. The top image shows the intensity of a bedroom scene. The left column shows the support provided to a single pixel using the absolute difference of intensity as the distance metric, while the right shows the support using the 1-cos(β), the angle between the estimated surface normal. The upper pair of images uses a high value for σ, which allows the support to come from large regions, while the σ value was set low for the bottom images, restricting support to a more local region.

4.3.5. Algorithm efficiency

It is difficult to make a direct comparison of running times to the previous method using MRF, as that approach was implemented only in Matlab using a less than optimal message passing schedule (simultaneous message passing rather than sequential (e.g. left, right, up, down), while the proposed algorithm was implemented in C++ with an aim to optimize for speed. However, I can report that the observed running time in the fastest mode (about 0.3 seconds per frame) is more than 2 orders of magnitude faster than the time reported in 3.3 (about 175 seconds per frame). The best results, as reported, were found when using the robust normal estimation of the Point Cloud Library (more than 2 seconds per depth image), though a reasonable compromise of performance and speed was found using the covariance method. The running time can be broken down into the various steps of the algorithm: surface normal estimation (as little as .25 seconds using the integral image trick), cost volume construction (.004 seconds, using a look-up table as a replacement for numerical integration on the fly), MST construction .02 seconds), NLCA support computation (.004 seconds).

Foreground Segmentation Using Range Cues

Background substitution is a regularly used effect in TV and video production, both professionally and with the at-home enthusiast. It's been an indispensable tool for the weatherman (via the blue-screen), and more recently becoming a popular feature in teleconferencing and internet chat.

The basic problem of background substitution is the segmentation of the foreground—those portions of interest from the original scene which we wish to keep—from the background. This problem is commonly worked out with the use of an alpha matte, which dictates the proportion of each displayed pixel that will be foreground and that which will be from the replacement background by assigning a value between 0 and 1 to each pixel. Typically, most pixels are either a 1 (all foreground) or 0 (all background), while the pixels at the borders of the foreground will have a value in between, allowing for a natural looking blending along the edges.

In television and film production, by far the most common technique for alpha matting is by way of a blue-screen, in which the action is filmed in front of a solid color (generally bright blue or green) which is easily identified and replaced. While this technique is both simple and effective, it does require a specially designed studio set and prohibits the blue or green hue from being used in the

foreground. Similar techniques can allow for an arbitrarily colored background provided an image of the scene devoid of foreground. More sophisticated methods can even allow for unseen and potentially nonstatic backgrounds, however these 'natural matting' techniques are not performed in real time.

In this section, I present a method of real-time background substitution based primarily on depth, for use with a time-of-flight depth sensor and paired color video camera, which can be performed against arbitrarily colored and non-static backgrounds, as demonstrated in Figure 5.1. It requires only a depth thresholding plane, defining a distance from the camera plane in which objects are accepted as foreground. Given this dividing plane, along with a depth image and corresponding color image, a trimap is automatically generated, and using a cross bilateral filter [66], a complete alpha-matte is created for the frame.

5.1. Motivation & Related Work

The problem of layer segmentation, background subtraction and/or substitution, or alpha- or natural-matting has been addressed by different fields with different methods, and of course for different purposes. Yet the underlying problem is very much related.

Figure 5.1. Background Substitution

5.1.1. Background Modeling

Background subtraction is often a preprocessing step for many computer vision applications. In an image or video, it is not typically the entire scene that is of interest, but rather a particular object or collection of objects that is of interest, which can be termed the foreground (and the rest called the 'background'). So whether the task is detection, tracking, classification, or something as high level as gesture interpretation or other event analysis, it can be useful to discard regions of the scene which are not of interest. As long as the required computation per pixel to remove background is less than the subsequent higher level task, it is reasonable to perform subtraction at each frame. Indeed, for problems such as simple tracking, the key task to be performed is to distinguish the foreground from the background, and thus this background subtraction relieves much of the

work which would be required for detection methods, such as template matching, if the entire frame needed to be searched. Some rather simple methods of background modeling, such as single- [113] or mixture-of-Gaussian [103] models, are easy to implement and quite efficient in both memory and computation and such are regularly used as a first step for many tasks.

Background subtraction is often a key step for intelligent surveillance and video security [36] [48] and the problem of gesture interpretation for human-computer-interaction [113]. It is also widely used in traffic detection and monitoring [64] [8] [26] [71] [61], which has potential for tasks like accident prevention or managing traffic signals for better flow.

Methods for background modeling can be categorized in a number of ways: parametric or non-parametric, pixel-level or global-level (whether pixels are modeled independently or in relation to each other), and predictive or non-predictive (whether the model is time dependent) to name a few. They may use selective updating, or they may update every pixel each frame; in selective updating, the method of determining which pixels are background may be arbitrarily complex. In situations where the background is expected to remain static, the model may be constructed entirely offline.

5.1.2. Pixel-level Background Modeling

One efficient yet effective framework for background modeling is to deal with each pixel independently. This can easily help minimize the complications

of interdependencies between pixels and relieves the need on any assumptions on the size or shape of objects or the texture of background. It also allows the model to be very locally adaptable, which can be very important in a dynamic scene. The typical procedure estimates one or a few expected values for each pixel, with some amount of leeway, so for an incoming frame, if the pixel is within a threshold of an expected value it will be classified as background.

For scenes in which the background is expected to stay relatively stable, an approach as simple as the temporal median per pixel is actually quite a reliable model [70] [26]. Even in conditions with a gradual change in overall illumination, this model will adapt itself. One disadvantage is that this method requires a buffer of frames, and its robustness to outliers (foreground pixels which remain stationary) will be dependent upon this. Still, frames may be sparsely subsampled while still retaining a degree of adaptability. Another drawback for this method is that the median value itself doesn't give much of an indication of what sort of leeway for incoming pixels to be classified as background.

A similar minded approach is to model each pixel as a Gaussian distribution [113]. While this approach could also use a buffer of previous frames to estimate the most recent probability density function, a more efficient is to simply maintain a running average, which estimates the mean adding the new pixel value with some weight; a simple approximation follows:

$$\mu_t = \lambda I_t + (1 - \lambda)\mu_{t-1} \tag{5-1}$$

where μ_t is the estimated mean at time t, and λ is the weight designated to the new pixel, deciding how quickly the model will adapt to changes. Unlike the temporal median approach, we can estimate the variance as well to determine what an appropriate threshold might be to accept pixels as background:

$$\sigma^2{}_t = \lambda(I_t - \mu_t)^2 + (1-\lambda)\sigma^2{}_{t-1} \qquad (5\text{-}2)$$

where $\sigma^2{}_t$ is the variance at time t. The threshold can be set as a certain number of standard deviations from the mean, rather than a hard value, so more pixels can be more accurately modeled with respect to their noise. It is important to note the dimensionality of the pixel values. The above formulation is for a one-dimensional signal, namely the intensity, but the same principle works for a multidimensional vector μ with covariance matrix $\mathbf{\Sigma}$. Often in practice, the dimensions are considered independent and $\mathbf{\Sigma}$ is a diagonal matrix $\boldsymbol{I}\sigma$.

The running Gaussian average will adapt well to a changing background appearance. If an object is placed into the scene, the estimated mean will shift toward the new object's appearance if the object remains stationary. This adaptability itself has benefits and pitfalls. Namely, if an object in the foreground remains still for too long, its appearance will fused into the background model, causing a false classification as background, and when it resumes movement, the true background will falsely classified as foreground until the model readjusts. Koller et al. proposed to update each pixel selectively based upon its classification [64], M, a binary value equal to one if the object is classified as background and zero otherwise:

$$\mu_t = M\mu_t + (1 - M)(\lambda I_t + (1 - \lambda)\mu_{t-1}) \quad (5\text{-}3)$$

It should be noted that as originally proposed, the value of M is decided by whether an incoming pixel value fits the model, thus there is an open possibility of an unwanted feedback loop which could prevent the proper updating of the model. This can be diminished in a few ways; firstly, the classification M may be subject to higher level evaluation (as will be discussed, many background subtraction algorithms work in multiple layers or steps), and secondly M need not be strictly binary.

The temporal median filter and running Gaussian average work well for scenes in which the background is stable, however it is not uncommon for a scene to have a regularly changing appearance, as in the case of a beach scene (moving water-line), waving branches, or an opening door. In these cases, the appearance is not well modeled by a single Gaussian. In [44], three Gaussians are used to model the specific hypotheses of road, vehicle and shadow. In [103] and [48] Stauffer and Grimson propose a more general mixture of Gaussian representation, without specific hypotheses, which can represent several modes for a regularly changing pixel. In this setting, each pixel is allowed a set number of modes for its appearance. In this sense, the constructed model is not aiming to capture the structure of the scene itself, but more so of the scene's observed appearance.

