Mobile Structured Light: Reconstruction using a Signal Processing Approach

Proefschrift voorgelegd tot het behalen van de graad van
Doctor in de Wetenschappen, Informatica, te verdedigen door:

Chris HERMANS

Promotor: prof. dr. Philippe Bekaert
Copromotor: prof. dr. Frank van Reeth
Mobile Structured Light: Reconstruction using a Signal Processing Approach

Proefschrift voorgelegd tot het behalen van de graad van Doctor in de Wetenschappen, Informatica, te verdedigen door:

Chris HERMANS

Promotor: prof. dr. Philippe Bekaert
Copromotor: prof. dr. Frank van Reeth
Acknowledgments

The work contained within this dissertation could not have been produced without the help and support of many different people. Therefore, before we go into technical territory, I would like to thank everyone who directly or indirectly contributed to its creation.

First of all, I would like to thank my advisor, Prof. dr. Philippe Bekaert, who offered me the opportunity to work in an exciting field or research, the support of fully fledged working environment, and the freedom to pursue my interests as I saw fit. Getting the chance to get my first work experience doing something that I truly enjoyed, was well appreciated.

I am also grateful to the members of my Ph.D. committee and jury who reviewed this dissertation and provided me with valuable comments: Prof. dr. Philippe Bekaert, dr. Benedict Brown, Ingo Feldmann, Prof. dr. ir. Jacques Verly, Prof. dr. Frank Van Reeth, Prof. dr. Wim Lamotte and Prof. dr. Marc Gyssens.

As a Ph.D student you get to travel around the world, presenting and discussing new ideas. This was a great opportunity to visit foreign countries, meet interesting people, and last but not least, try the local cuisine. I’d like to thank Roger Claes and Ingrid Konings for taking care of the financial and administrative aspects that come with these visits.

Before we could visit these conferences however, of course first a lot of effort had to go into actually getting the work done and published. As this is not a solitary job, I would like to thank my colleagues from the computer graphics department, past and present: Codruta Ancuti, Cosmin Ancuti, Tom Cuypers, Bert De Decker, Fabian Di Fiore, Maarten Dumont, Jan Fransens, Karel Frederix, Mark Gerrits, Patrik Goorts, Tom Haber, Erik Hubo, Steven Maesen, Tom Mertens, Sammy Rogmans, Johannes Taelman, Tom Van Laerhoven, Cedric Vanaken and William Van Haevre.
Last but not least, I would like to thank the people closest to me: my girlfriend Heleen Evers for all her love, and the necessary moral support when I occasionally hit a brick wall in my research, reminding me that in time a solution always presents itself; my parents, for giving me the opportunity to lead the life I live and always being there for me when I needed them; and my good old friend Koen Heyens, for providing the necessary distractions from work when needed. Finally I would like to give an honorable mention to my dog Dux, who loyally lay at my feet during the whole time I was writing this text (or at least, for as long as I had cookies to spare).

Diepenbeek April 2011.
Abstract

In this dissertation we introduce a new set of 3D shape acquisition methods, which we have called *mobile structured light*. Unlike the more classical structured light methods, in which a static projector illuminates a (static) scene with a variety of time-varying illumination patterns, our proposed technique makes use of a mobile projector emitting a single static sinusoidal illumination pattern. This projector, which we refer to as a *sliding projector*, is translated at a constant velocity in the direction of the projector’s horizontal axis. Optionally, we use a secondary illumination pattern, a De Bruin pattern tailored onto the original wave pattern, to allow for online extrinsic calibration of our setup.

Illuminating the object in this manner allows us to perform fast per pixel computations, in which we can analyze the incoming illumination using traditional methods from signal processing literature. By employing Fourier analysis to decompose the observed illumination sequence into a corresponding set of frequency components, we are able to recover pixel depth. We have observed that there exist a linear relationship between the depth of a scene point, defined as the distance between the projector’s principal plane and the scene point, and the frequency of the observed illumination sequence in the corresponding pixel. Thus, either with precise pre-calibration of the setup, or using a secondary pattern for extrinsic post-calibration, we are able to convert these depth values into a valid 3D reconstruction.