In mixture of Gaussian model, there are N modes for each pixel of I, and each mode has a mean μ_n and variance σ^2_n as well as an estimate for the prior probability ξ_n

of seeing a sample from that mode. The probability that an incoming pixel belongs to the background is the sum of the probabilities for each mode:

$$P(X_t) = \sum_{n=1}^{\mathrm{N}} \xi_{\mathrm{n},t-1} \mathcal{N}\left(X_t, \mu_{n,t-1}, \sum_{n,t-1}\right) \tag{5-4}$$

where each parameter is the estimate at time t and $\mathcal{N}(\cdot;\mu,\sigma^2)$ is the Gaussian probability density function.

In a most accurate formulation, the parameters for each mode can be updated by expectation maximization [44]. However, as a more efficient approximation, only the most likely of modes is updated with the new value, and if no existing mode is deemed likely enough (by a set threshold) then the least likely mode (by the ratio ξ_n/σ_n) is discarded and a new mode is created. A full probabilistic framework is described in [100]. The generality of the mixture of Gaussians make the technique robust to a variety of situations, given that no frame buffer is required, and using fast-update method, it is efficient in memory and computation; thus it is no surprise that it is quite popular and the basis of many background subtraction schemes [58] [53].

The mixture of Gaussians approach requires the user to set a fixed number of modes. Realistically, some pixel location will require more to be accurately modeled, and some less. We can avoid this drawback by using the non-parametric approach of estimating the probability density function by a histogram or, more popularly, the kernel density estimation [37] [36]. This technique, like median filter, requires a buffer of previous frames. Each sample x_n in the buffer (chosen

selectively) represents the center of a Gaussian, and the probability density as a function of x is given as:

$$p(x) = \frac{1}{N}\sum_{n=1}^{\mathrm{N}} \mathcal{N}(x - x_n, \sum_n) \qquad (5\text{-}5)$$

In [78] Mittal and Paragois use this approach with a 5-dimensional descriptor for each pixel which includes color and motion estimate from optical flow; additionally, they use a variable bandwidth window for the kernel based on the estimated uncertainty.

5.1.3. Region- and Global-level Background Modeling

While modeling each pixel independently has its advantages, clearly much information is being lost by ignoring the spatial relationship between background pixels. Thus, many background subtraction methods do take advantage of pixel relationships at a local region level and even globally. It is also not uncommon to work at multiple levels, defining a framework for each. The regional support between pixels is especially important when initializing a model in a cluttered scene, where the distinction between foreground and background cannot at all be taken for granted.

In [50] hypotheses for background values are made at each pixel independently, based on consistent values over time period. These hypotheses are then weighted by local information concerning the optical flow around that region for the time period of the hypothesis.

A good example of modeling at different levels is the hierarchical approach presented in [108]: Toyama et al. propose the Wallflower system, which works on three levels. It includes a pixel-level model (considers each pixel independently), a region-level area, which aims to group regions of pixels whose movement is self-consistent so that a foreground objects with homogenous internal appearance won't be absorbed into the background (the aperture problem—though the observed value of a pixel is sourced from different locations on the moving object, they all may have the same intensity value and thus no change is observed), and a frame level, which seeks to spot global changes (such as the turning on of a light) and swap in an entirely different set of models for the scene.

Javed et al. use a similar hierarchical approach to detect changes at three levels [58]. As previously noted, they use a mixture of Gaussians at the pixel level, though along with color, a gradient model is built in parallel, which is relied upon when a global appearance change is detected in the frame. They also apply region-level information to immediately identify previously covered background when a background object is moved (consider a parked car pulling away)—this is typically falsely identified as foreground until it is adapted into the model.

A poplar framework for utilizing spatial information is to represent the spatial (and temporal) connections between pixels as a Markov Random Field [93] [61] [84]. This makes the reasonable assumption that the state of each pixel (its labeling) is dependent only upon its neighboring pixel. In [118], each background is modeled as an image pyramid with different resolutions, and the

pixels are link in the MRF to adjacent pixels spatially, temporally, and between pyramid layers.

Spatial relationships between pixels can be acknowledged at a truly global level through principal component analysis of a series of frames. The so-called eigenbackground approach [82] treats each frame as a column in a matrix, and after standard eigenvector decomposition of its covariance matrix, only the eigenvectors with the greatest corresponding eigenvalues are retained. By projecting a new frame into the eigenspace and then reconstructing it, only the static parts of the scene will be recreated, so a simple differencing will produce the foreground.

An interesting set of techniques rely on a different set of assumptions, which are based on a global consistency framework. They rely on the assumption that the background should typically be smooth, or at least smoother than the background being occluded by a foreground object [1]. They are designed to work in an offline setting and create a static model of the background. In this paradigm, no assumption is made about the movement of foreground, the temporal coherence of the frames, or how frequently the background is visible (as long as the background is visible at each pixel location in at least one frame). In [117], Xu and Huang use loopy belief propagation (and has a similar assumption to most Markov Random Field approaches: that only neighboring pixels influence each other) to enforce smoothness between pixels. From a set of frames, each pixel location is assigned a value from one of the frames such that its appearance is

most similar to the adjacent pixels. In [22], a single unchanging pixel from the scene is chosen as the background seed, and the background model is grown by adding spatio-temporally adjacent pixels which satisfy a smoothness constraint.

5.1.4. Foreground Segmentation

A problem very related to background modeling is that of foreground segmentation. In some settings, the problem is essentially equivalent: precise subtraction of a well modeled background leaves only the foreground, and the problem is solved. But it should be noted that solutions to the foreground segmentation problem do not necessarily even require an explicit background model, and in many proposed solutions a background model is used as only one part of the entire solution.

Segmentation may be useful for tasks such as video encoding, where foreground regions can be encoded distinctly [77], video effects where it is desirable to place the foreground in front of an arbitrary background scene [25], gesture recognition, where the human figure should be isolated from the rest of the scene, and, as probably the most popular and practical application, as a key element in tracking—whether for detection of the tracked object itself or to assist object model updating as in a discriminative tracker.

Segmentation algorithms come in a variety of flavors. Some are designed to segment only background from foreground, while some can distinguish multiple layers. There are segmentation algorithms designed to be applied to

single images, and others video sequences. There are various assumptions that can be made about the scene and the characteristics regarding the foreground and background (such as motion, color, or texture consistency) and these assumptions should drive the choice for algorithm design. Also, there may be various types of prior knowledge about the scene available (such as estimates of the background's appearance or 3-D spatial information as provided by a range sensor), and this information should be exploited as much as possible.

In such as tracking and gesture recognition, segmentation itself need not be perfect to produce perfect results overall task. For tasks where the goal is to produce an image or video, however, pixel ownership is necessarily very precise, often splitting up pixel ownership around borders. This is referred to as alpha-matting, where for the foreground source image, each pixel is assigned an alpha-value between 0 and 1 to declare what proportion of it belongs to the foreground. This creates a smooth transition between foreground and the substitute background that creates more natural appearance. For such precise segmentation of potentially complex scenes, the segmentation algorithm should be seeded with locations known to be foreground or background. A popular approach is to provide a trimap [12], which labels pixels as foreground, background, or unknown. Often the trimap is generated manually, but there are some attempts to automate this task as in [111] and the method presented in chapter 4. Other techniques rely on boundary selection tools or scribble-based region selection [12] [94].

5.1.5. Non-background based segmentation

The coherence of motion, color or texture within regions of the scene can provide the necessary cues to appropriately segment the objects of interest. A sensible method of segmenting foreground without explicitly knowing the background appearance is through the use of optical flow [6] to estimate motion in the scene. The motion estimate for each pixel can serve as a two-dimensional descriptor [78], which can be used by itself or combined with color or intensity descriptors to cluster similar pixels within a spatial region.

Stereo cameras can provide useful cues that can aid in foreground segmentation [65], not unlike the depth cues presented in section 5.1.4. Accuracy of depth estimates from stereo algorithms, though, is conditional on the scene's appearance (specifically texture), and the computation of such likelihoods is not trivial.

Criminsi et al. detail a real-time algorithm to segment foreground in a video using motion, color, and contrast cues [25]. However, they avoid the explicit estimation of each pixel's motion, optical flow, due to its computational expense, and instead use an approximate spatio-temporal derivative which can discriminate between motion and stasis. They employ a Markov Random Field to model pixel relationships, which includes a temporal prior to enforce consistency between frames, a spatial prior to encourage spatially local consistency of labels

(except where contrast is high), a color likelihood based on color distributions of previous labeling, and motion likelihood.

For high resolution single images, there are a family of approaches which rely on guidance from the user, such as a trimap or an approximate path of the boundary. Inelligent Scissors [80] was an early graph-based approach that chooses the lowest cost path between points designated by the user as being on the boundary, and low cost is associated with strong edge features. The Bayes matting approach takes a trimap and determines the alpha-values at boundaries based on color similarities from the labeled regions [96]. A very popular approach introduced as normalized cuts [100], takes a graph theoretic approach by considering pixels in an image as nodes in a graph, where edges are between adjacent pixels. By choosing a metric to measure similarity between adjacent pixels, the edges can be weighted, and the problem reduces to finding the 'cut' in the graph, the partitioning which provides the lowest similarity between the two sets. Many segmentation algorithms have been based on this approach, including graph cut [12], grab cut [94], and grow cut [110].

5.1.6. Background-based foreground segmentation

One genre of foreground segmentation, which is used as part of a tracking system uses a background model with respect to a foreground model in order to perform foreground extraction. The works discussed in section 5.1.1 cover a fair representation of these techniques. Many of these approaches aim to segment the

scene into many regions and explicitly assign each region a layer [122] [114], which is useful for handling the problem of occlusion. The tracking method is referred to as layer tracking.

Sun et al. present an algorithm for segmentation which requires a background as input to perform initial segmentation, and then clean up errors caused by unpredicted clutter using a graph cut based approach, called background cut [104].

5.1.7. Matting and Background Replacement

As mentioned, the use of blue screens, or chroma-keying, is quite prevalent due its simplicity and was first innovated in the late 1930's—originally designed to work directly on film. The technique has carried on to and been enhanced by digital processing [102], which can operate on a per-pixel basis, allowing for such features as smooth blending and partial transparency.

The idea of replacing each constant color pixel is easily expanded to include any color at any pixel, so that rather than requiring the background to be entirely blue, it can take on any appearance, provided that the empty scene is known in advance. This method is employed in commercial products such as Apple's iChat [3]. However, ambiguities can arise whenever the foreground is similar in color to the expended background pixel, which can be more difficult to avoid than a single blue or green hue, given the arbitrary background requirements. Further, even minor changes in the background can potentially

cause artifacts to appear, and a slight bump to the camera can disrupt the entire background model.

The general problem of background subtraction has been of interest to the computer vision community for some time for reasons outside of visual effects; for example, it is quite useful in tracking or to automatically detect unknown objects of interest. These techniques are designed to work in much less constrained circumstances, such as an arbitrarily colored background, or even a slowly changing scene. In [54] the background is modeled as a weighted combination of previous frames and pixels differing by more than a set threshold are labeled foreground. Elgammal et al. use a Gaussian distribution to model pixel values [37] and in [103] Stauffer and Grimson present the popular mixture of Gaussians model. These methods all can be performed easily in real-time and are adaptive, in varying degrees, to small changes in nonstatic backgrounds (such as trees and bushes) and sudden but persistent changes (an object set down or a camera bump). However, updating the background model can often lag and the segmentation will rarely be precise enough for natural blending.

Kolmogorov et al. describe a solution to the real-time background substitution problem using binocular stereo video equipment [65]. Their methods fuse depth-from-stereo information with color/contrast cues to perform segmentation; however, ambiguities in stereo matching do produce occasional artifacts.

5.1.8. Trimap Generation

Typical methods of natural matting first require the approximate location of segment edges, given by a trimap, which segments an image scene into foreground, background, or indeterminate. In [111], Wang et al. present a method to automatically generate the trimap based on depth cues from a TOF sensor. After upsampling the depth map to color image resolution by way of a cross bilateral filter, a depth threshold is applied, and the binary map is eroded and dilated. The differing pixels are the areas of the trimap to segmented using Bayesian or Poisson matting. I similarly use the depth data to generate a trimap, though in my method we estimate the unknown region before applying the bilateral filter.

5.2. The Substitution Method

The presented method of background substitution is designed around using a time-of-flight depth sensor paired with a RGB color video camera, such as the CanestaVision [15]. The cameras are registered to each other such that each depth measurement can be projected onto the RGB plane, and typically the spatial resolution of the depth image is significantly less than that of the color image. Given a dividing plane, the scene can be segmented by only depth into a trimap. The indeterminate areas of the trimap are processed with a cross bilateral filter

to assign an alpha-value to each color pixel along the edges, producing a natural looking blending.

5.2.1. Low Resolution Depth Projection and Segmentation

The depth and color cameras are calibrated and registered such that the relation between the fields of view is known. Still, because the camera centers are different, the correspondence of pixels in each sensor is not fixed, but rather dependent upon depth of objects in the scene. For each incoming pair of frames, each depth measurement is projected onto the corresponding color pixel or pixels (the low resolution depth measurements typically cover multiple color pixels). This can require a fixed, but nontrivial amount of computation time. This time can be reduced; a background depth model can be constructed given an initially empty scene, following the method of [37], and projection is limited to foreground depths and those areas immediately surrounding.

It should be noted that there are inherent issues when the depth and color measurements come from different sources. First, there is no guarantee that in the projection process every color pixel will be assigned a depth value, and in practice this is certainly not the case. Further, due to parallax, depth of some color pixels will not be seen by the TOF sensor, and other color pixel locations will be have two depth readings. In the latter case, the lesser depth value is assigned as the scene geometry dictates that this nearer surface is necessarily what the color camera observes.

After the projection process most pixels have associated depth values, some do not.

5.2.2. Trimap Generation

To separate foreground from background a thresholding plane is defined, either specified by the user or determined by algorithmic means. Pixels are then each assigned probabilities of being foreground based on their depth measurement or lack thereof. The model for assigning probabilities can be arbitrarily complicated, but I use a simple approach of assigning pixels one of three parameterizable values: pixels with depth within threshold are high likelihood of being foreground, outside of threshold are near zero likelihood, and pixels of unknown are assigned a low probability. The likelihood of unknown depths are weighted towards being background for three reasons: (1) parallax causes missing depth measurements for background only (foreground does not suffer this problem) and (2) further measurements are less accurate (inherent in TOF sensors), and are more likely to be projected incorrectly into the color plane, and (3) if the background is too far away, there is simply no depth reading.

After each pixel is assigned a foreground probability based on its depth, an estimate is made of the likelihood that it is foreground, based on the surrounding pixels:

$$P_{FG}(C_i|D_{j\in N_i}) \approx \frac{1}{Z}\sum_{j\in \mathbb{N}_i} w_{ij} P_{FG}(C_j|D_j, D_{thresh}) \qquad (5\text{-}6)$$

where C_i is a color pixel, D_i is the associated depth measurement, $\mathbb{N}_i$ is the neighborhood of the pixel i, w_{ij} is the weight of pixel j with respect to i, and Z is a normalizing factor equal to the sum of all weights. The weights may be chosen based on some measure of similarity or spatial proximity, however, with computational efficiency in mind, each pixel is weighted equally, and the neighborhood consists of a square window surrounding the pixel. In this way, we can use the integral image trick [68] to compute the likelihood estimate for an arbitrary sized window in time linear with the size of the image.

Once each pixel is assigned a likelihood, a trimap is created. The trimap classifies each pixel into one of three categories: definitely foreground if $P_{FG}(C_i)$ is greater than a threshold t_{FG}, definitely background if $P_{FG}(C_i)$ is less than a threshold t_{BG}, or uncertain otherwise. The term *definite* is used rather loosely for our purposes, in that the confidence of segmentation must only meet a defined threshold. However once designated as foreground or background in the trimap, the designation will not change, and the segmentation is absolute, i.e. alpha-values are 1 or 0, and that pixel is not blended. Those pixels falling in the undetermined portion of the trimap are assigned alpha-values through a bilateral filtering process.

An example trimap is demonstrated in Figure 5.2(d), in which the undetermined area is about 3% of the frame. As is desired, the trimap tends to fall along the border between foreground and background without having to explicitly define this border. The thickness of the trimap band is determined by

the size of the neighborhood (larger neighborhoods allow for thicker bands) and the threshold values t_{FG} and t_{BG} (thicker bands as the thresholds are set closer to 1 and 0, respectively). For any pixels within the trimap, their depth values will play no part in the assignment of an alpha-value, which makes this method quite robust to edge artifacts in the time-of-flight measurements. On the other side of that coin, however, is that if color pixels are labeled with an incorrect depth and do not end up as part of the trimap, then they can cause a local mass of similarly colored pixels to be erroneously labeled. One particular case at risk is that of motion artifacts, in which sometimes large blocks of pixels are measured at a consistent but incorrect depth. With the mixed pixel effect, some combinations of distances will result in completely invalid measurements which can be easily identified and discarded, but in other cases will results in a valid but incorrect depth. Such is one case that demonstrates the importance of a short capture time, as championed in the previous chapters.