As we have explicitly cast depth estimation as a signal processing problem, we are able to borrow from the vast literature that exists on the discussed topics. In this dissertation, we have examined the influence of applying frequency refinement techniques from the single tone estimation problem to our own approach, and noticed a significant improvement in accuracy and reduction in required frames. Furthermore, we have shown compatibility between new developments in the area of compressed sensing, which could potentially be applied to a pre-calibrated implementation of our technique.
The proposed approach has several advantages over classical structured light methods. As the method performs depth estimation on a per pixel basis, it is able to preserve sharp edges in the produced depth image. Furthermore, unlike classical structured light methods, the quality of our results is not limited by projector or camera resolution, but is solely dependent on the temporal sampling density of the captured image sequence. Additional benefits include a significant robustness against common problems encountered with structured light methods, such as occlusions, specular reflections, subsurface scattering, interreflections, and to a certain extent projector defocus.

Finally, we have discussed the dual relationship between epipolar plane image analysis, a passive shape acquisition technique which uses a linearly translating camera in order to capture the scene structure, and the proposed mobile structured light method. EPI analysis exploits the large amount of inherent structure in the recorded image volume to recover scene structure, relying on data redundancy and a color consensus matching criterium to establish cross-frame correspondences. It is this same data redundancy that determines the accuracy of the observed frequency in our proposed dual technique, without the requirements of an order constraint or the restrictive nature of a color consistency assumption. Thus, mobile structured light allows for the acquisition of a much larger variety of materials, whereas EPI analysis was restricted mainly to Lambertian surfaces.
Contents

Acknowledgments i
Abstract iii
Contents v
List of Figures xi

1 Introduction 1
  1.1 Problem Statement and Motivation .......................... 2
  1.2 Summary of Contributions .................................. 5
  1.3 Overview of the Dissertation ................................ 5

2 Related Work - Shape Acquisition 7
  2.1 Contact Scanners ........................................... 9
  2.2 Non-contact Scanners, Passive .............................. 9
  2.3 Non-contact Scanners, Active ............................... 11

3 Related Work - Camera/Projector 3D Scanning Systems 19
  3.1 Why use active illumination? ............................... 20
  3.2 Temporal Encoding ......................................... 22
    3.2.1 Binary Codes ......................................... 22
    3.2.2 N-ary Codes .......................................... 25
    3.2.3 Phase shifting ........................................ 28
  3.3 Spatial Encoding ............................................ 31
    3.3.1 De Bruijn Sequences and M-Sequences ................. 32
    3.3.2 M-Arrays ............................................. 35
  3.4 Spatio-Temporal Encoding ................................. 37
    3.4.1 Spacetime Stereo .................................... 37
CONTENTS

6 Calibration - Removing the Camera Placement Constraint 85
  6.1 Labeling the Projector Pixels ................................. 86
    6.1.1 Temporal Encoding ...................................... 86
    6.1.2 Signal-to-Code Conversion ............................... 87
    6.1.3 Time Multiplexing ...................................... 88
  6.2 Recovering Structure and Projector Motion ..................... 88
  6.3 Recovering Camera Parameters & Dense Depth Map ............... 90
  6.4 Calibrating the Setup ...................................... 90
  6.5 Discussion ................................................ 90
    6.5.1 Time Multiplexing ...................................... 92
    6.5.2 Signal-to-Code Conversion ............................... 92
    6.5.3 Estimated Reconstruction ................................ 93
  6.6 Conclusion .............................................. 94

7 Epipolar Plane Image Analysis & Mobile Structured Light 95
  7.1 Related Work ............................................. 96
    7.1.1 Sparse Feature Based Methods ......................... 96
    7.1.2 Dense Plane-Sweeping Methods ......................... 97
    7.1.3 Triangulation vs. Plane-Sweeping in Active Vision .... 98
    7.1.4 Viewpoint Encoding Equivalent in Passive Stereo? .... 98
  7.2 Epipolar Plane Image Analysis: A Dense Alternative ........ 98
    7.2.1 What is EPI Analysis? ................................ 98
    7.2.2 From Classical Stereo to EPI Analysis ................ 99
    7.2.3 Matching Criteria ..................................... 100
    7.2.4 EPI Structure and Occlusions .......................... 100
    7.2.5 Dense Reconstruction .................................. 101
  7.3 From EPI Analysis to Mobile Structured Light ................ 105
  7.4 Beyond Linear Camera (and Projector) Paths .................. 106
    7.4.1 Circular Camera Trajectories ......................... 106
    7.4.2 Circular Projector Trajectories ....................... 111

8 Conclusions 115
  8.1 Summary ............................................... 116
    8.1.1 Depth by Frequency Estimation ....................... 116
    8.1.2 Removing the Camera Constraint ...................... 117
    8.1.3 The EPI-MSL Duality .................................. 117
    8.1.4 Comparison to State of the Art ....................... 118
  8.2 Future Work ............................................ 118
<table>
<thead>
<tr>
<th>viii</th>
<th>CONTENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Appendices</td>
<td>121</td>
</tr>
<tr>
<td>A Scientific Contributions and Publications</td>
<td>123</td>
</tr>
<tr>
<td>B Samenvatting (Dutch Summary)</td>
<td>125</td>
</tr>
<tr>
<td>Bibliography</td>
<td>129</td>
</tr>
</tbody>
</table>
## List of Figures