5.2.3. Cross Bilateral Filter

When assigning each pixel its initial probability of being foreground, its depth is compared to the threshold, and it is assigned an alpha value of 1 or 0, which is recorded into what is called the 'sparse alpha-matte,' as in Figure 5.2(c). These values are based on each pixel alone, rather than the neighborhood (as with the trimap). Pixels without depth measurements do not get an alpha-value (hence

the name sparse—even though most pixels *do* have values) and are not included as support in bilateral filtering.

A cross bilateral filter is then applied to the sparse alpha-matte, using the color image as the guide for the range filter. The bilateral filter was introduced to the computer vision field by Tomasi and Manduchi [107] as a method of smoothing grayscale images; the idea is to preserve edges by taking a weighted average of local pixels and where the weight of each pixel in the filter is determined by its distance from the filtered pixel in both the grid lattice and range space. With cross bilateral filtering [66], the range weight comes from a different feature than the one being filtered over. In this case, the algorithm filters the alpha values, and bases the weights on distance in the grid lattice and the color space.

The refined estimate for the alpha value A_i of each pixel is

$$\mathrm{A}_i = \frac{1}{Z_i} \sum_{j \in \mathbb{N}_p, \exists \alpha_i} \alpha_j f(\|i - j\|) g\left(\left\|I_i - I_j\right\|\right), \tag{5-7}$$

where α_j is the alpha-value from the sparse alpha-matte, f is the spatial filter kernel (in this case, a Gaussian centered at i), g is the range filter kernel (also a Gaussian), I is the color image, $\mathbb{N}_p$ is the neighborhood surrounding i (implemented here as a square window), and Z_i is a normalizing factor, the sum of the product of filter weights defined as

$$Z_i = \sum_{j \in N_i, \exists \alpha_i} f(\|i - j\|) g(\left\|I_i - I_j\right\|) \tag{5-8}$$

The distance between colors is measured as a Euclidean the RGB color space, assuming a 256 value quantization in each channel. The size of the neighborhood window and sigma values (standard deviation) for both Gaussian kernels are parameterizable by the user.

5.3. Experimental Results

The described method has been implemented in C++ and tested on an AMD Athlon 64 X2 Dual Core processor at 2 Ghz, using a CanestaVision camera. The ToF depth sensor has a resolution of 160×120 pixels, and the RGB color camera has a resolution of 640×480 pixels. Phase unwrapping for the depth image is performed using a dual-frequency approach. I was able to run the algorithm as describe at rate of 10 frames per second with the following parameter settings. For the trimap generation, a window size of 13×13, foreground threshold of $t_{FG} = 95\%$ and background of $t_{BG} = 15\%$, assuming measurements below the threshold are surely foreground, those above are surely background, and unmeasured pixels have a 25% chance of being background. As mentioned previously, these settings lead to about 2-3% of the pixels in each frame to require bilateral filtering. For the bilateral filter, I used a window size of 30 pixels, with a spatial sigma of 30 pixels, and a color sigma of 12.

Figure 5.2 demonstrates the overall process, starting from the original pair of color and depth images, the initial segmentation provided by simple

thresholding, the trimap and resulting alpha-matte, followed by the final substitution of a new background.

The method for real-time background substitution guided by a time-of-flight range sensor outlined here places few restrictions on background and foreground and merely requires a user define dividing plane. It has been demonstrated to run adequately in real-time using a non-optimized implementation of the bilateral filter (though designed for only integer operations in the kernel, and exponential is approximated by a lookup table). Still, much of the processing time goes into the bilateral filter. This could time could be greatly reduced by using a quantized approximation of the bilateral filter, as described in [85]. Further subsampling of the spatial or color domain, or a smaller window for filtering could increase performance as well.

In this implementation the user is required to set a threshold value for the dividing plane. However, it seems quite reasonable that simple clustering methods such a k-means or mean shift approach could quickly estimate a reasonable dividing plane based on depth alone. Additionally, heuristic knowledge that the foreground is generally at the center of the field of view, or that the object of interest tends to move more than the background could also guide a completely automated segmentation scheme.

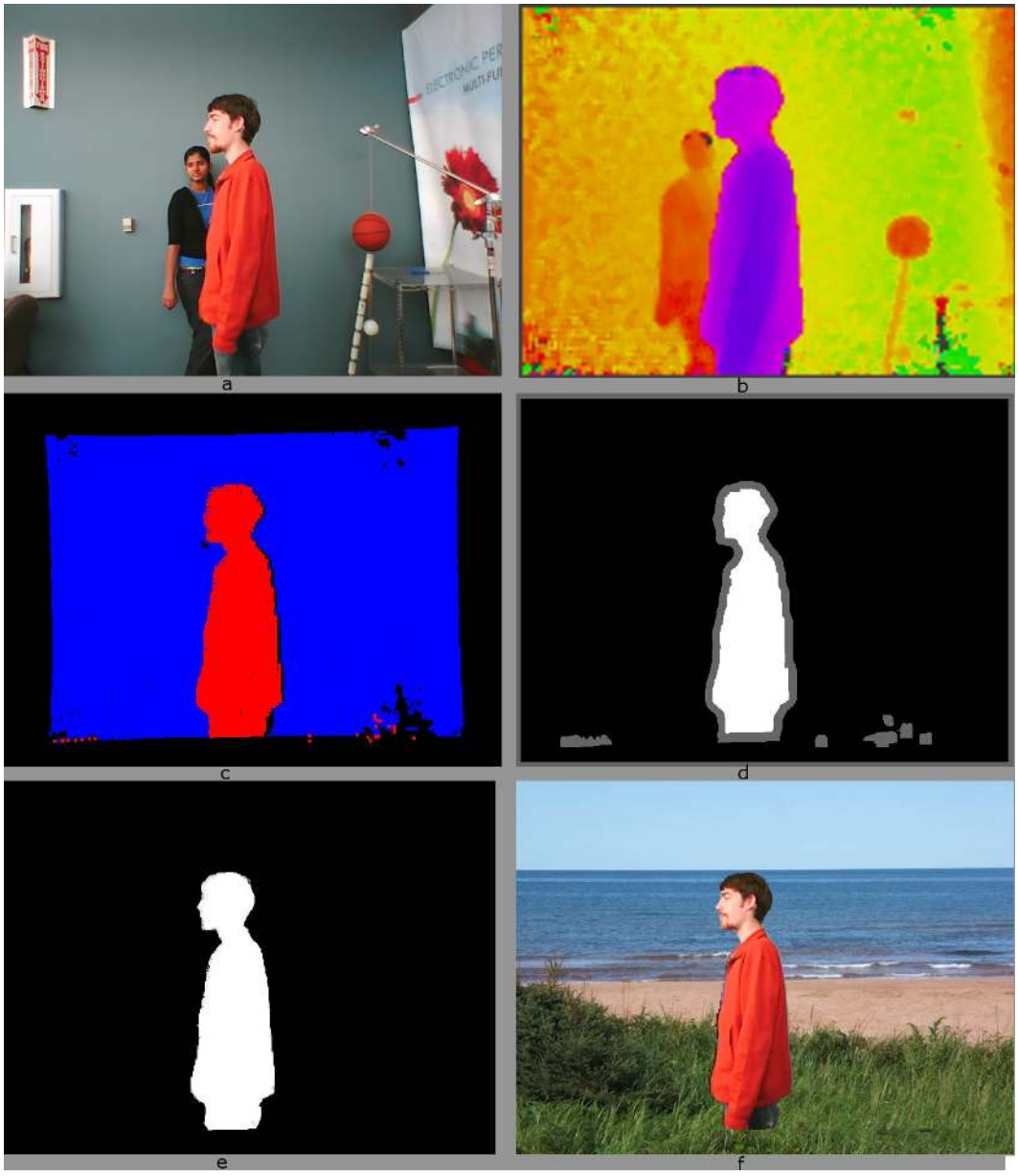


Figure 5.2. Diagram of the alpha-matte process. The original scene from the RGB color frame. (b) The depth image from range sensor, at 16 times normal resolution. (c) Sparse alpha-matte, where background is represented by blue, foreground is red, and black are pixels with no depth value. (d) Trimap, grey is undetermined area. (e) Final alpha matting. (f) Foreground overlayed on beach background using alpha matte.

Appendix A
Derivation of Surface Normal Distribution

This section presents the derivation of an estimate for the distribution of the angle β between the surface normal and the angle of incident of the incoming light. I assume no prior knowledge of surface orientations and grant that all orientations are equally likely. I recognize that this is not a reasonable assumption: the Manhattan world paradigm belies this notion and the wrapped phase measurements, while very noisy, provide a reasonable source of information for a more accurate estimate of the distribution. At present I leave this for future investigation, and move with the hypothesis that simply imposing the natural limits of the surface orientation can improve phase unwrapping performance.

To derive the distribution of β, begin by visualizing the orientation of the surface normal as a uniformly distributed hemisphere with the angle of incidence in the center, as shown in Figure A-1. The camera can only view a surface which

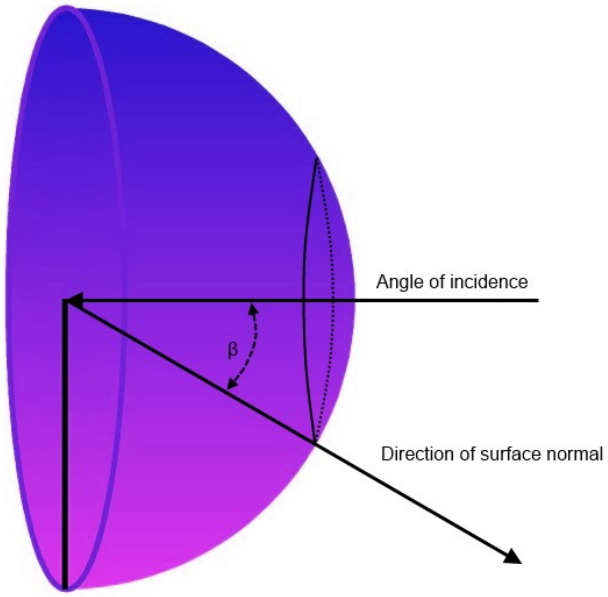


Figure A-1. The distribution of surface normal is represented by the hemisphere, with the angle of incidence head on. An example of the direction of the surface normal is illustrated, with the resulting angle β.

is facing it, limiting the range of β to be from 0 (the surface is directly facing the camera) to $\pi/2$ (the surface is facing perpendicularly). Now keeping in mind the image of a hemisphere as the domain of the direction of the surface normal, consider how the portion of the surface defined by β is represented by a circle whose circumference is proportional to $sin(\beta)$, also illustrated in Figure A-1. At the same time, the visible size of the surface area decreases as the surface normal turns away from the incident angle and is proportional to $cos(\beta)$. Putting these together as a proper distribution, including a constant factor so the probability integrates to 1, we can define the probability distribution function for β as

$$p(\beta) = 2sin(\beta)cos(\beta).$$

Appendix B Derivation of Conditional Likelihood

The conditional likelihood of the intensity, given the wrapped phase measurement, wrap state number, albedo, and surface normal is represented by the Dirac delta function. The Dirac delta function is specifically chosen because it is a function whose integral is equal to 1 if the zero is within the limits of integration: a reasonable choice to represent a probability distribution for when a specific condition is met. The key characteristic of the Dirac delta to note is its composition with a function:

$$\int_{-\infty}^{\infty} f(x)\delta\big(g(x)\big)dx = \sum_{i\in\Re}\frac{f(x_i)}{g'(x_i)} \tag{B-1}$$

where $\Re$ are the roots of $g(x)$. In our case we have the Dirac delta as a function of the surface normal angle β with a single root as defined from equation (3-4)

$$\beta_0 = \cos^{-1}\left(\frac{BD^2}{\rho L}\right) \tag{B-2}$$

keeping in mind that D is related to Φ and K by equations (3-1) and (3-2) and can be used interchangeably:

$$D = \frac{c(\phi + K)}{2f_m}. \tag{B-3}$$

Let $f(\beta) = \sin(\beta)\cos(\beta).$

Let $g(\beta) = B - \frac{L \cdot \rho \cdot \cos(\beta)}{D^2}.$ Then $g'(\beta) = \frac{L \cdot \rho \cdot \sin(\beta)}{D^2}.$

Recall from equation (3-8)

$$\mathcal{L}(D|B) = p(B|D) = 2\int_{\rho=0}^{1}\int_{\beta=0}^{\frac{\pi}{2}} \delta\left(B - \frac{L \cdot \rho \cdot \cos(\beta)}{D^2}\right)\sin(\beta)\cos(\beta)\, d\beta d\rho.$$

Noting the limits of integration for β, the root will be within the bounds of integration when $0 \le \cos^{-1}\left(\frac{BD^2}{\rho L}\right) \le \pi/2$, which implies $\rho \ge \frac{BD^2}{L}$.

Combining eq. (3-8) and (B-1), and applying the new bounds of ρ:

$$\int_{\beta=0}^{\frac{\pi}{2}} \delta\left(B - \frac{L \cdot \rho \cdot \cos(\beta)}{D^2}\right)\sin(\beta)\cos(\beta)\, d\beta = \int_{-\infty}^{\infty} f(\beta)\delta\big(g(x\beta)\big)d\beta = \frac{f(\beta_0)}{|g'(\beta_0)|}.$$

Then

$$\begin{aligned}
p(B|D) &= 2\int_{\rho=\frac{BD^2}{L}}^{1} \frac{f(\beta_0)}{|g'(\beta_0)|}\, d\rho \\
&= 2\int_{\rho=\frac{BD^2}{L}}^{1} \sin(\beta_0)\cos(\beta_0)\frac{D^2}{L \cdot \rho \cdot \sin(\beta_0)} d\rho \\
&= 2\int_{\rho=\frac{BD^2}{L}}^{1} \frac{\sin\left(\cos^{-1}\left(\frac{BD^2}{\rho L}\right)\right)\cos\left(\cos^{-1}\left(\frac{BD^2}{\rho L}\right)\right)D^2}{L \cdot \rho \cdot \sin\left(\cos^{-1}\left(\frac{BD^2}{\rho L}\right)\right)} d\rho \\
&= 2\int_{\rho=\frac{BD^2}{L}}^{1} \frac{BD^2}{\rho L}\frac{D^2}{L \cdot \rho}\, d\rho \\
&= \frac{2BD^4}{L^2}\int_{\rho=\frac{BD^2}{L}}^{1} \frac{1}{\rho^2} d\rho \\
&= \frac{2D^2}{L}\left[1 - \frac{B \cdot D^2}{L}\right]
\end{aligned}$$

The same approach can be taken in order to find $p(D|B)$, albeit with an extra step as we will first determine $p(D^2|B)$ and derive our desired distribution from there. We start by again using the Dirac delta to model the probability density of the square of the distance, if all the other variables are known:

$$p(D^2|D,\rho,\beta) = \delta\left(D^2 - \frac{L\cdot\rho\cdot\cos\beta}{B}\right).$$

Assuming the same distributions for slant β and albedo ρ, we see a similar form:

$$p(D^2|B) = 2\int_{\rho=0}^{1}\int_{\beta=0}^{\frac{\pi}{2}} \delta\left(D^2 - \frac{L\cdot\rho\cdot\cos(\beta)}{B}\right)\sin(\beta)\cos(\beta)\,d\beta d\rho.$$

Despite the fact that the brightness and square of distance have swapped positions within the Dirac delta distribution, the root of the function with respect to β remains unchanged. What does change, however, is the derivative of this inner term with respect to β.

Let $g(\beta) = D^2 - \frac{L\cdot\rho\cdot\cos(\beta)}{B}$. Then $g'(\beta) = \frac{L\cdot\rho\cdot\sin(\beta)}{\mathrm{B}}$.

The rest of the derivation remains unchanged, leading to

$$p(D^2|B) = \frac{2B}{L}\left[1 - \frac{B\cdot D^2}{L}\right].$$

Of course, it is the probability density of D that we are interested, which is derived from the above via a transformation of random variables. The general case for an increasing function $y = g(x)$, as demonstrated in [112] is given as:

$$p_y(y) = p_x\big(g^{-1}(y)\big)\frac{\mathrm{d}}{dx}g^{-1}(y).$$

So for the simple case of $y = \sqrt{x}$ we have

$$p_y(y) = p_x(y^2)2y.$$

Substituting x for D^2 and y for D gives:

$$p(D|B) = \frac{4DB}{L}\left[1 - \frac{B \cdot D^2}{L}\right].$$

Appendix C Derivation of Conditional Likelihood with Normal Estimate

Using the wrapped phase measurements in the local area of each pixel, it is possible to make an estimate of the surface normal. As demonstrated in Appendix A, if the surface is assumed to be equally likely to be facing any direction (within the hemisphere facing the camera) then the probability density of the slant with respect to the pixel ray can be modeled as:

$$p(\beta) = 2\sin(\beta)\cos(\beta).$$

Instead, this will be modeled as a Gaussian centered at the slant determined by the normal estimate, $\hat{\beta}$ and assumed distance D (recall that the choice of K—and hence D—determines the value of $\hat{\beta}$), :

$$p\left(\beta \middle| D, \hat{\beta}\right) = N\left(\hat{\beta}, \sigma_{\beta}^{2}\right).$$