2.1 Depth map ......................................................... 9  
2.2 Visual hull ....................................................... 10  
2.3 Triangulation based laser scanning ........................ 12  
2.4 Time-of-flight devices ......................................... 12  
2.5 X-ray computed tomography (CT) .......................... 14  
2.6 (Functional) magnetic resonance imaging ............... 15  
2.7 Ultrasound imaging ............................................ 16  
2.8 Structured light .................................................. 17  

3.1 Camera/projector 3D scanning ............................... 20  
3.2 Gray codes vs. binary codes: theoretical analysis .... 23  
3.3 Gray codes vs. binary codes: practical analysis ....... 24  
3.4 Intensity encoding .............................................. 27  
3.5 Phase shifting, using a number-theoretical approach .. 28  
3.6 The influence of projector blur ............................... 31  
3.7 De Bruijn graph ................................................ 32  
3.8 De Bruijn patterns used for spatial encoding .......... 33  
3.9 M-array representations ...................................... 36  
3.10 M-Arrays used for spatial encoding ...................... 36  
3.11 Spacetime stereo ............................................ 37  
3.12 Stereo from flickering caustics ........................... 38  
3.13 Grey level patterns used for direct encoding .......... 40  
3.14 Viewpoint encoding .......................................... 42  
3.15 Viewpoint code example ..................................... 43  
3.16 Depth from (de)focus ........................................ 46  
3.17 Depth from (de)focus and global light transport ..... 46  
3.18 Comparison between single/multi-focal depth from projector defocus 46
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Single tone estimation</td>
<td>48</td>
</tr>
<tr>
<td>4.2</td>
<td>Single tone estimation by inspecting the Fourier power spectrum</td>
<td>49</td>
</tr>
<tr>
<td>4.3</td>
<td>Coded strobing</td>
<td>52</td>
</tr>
<tr>
<td>5.1</td>
<td>Depth from frequency analysis overview</td>
<td>57</td>
</tr>
<tr>
<td>5.2</td>
<td>The linear relationship between the depth $\mu$ of the scene points and the period $p$ of their observed periodic patterns</td>
<td>58</td>
</tr>
<tr>
<td>5.3</td>
<td>Illustrating: $I(t) = P(t) \cdot R(t) \cdot V(t)$</td>
<td>58</td>
</tr>
<tr>
<td>5.4</td>
<td>Estimating the underlying reflectance function $R(t)$</td>
<td>61</td>
</tr>
<tr>
<td>5.5</td>
<td>Estimating the underlying visibility function $V(t)$</td>
<td>63</td>
</tr>
<tr>
<td>5.6</td>
<td>A picture of our shape acquisition setup</td>
<td>65</td>
</tr>
<tr>
<td>5.7</td>
<td>Influence of the visibility function on the signal power spectrum</td>
<td>66</td>
</tr>
<tr>
<td>5.8</td>
<td>Influence of a specular highlight on the signal power spectrum</td>
<td>67</td>
</tr>
<tr>
<td>5.9</td>
<td>Projector depth vs. camera depth</td>
<td>71</td>
</tr>
<tr>
<td>5.10</td>
<td>How to obtain camera depth values instead of projector depth values</td>
<td>72</td>
</tr>
<tr>
<td>5.11</td>
<td>A set of results produced by our depth acquisition algorithm</td>
<td>74</td>
</tr>
<tr>
<td>5.12</td>
<td>Comparison of our results on a synthetic dataset to the ground truth</td>
<td>75</td>
</tr>
<tr>
<td>5.13</td>
<td>Comparison of our results on another synthetic dataset to the ground truth</td>
<td>76</td>
</tr>
<tr>
<td>5.14</td>
<td>Influence of Fourier interpolation on the accuracy of our depth estimates</td>
<td>78</td>
</tr>
<tr>
<td>5.15</td>
<td>House dataset: unrefined depth map</td>
<td>83</td>
</tr>
<tr>
<td>5.16</td>
<td>House dataset: refined depth map</td>
<td>84</td>
</tr>
<tr>
<td>6.1</td>
<td>Encoding 2D projector pixel positions with a 1D De Bruijn sequence</td>
<td>87</td>
</tr>
<tr>
<td>6.2</td>
<td>Time multiplexing</td>
<td>89</td>
</tr>
<tr>
<td>6.3</td>
<td>Recovering sparse structure and projector motion</td>
<td>89</td>
</tr>
<tr>
<td>6.4</td>
<td>Camera depth results</td>
<td>91</td>
</tr>
<tr>
<td>6.5</td>
<td>Rendered reconstructions of the ‘house’ dataset</td>
<td>92</td>
</tr>
<tr>
<td>7.1</td>
<td>The epipolar plane image volume</td>
<td>99</td>
</tr>
<tr>
<td>7.2</td>
<td>The EPI-strip</td>
<td>100</td>
</tr>
<tr>
<td>7.3</td>
<td>The EPI-tube</td>
<td>101</td>
</tr>
<tr>
<td>7.4</td>
<td>The epipolar plane image vs. disparity space image duality</td>
<td>103</td>
</tr>
<tr>
<td>7.5</td>
<td>Estimating diffuse and specular components for each EPI-strip</td>
<td>104</td>
</tr>
<tr>
<td>7.6</td>
<td>The dual EPI image</td>
<td>106</td>
</tr>
<tr>
<td>7.7</td>
<td>ICT analysis of an orthographic camera, following a circular camera trajectory</td>
<td>107</td>
</tr>
<tr>
<td>7.8</td>
<td>Mapping a scene point onto a perspective camera, following a circular camera trajectory</td>
<td>109</td>
</tr>
</tbody>
</table>
7.9 ICT analysis of a perspective camera, following a circular camera trajectory ........................................... 110
7.10 Influence of the projector trajectory on the incoming illumination - circular, orthonormal projector case .................................................. 112
Chapter 1