Picking up at the beginning of Equation (3-8):

$$\begin{aligned} p\left(B \middle| D, \hat{\beta}\right) &= \int_{0}^{1}\int_{0}^{\frac{\pi}{2}} p\left(B|D,\beta,\rho,\hat{\beta}\right)p(\beta|\rho,\hat{\beta})p\left(\rho\middle|\hat{\beta}\right)d\beta\, d\rho \\ &= \frac{1}{\sigma_{\beta}\sqrt{2\pi}}\int_{0}^{1}\int_{0}^{\frac{\pi}{2}} \delta\left(B - \frac{L\cdot\rho\cdot\cos(\beta)}{D^{2}}\right) e^{\frac{-(\beta-\bar{\beta})^{2}}{2\sigma_{\beta}^{2}}} d\beta\, d\rho \end{aligned}$$

Again, integrating with the Dirac delta, we can use the formula:

$$\int_{-\infty}^{\infty} f(x)\delta\big(g(x)\big)dx = \sum_{i\in\Re}\frac{f(x_i)}{|g'(x_i)|}$$

Let $g(\beta) = B - \frac{L \cdot \rho \cdot \cos(\beta)}{D^2}$. Then $g'(\beta) = \frac{L \cdot \rho \cdot \sin(\beta)}{D^2}$,

with root at $\beta_0 = \cos^{-1}\left(\frac{BD^2}{\rho L}\right)$.

Let $f(\beta) = e^{\frac{-(\beta-\hat{\beta})^2}{2\sigma_\beta^2}}$.

Then

$$p\left(B\middle|D,\hat{\beta}\right) = \int_{\rho=\frac{BD^2}{L}}^{1} \frac{f(\beta_0)}{|g'(\beta_0)|} d\rho$$

$$= \frac{1}{\sigma_\beta\sqrt{2\pi}} \int_{\rho=\frac{BD^2}{L}}^{1} \frac{D^2}{L \cdot \rho \cdot \sin(\beta_0)} e^{\frac{-(\beta_0-\hat{\beta}_q)^2}{2\sigma_\beta^2}} d\rho$$

$$= \frac{D^2}{L \cdot \sigma_\beta\sqrt{2\pi}} \int_{\rho=\frac{BD^2}{L}}^{1} \frac{e^{\frac{-\left(\cos^{-1}\left(\frac{BD^2}{\rho L}\right)-\hat{\beta}_q\right)^2}{2\sigma_\beta^2}}}{\sqrt{\rho^2 - \left(\frac{BD^2}{L}\right)^2}} d\rho$$

Works Cited

[1] Aseem Agarwala et al., "Interactive digital photomontage," *ACM Trans. Graph*, vol. 23, pp. 294-302, 2004.

[2] Heikki Ailisto et al., "Scannerless imaging pulsed-laser range finding," *Journal of Optics A: Pure and Applied Optics*, vol. 4, no. 6, p. S337, 2002.

[3] Apple, Inc. (2008) [Online]. http://www.apple.com/macosx/features/ichat.html

[4] Cyrus S. Bamji and et al., "A 0.13 µm CMOS System-on-Chip for a 512× 424 Time-of-Flight Image Sensor With Multi-Frequency Photo-Demodulation up to 130 MHz and 2 GS/s ADC," *Solid-State Circuits, IEEE Journal of* , vol. 50, no. 1, pp. 303-319, 2015.

[5] Stephen T Barnard and Martin A Fischler, "Computational stereo," *ACM Computing Surveys (CSUR)*, vol. 14, no. 4, pp. 553-572, 1982.

[6] J L Barron, D J Fleet, S S Beauchemin, and T A Burkitt, "Performance of optical flow techniques," in *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR '92., 1992 IEEE Computer Society Conference on*, 1992, pp. 236-242.

[7] John Barron and Jitendra Malik, "Shape, albedo, and illumination from a single image of an unknown object," in *Computer Vision and Pattern Recognition*, Providence, RI, USA, 2012.

[8] Jorge Batista, Paulo Peixoto, Catarina Fern, and Miguel Ribeiro, "A Dual-Stage Robust Vehicle Detection and Tracking for Real-time Traffic Monitoring," in

[9] C. Beder, B. Barzak, and R. Koch, "A combined approach for estimating patchlets afrom pmd depth images and stereo intensity images," in *German Association for Pattern Recognition (DAGM)*, Heidelberg, 2007.

[10] Arrigo Benedetti, Travis Perry, Mike Fenton, and Vishali Mogallapu, "Methods and systems for geometric phase unwrapping in time of flight systems.," US 20140049767 A1, 2014.

[11] Martin Böhme, Martin Haker, Thomas Martinetz, and Erhardt Barth, "Shading constraint improves accuracy of time-of-flight measurements,"

Computer vision and image understanding, vol. 114, no. 12, pp. 329-1335, 2010.

[12] Y Y Boykov and M P Jolly, "Interactive graph cuts for optimal boundary region segmentation of objects in N-D images," in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 1, 2001, pp. 105-112.

[13] Yuri Boykov, Olga Veksler, and Ramin Zabih, "Fast approximate energy minimization via graph cuts," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 11, pp. 1222-1239, 2001.

[14] Lisa Gottesfeld Brown, "A survey of image registration techniques," *ACM computing surveys (CSUR)*, vol. 24, no. 4, pp. 325-376, 1992.

[15] Canesta Inc. (2008) [Online]. http://www.canesta.com

[16] Yeou-Yen Cheng and James C Wyant, "Multiple-wavelength phase-shifting interferometry," *Applied optics*, vol. 24, no. 6, pp. 804-807, 1985.

[17] O.T.-C. Chen, Kuan-Hsien Lin, and Zhe Ming Liu, "High-efficiency 3D CMOS image sensor," in *OptoElectronics and Communications Conference and International Conference on Photonics in Switching (OECC/PS)*, 2013, pp. 1-2.

[18] Ouk Choi and S. Lee, "Fusion of time-of-flight and stereo for disambiguation of depth measurements," in *Asian Conference on Computer Vision*, Daejeon, Korea, 2012, pp. 640-653.

[19] Ouk Choi and Seungkyu Lee, "Wide range stereo time-of-flight camera," in *International Conference on Image Processing*, Orlando, 2012.

[20] O. Choi, S. Lee, and H Lim, "Interframe consistent multifrequency phase unwrapping for time-of-flight cameras," *Optical Engineering*, vol. 52, no. 5, 2013.

[21] Ouk Choi et al., "Range unfolding for time-of-flight depth cameras," in *International Conference on Image Processing (ICIP)*, Hong Kong, 2010.

[22] Andrea Colombari, Andrea Fusiello, and Vittorio Murino, "Background Initialization in Cluttered Sequences," in *CVPRW '06: Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop*, Washington, DC, USA, 2006, p. 197.

[23] Ryan Crabb and Roberto Manduchi, "Probabilistic Phase Unwrapping for Single-Frequency Time-of-Flight Range Cameras," in *Proc. International Conference on 3D Vision (3DV 14)*, Tokyo, 2014.

[24] Ryan Crabb, Colin Tracey, Akshaya Puranik, and James Davis, "Real-time Foreground Segmentation via Range and Color Imaging," in *CVPR Workshop on Time of Flight Camera Based Comp Vis*, Anchorage, AK, 2008.

[25] A Criminisi, G Cross, A Blake, and V Kolmogorov, "Bilayer segmentation of live video," in *In IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006, pp. 53-60.

[26] R Cucchiara, C Grana, M Piccardi, and A Prati, "Detecting moving objects, ghosts and shadows in video streams," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, October 2003, pp. 1337-1342.

[27] Rene Daendliker, Kurt Hug, Jacob Politch, and Eric Zimmermann, "High-accuracy distance measurements with multiple-wavelength interferometry," *Optical Engineering*, vol. 34, no. 8, pp. 2407-2412, 1995.

[28] C. Dal Mutto, P. Zanuttigh, and G. M. Cortelazzo, "A probabilistic approach to ToF and stereo data fusion," in *3DPVT*, Paris, 2010.

[29] C. Dal Mutto, P. Zanuttigh, and G. M. Cortelazzo, *Time-of-Flight Cameras and Microsoft Kinect™*.: Springer Science & Business Media, 2012.

[30] Andrew Delong, Anton Osokin, Hossam N Isack, and Yuri Boykov, "Fast approximate energy minimization with label costs," *International Journal of Computer Vision*, vol. 96, no. 1, pp. 1-27, 2012.

[31] Umesh R Dhond and Jake K Aggarwal, "Structure from stereo-a review," *IEEE transactions on systems, man, and cybernetics*, vol. 19, no. 6, pp. 1489-1510, 1989.

[32] AA Dorrington, MJ Cree, DA Carnegie, and AD Payne, "Selecting signal frequencies for best performance of Fourier-based phase detection," in *Proc. Twelfth New Zealand Electronics Conference*, 2005, pp. 189-193.

[33] Adrian A Dorrington, Michael J Cree, Andrew D Payne, Richard M Conroy, and Dale A Carnegie, "Achieving sub-millimetre precision with a solid-state full-field heterodyning range imaging camera," *Measurement Science and Technology*, vol. 18, no. 9, p. 2809, 2007.

[34] D. Droeschel, D. Holz, and S. Behnke, "Multi-frequency phase unwrapping for time-of-flight cameras," in *Intelligent Robots and Systems (IROS)*, Taipei, 2010.

[35] D. Droeschel, D. Holz, and S. Behnke, "Probabilistic phase unwrapping for time-of-flight cameras," in *41st International Symposium on Robotics (ISR)*, Munich, Germany, 2010, pp. 1-7.

[36] Ahmed Elgammal et al., "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance," in *Proceedings of the IEEE*, 2002, pp. 1151-1163.

[37] Ahmed Elgammal, David Harwood, and Larry Davis, "Non-parametric model for background subtraction," in *FRAME-RATE Workshop, IEEE*, 2000, pp. 751-767.

[38] J. F., Niemann, A., & Koch, R. Evers-Senne, "Visual reconstruction using geometry guided photo consistency," in *Vision, Modeling, and Visualization 2006*, Aachen, Germany, 2006.

[39] D. Falie and V. Buzuloiu, "Wide range time of flight camera for outdoor surveillance," in *Microwaves, Radar and Remote Sensing Symposium (MRRS)*, Kiev, 2008.

[40] Olivier Faugeras, *Three-dimensional computer vision: a geometric viewpoint*: MIT press, 1993.

[41] Pedro F. Felzenszwalb and Daniel P. Huttenlocher, "Efficient belief propagation for early vision," *International journal of computer vision*, vol. 70, no. 1, pp. 41-54, 2006.

[42] Mario Frank et al., "Theoretical and experimental error analysis of continuous-wave time-of-flight range cameras," *Optical Engineering*, vol. 48, no. 1, pp. 13602-13602, 2009.

[43] Brendan J. Frey, Ralf Koetter, and Nemanja Petrovic, "Very loopy belief propagation for unwrapping phase images," in *Neural Information Processing Systems (NIPS)*, Vancouver, 2001.

[44] N Friedman and S Russell, "Image segmentation in video sequences: a probabilistic approach," in *Proceedings of the 13th Conference on Uncertainty in Artificial Intelligence*, 1997.

[45] Dennis Ghiglia and Mark Pritt, *Two-dimensional phase unwrapping: theory, algorithms, and software*. New York: Wiley, 1998.

[46] S Burak Göktürk, Hakan Yalcin, and Cyrus Bamji, "A Time-Of-Flight Depth Sensor - System Description, Issues and Solutions," in *CVPRW '04: Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04) Volume 3*, Washington, DC, USA, 2004, p. 35.

[47] R.M Goldstein, H. A. Zebker, and C. L. and Werner, "Satellite radar interferometry: two-dimensional phase unwrapping," *Radio Science*, vol. 23, no. 4, pp. 713-720, 1988.

[48] W E Grimson, C Stauffer, and R Romano, "Using adaptive tracking to classify and monitor activities in a site," in *CVPR*, 1998.

[49] Sigurjon Arni Gudmundsson, Henrik Aanaes, and Rasmus Larsen, "Fusion of stereo vision and time-of-flight imaging for improved 3d estimation," *International Journal of Intelligent Systems Technologies and Applications*, vol. 5, no. 3, pp. 425-433, 2008.

[50] D Gutchess, M Trajkovi C, E Cohen-solal, D Lyons, and A K Jain, "A background model initialization algorithm for video surveillance," in *Proc. IEEE ICCV 2001, Pt.1*, 2001, pp. 733-740.

[51] Martin Haker, *Gesture-Based Interaction with Time-of-Flight Cameras.*, 2010.

[52] R., & Zisserman, A Hartley, *Multiple View Geometry.*: Cambridge university press, 2003.

[53] M. Harville, "A framework for high-level feedback to adaptive, per-pixel, mixture-of-gaussian background models," in *European Conference on Computer Vision*, Copenhagen, 2002.

[54] Janne Heikkilä and Olli Silvén, "A Real-Time System for Monitoring of Cyclists and Pedestrians," in *VS '99: Proceedings of the Second IEEE Workshop on Visual Surveillance*, Washington, DC, USA, 1999, p. 74.

[55] Zhong Heping, Zhang Sen, and Tang Jinsong, "Path following algorithm for phase unwrapping based on priority queue and quantized quality map," in *IEEE Computational Intelligence and Software Engineering*, 2009, pp. 1-4.

[56] J. Huang and D. Mumford, "Statistics of natural images and models," in *Computer Vision and Pattern Recognition*, Fort Collins, CO, USA, 1999.

[57] G. J. Iddan and G. Yahav, "3D Imaging in the studio (and elsewhere)," in *Proc. SPIE 4298, Three-Dimensional Image Capture and Applications*, 2001.

[58] O. Javed, K. Shafique, and M. Shah, "A hierarchical approach to robust background subtraction using color and gradient information," in *Workshop on Motion and Video Computing*, 2002, pp. 22--27.

[59] Adrian PP Jongenelen et al., "Analysis of errors in tof range imaging with dual-frequency modulation," *Instrumentation and Measurement, IEEE Transactions on*, vol. 60, no. 5, pp. 1861-1868, 2011.

[60] A. Kadambi et al., "Coded time of flight cameras: sparse deconvolution to address multipath interference and recover time profiles," *ACM Transactions on Graphics*, vol. 32, no. 6, p. 167, 2013.

[61] Shunsuke Kamijo, Katsushi Ikeuchi, and Masao Sakauchi, "Vehicle tracking in low-angle and front view images based on spatio-temporal markov random fields," in *In Proceedings of the 8th World Congress on Intelligent Transportation Systems*, 2001.

[62] K.T. Kim, M. Siegel, and J.Y. Son, "Synthesis of a high-resolution 3D stereoscopic image pair from a high-resolution monoscopic image and a low-resolution depth map," in *SPIE: Stereoscopic Displays and Applications IX*, San Jose, USA, 1998.

[63] Sebastian Knorr, Matthias Kunter, and Thomas Sikora, "Stereoscopic 3D from 2D video with super-resolution capability," *Signal Processing: Image Communication*, vol. 23, no. 9, pp. 665-676, 2008.

[64] D Koller et al., "Towards robust automatic traffic scene analysis in real-time," in *Proceedings on the International Conference on Pattern Recognition*, Israel, 1994, pp. 126-131.

[65] V Kolmogorov, A Criminisi, A Blake, G Cross, and C Rother, "Bi-Layer Segmentation of Binocular Stereo Video," in *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, San Diego, CA, 2005, pp. 407-414.

[66] Johannes Kopf, Michael F Cohen, Dani Lischinski, and Matt Uyttendaele, "Joint bilateral upsampling," *ACM Trans. Graph.*, vol. 26, no. 3, p. 96, 2007.

[67] Robert Lange, "3D time-of-flight distance measurement with custom solid-state image sensors in CMOS/CCD-technology," *Diss., Department of Electrical Engineering and Computer Science, University of Siegen*, 2000.

[68] J P Lewis, "Fast Template Maching," in *VI95*, 1995, pp. 120-123.

[69] Marvin Lindner and Andreas Kolb, "Compensation of motion artifacts for time-of-flight cameras," in *Dynamic 3D Imaging*.: Springer, 2009, pp. 16-27.

[70] B P Lo and S A Velastin, "Automatic congestion detection system for underground platforms," *Intelligent Multimedia, Video and Speech Processing, 2001. Proceedings of 2001 International Symposium on*, pp. 158-161, 2001.

[71] D Magee, "Tracking Multiple Vehicles using Foreground, Background and Motion Models," in *Image and Vision Computing*, vol. 22, September 2004, pp. 143-155.

[72] David Marr and Tomaso Poggio, "A computational theory of human stereo vision," *Proceedings of the Royal Society of London B: Biological Sciences*, vol. 204, no. 1156, pp. 301-328, 1979.

[73] Y., Terasaki, H., Sugimoto, K., & Arakawa, T. Matsumoto, "Conversion system of monocular image sequence to stereo using motion parallax," in *International Society for Optics and Photonics}: Electronic Imaging'97*, 1997, pp. 108--115.