Introduction

Contents

1.1 Problem Statement and Motivation .................. 2
1.2 Summary of Contributions ............................ 5
1.3 Overview of the Dissertation .......................... 5
1.1 Problem Statement and Motivation

In 1966, the Artificial Intelligence pioneer Marvin Minsky instructed one of his undergraduates to solve the problem of computer vision as a summer project. Now, more than 40 years later, this very same problem still remains unsolved, despite the huge amount of efforts that have been undertaken by people in the research community. Even though the general problem of visual understanding is widely regarded as not to be solved any time soon, researchers in this area have not stood by idly all these years. The larger problem has been subdivided into more easily manageable proportions, giving birth to many specialized subdomains. This has allowed researchers to tackle many practical problems in controlled environments, often with success. In this work, we will focus on one of the most basic, yet still one of the most interesting problems in the domain computer vision today: the task of shape acquisition.

Also commonly referred to as the 3D reconstruction problem, shape acquisition consists of the task of acquiring the exact shape or form of an object or scene. Capturing such structures is desirable for a wide variety of applications, and has been a vital component in many technologies, both today and in days past. E.g. the earliest practices in Western Europe date back to the days of the Bronze Age, where shape acquisition played an essential role in the mass production of tools, weapons, armor, and various building materials. After initially creating a master copy\(^1\) of the desired object in a pliable material (e.g. clay), a negative-image of the prototype was made in the form of a mold, effectively capturing the shape of the object. Subsequently, a liquid composition of smelted copper and tin\(^2\) ore could be poured into the mold, which after hardening produced a bronze instance of the desired object. Once a single mold was made (or multiple molds from the same master object), the object was ready for mass production. Even though the processes of molding and casting are still an essential parts of industrial production facilities today, modern day shape acquisition algorithms are more commonly focused on the task of quality control, rather than the capture of an initial prototype. Additionally, with the advent of the digital age, the technology behind the acquisition process has been rapidly evolving. As the storage of the captured 3D data is no longer restricted to real-world objects such as a mold, but can be contained on the hard disk of a computer, it has become possible to compare

\(^1\)In modern-day manufacturing, this initial prototype is most commonly referred to as the pattern.

\(^2\)The earliest bronze alloys were actually made of copper and arsenic. However, as this material is toxic to humans, its use diminished and eventually died out after metallurgists found an alternative in the form of tin.
1.1 Problem Statement and Motivation

the structure of a scanned object against the structure of a master copy already stored in the system. If the scanned product does not fall within predefined error bounds, it can be removed from the production pipeline in a fully automated fashion, without any required assistance from human elements.