[74] Stefan May, David Droeschel, Stefan Fuchs, Dirk Holz, and Andreas Nüchter, "Robust 3D-mapping with time-of-flight cameras," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, 2009, pp. 1673-1678.

[75] Shane H. McClure, Cree Michael J., Adrian Dorrington, and Andrew Payne, "Resolving depth-measurement ambiguity with commercially available range imaging cameras," in *IS&T/SPIE Electronic Imaging*, San Jose, CA, USA, 2010.

[76] J. Mei, A. Kirmani, A. Colaco, and V.K. Goyal, "Phase unwrapping and denoising for time-of-flight imaging using generalized approximate

message passing," in *20th IEEE International Conference on Image Processing (ICIP)*, Melbourne, Australia, 2013, pp. 364-368.

[77] Toshihiko Misu et al., "Scene-adaptive switching of segmentation methods for object-based video encoding," *Syst. Comput. Japan*, vol. 35, no. 8, pp. 31-44, 2004.

[78] Anurag Mittal and Nikos Paragios, "Motion-based background subtraction using adaptive kernel density estimation," , 2004, pp. 302-309.

[79] Todd C Monson et al., "Characterization of scannerless LADAR," in *AeroSense'99*, 1999, pp. 409-420.

[80] Eric N Mortensen and William A Barrett, "Intelligent scissors for image composition," in *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, New York, NY, USA, 1995, pp. 191-198.

[81] Thierry Oggier et al., "An all-solid-state optical range camera for 3D real-time imaging with sub-centimeter depth resolution (SwissRanger)," in *Optical Design and Engineering*, 2004, p. 534.

[82] N M Oliver, B Rosario, and A P Pentland, "A Bayesian Computer Vision System for Modelling Human interactions," *IEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, August 2000.

[83] Serban Oprisescu, Dragos Falie, Mihai Ciuc, and Vasile Buzuloiu, "Measurements with ToE cameras and their necessary corrections," in *International Symposium on Signals, Circuits and Systems*, Iasi, Romania, 2007.

[84] N Paragios and V Ramesh, "A MRF-based approach for real-time subway monitoring," *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, p. 1034, 2001.

[85] Sylvain Paris and Frédo Durand, "A fast approximation of the bilateral filter using a signal processing approach," in *In: Proceedings of European Conference on Computer Vision '06*, 2006, pp. 568-580.

[86] A. D. Payne, A. P. Jongenelen, A. A. Dorrington, M. J. Cree, and D. A Carnegie, "Multiple frequency range imaging to remove measurement ambiguity," in

Conference on Optical 3-D Measurement Techniques, Vienna, Austria, 2009.

[87] Matthias Plaue, "Technical report: Analysis of the PMD imaging system," *Interdisciplinary Center for Scientific Computing, University of Heidelberg*, 2006.

[89] C and Giani, M and Leuratti, N Prati, "SAR Interferometry: A 2-D phase unwrapping technique based on phase and absolute values informations," in *Geoscience and Remote Sensing Symposium*, 1990, pp. 2043-2046.

[90] M.D. Pritt, "Phase unwrapping by means of multigrid techniques for interferometric SAR," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 34, pp. 728-738, 1996.

[91] Holger Rapp, "Experimental and theoretical investigation of correlating TOF-camera systems," 2007.

[92] Andreas Reichinger. (2011) Kinect Pattern Uncovered. [Online]. https://azttm.wordpress.com/2011/04/03/kinect-pattern-uncovered/

[93] J Rittscher, J Kato, S Joga, and A Blake, "A Probabilistic Background Model for Tracking," in *Proceedings of European Conference on Computer Vision*, 2000, pp. 336-350.

[94] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake, "GrabCut: interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 309-314, August 2004.

[95] Radu Bogdan Rusu and Steve Cousins, "3D is here: Point Cloud Library (PCL)," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, 2011.

[96] Mark A Ruzon and Carlo Tomasi, "Alpha Estimation in Natural Images," *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, vol. 1, p. 1018, 2000.

[97] Daniel Scharstein and Richard Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International journal of computer vision*, vol. 47, no. 1-3, pp. 7-42, 2002.

[98] Mirko Schmidt, "Analysis, modeling and dynamic optimization of 3d time-of-flight imaging systems," 2011.

[99] Martin Otmar Schmidt, "Spatiotemporal analysis of range imagery," 2008.

[100] Jianbo Shi and Jitendra Malik, "Motion segmentation and tracking using normalized cuts," in *Computer Vision and Pattern Recognition, IEEE Sixth International Conference on*, Santa Barbara, CA, USA., May 1998, pp. 1154--1160.

[101] Jianbo Shi and Jitendra Malik, "Normalized Cut and Image Segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 8, pp. 888-905, 2000.

[102] Alvy Ray Smith and James F Blinn, "Blue screen matting," in *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, New York, NY, USA, 1996, pp. 259-268.

[103] Chris Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1999, p. 2246.

[104] Jian Sun, Weiwei Zhang, Xiaoou Tang, and Heung-yeung Shum, "Background cut," in *In ECCV*, 2006, pp. 628-641.

[105] R. Szeliski et al., "A comparative study of energy minimization methods for markov random fields with smoothness-based priors," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 6, pp. 1068-1080, 2008.

[106] Dean Takahashi, "Microsoft discloses details on Xbox One and Kinect chips," *VentureBeat*, August 2013.

[107] C Tomasi and R Manduchi, "Bilateral Filtering for Gray and Color Images," , 1998, pp. 839-846.

[108] K Toyama, J Krumm, and B Meyers, "Wallflower: Principles and practice of background maintenance," in *International Conference on Computer Vision*, 1999.

[109] G., and Bioucas-Dias, J. Valadao, "Phase imaging: Unwrapping and denoising with diversity and multi-resolution," in *International Workshop on Local and Non-Local Approximation in Image Processing (LNLA)*, Lausanne, Switzerland, 2008.

[110] V. Vezhnevets and V. Konouchine, ""GrowCut" - Interactive Multi-Label N-D Image Segmentation By Cellular Automata," in *Graphicon '05*, Novosibirsk Akademgorodok, RU, 2005.

[111] Oliver Wang, Jonathan Finger, Qingxiong Yang, James Davis, and Ruigang Yang, "Automatic Natural Video Matting with Depth," in *PG '07: Proceedings of the 15th Pacific Conference on Computer Graphics and Applications*, Washington, DC, USA, 2007, pp. 469-472.

[112] Josheph C. Watkins. (2009, September) Transformations of Random Variables. Document. [Online]. http://math.arizona.edu/~jwatkins/f-transform.pdf

[113] Christopher Wren, Ali Azarbayejani, Trevor Darrell, and Alex Pentl, "Pfinder: Real-Time Tracking of the Human Body," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 780-785, 1997.

[114] Jiangjian Xiao and Mubarak Shah, "Motion Layer Extraction in the Presence of Occlusion Using Graph Cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1644-1659, 2005.

[115] Z Xu, "Investigation of 3D-Imaging Systems Based on Modulated Light and Optical RF-Interferometry," *ORFI), ZESS Forschungsberichte, Shaker-Verlag, ISBN*, pp. 3-8265, 1999.

[116] W. Xu et al., "Phase-unwrapping of SAR interferogram with multi-frequency or multi-baseline," in *Geoscience and Remote Sensing Symposium*, Pasadena, CA, USA, 1994.

[117] Xun Xu and Thomas S Huang, "A Loopy Belief Propagation Approach for Robust Background Estimation," in *CVPR 08*, 2008.

[118] Wei Xu, Yue Zhou, Yihong Gong, and Hai Tao, "Background Modeling Using Time Dependent Markov Random Field With Image Pyramid," ,

[119] Qingxiong Yang, "A non-local cost aggregation method for stereo matching," in *Computer Vision and Pattern Recognition (CVPR)*, 2012.

[121] Zhengyou Zhang, "Microsoft kinect sensor and its effect," *IEEE Multimedia*, vol. 19, no. 2, pp. 4-10, 2012.

[120] Liang Zhang, Carlos Vazquez, and Sebastian and Knorr, "3D-TV content creation: automatic 2D-to-3D video conversion," *Broadcasting, IEEE Transactions on*, vol. 57, no. 2, pp. 372--383, 2011.

[122] Y Zhou and H Tao, "An Background Layer Model for Object Tracking through Occlusion," in *ICCV03*, 2003.

[123] J. Zhu, L. Wang, R. Yang, and J. Davis, "Fusion of time-of-flight depth and stereo for high accuracy depth maps," in *Computer Vision and Pattern Recognition*, Anchorage, 2008.