Industrial inspection is of course not the only area of application for shape acquisition systems. Historically speaking, they have also played a role in the domains of archaeology (paleontology) and cultural heritage, serving the common goal of preserving artifacts that could otherwise be lost to the ages. Just like casting was one of the early techniques in industrial manufacturing, plaster casting was used as a means of reproducing famous sculptures as early as the 16\textsuperscript{th} century\textsuperscript{3}. Similarly, in paleontology plaster casts have been used as a way to preserve the shape of organisms whose original remains have been completely dissolved or otherwise destroyed, leaving behind only an organism-shaped hole in the rock. However, it is obvious that the casting process is not universally applicable to all historical artifacts or sites, and has one additional important disadvantage: it is a contact-based method, and as such it has the potential of altering or damaging the captured subject. During the period of the last two decades, modern day archaeologists have started looking for alternatives, adopting non-invasive, non-contact scanners not unlike those used in the manufacturing industry. One of the earliest examples of this trend, the Digital Michelangelo Project of Marc Levoy and his students at Stanford University [Levoy 00], employed a laser triangulation rangefinder to scan large statues such as the David of Michelangelo, after which the resulting range images were assembled into a single seamless polygon mesh. Additionally, techniques have been developed to scan entire sites instead of individual objects. One example of such a technique is the purely image-based 3D reconstruction method of Pollefeys \textit{et al.} [Pollefeys 01], which was developed for the purpose of reconstructing the archaeological site of Sagalassos in Turkey. The obtained data was useful to archaeologists in many ways, both as an instrument for hypotheses testing, and as a means to provide the public with a virtual tour of the site. Additionally, it would make an excellent candidate for augmented reality, combining the observed real-world data with (potentially annotated) virtual data. For the purpose of mixed/augmented reality, the shape acquisition step is required to provide a reference frame for the observed real-world data. If the observer’s position and orientation is known, the AR application knows what virtual data to submit to the viewer, overlaying the observed scene.

\textsuperscript{3}Leone Leoni, an Italian sculptor with international fame, assembled a collection of casts in Milan, collecting "as many of the most celebrated works, carved and cast, antique and modern as he was able to obtain anywhere". Such private collections, however, remained modest and uncommon until the 18\textsuperscript{th} century.
Archaeological sites are not the only areas for which it would be interesting to have a clear picture of the scene structure. In the field of mobile robotics, autonomous robots that are in unfamiliar terrain generally require a sense of their surroundings in order to perform the tasks they have been given. Depending on the task at hand, whether that may be exploring underwater caves in the deep ocean, or deep-space exploration of Mars, the basic questions remain the same: "What does the world look like?" and "Where am I?". Based on these two characteristic questions, a large body of work has been developed on SLAM (simultaneous localization and mapping) techniques. These methods continuously build up a map of the unknown environment, while at the same time keeping track of their current location.

Possibly the largest application area of shape acquisition algorithms today might be the entertainment sector. With the commercial breakthrough of 3D cinema, movies like *Beowulf*, *Avatar*, or *Alice in Wonderland* have significantly increased the number of base requirements for movies during the production process. Whereas knowledge of the 3D structure of objects (and actors) in the scene used to be required only in scenes that included special effects added later in post-production, it now is a base requirement for the coming generation of movies. In order to produce a realistic rendering of such scenes however, additional information is required about their structure. This information comes in the form of the reflectance properties of objects and actors alike, as it is imperative to know how light interacts with these subjects should we add additional synthetic elements to the scene, or modify existing ones from the captured footage. Whereas most simple objects can be described by a combination of absorption, diffuse and specular reflection, it is harder to capture and render objects that require more complex interactions such as refraction, subsurface scattering, diffraction, etc... As such this still remains an active research topic today. Historically speaking most of the advances in the domain of computer graphics reached the public through the movies, the market for 3D computer games has recently surpassed the box office as the top entertainment sector. Due to the ever increasing hardware support for realistic rendering in real-time, the use of detailed 3D models is steadily becoming commonplace in the industry. By scanning real-world objects, instead of having to let 3D artist create them by hand, valuable time can be preserved for other tasks.

The work presented in this dissertation is part of the recent research effort to capture and model objects with increasingly complex reflectance properties. To be more specific, our proposed shape acquisition method focuses on static objects and/or scenes with potentially complex reflectance properties, which we capture by the means of a controllable mobile illumination source.
1.2 Summary of Contributions

In this dissertation we present a novel class of active shape acquisition methods. Whereas the vast majority of all current structured light methods apply some kind of spatial and/or temporal coding scheme, so far the use of structured viewpoint changes in active stereo has been largely neglected by the computer vision community. By changing the viewpoint of the projector in predefined manner, we can add an additional cue to the scanning process.

In this work, we exploit this cue by translating the projector horizontally with respect to its image plane, in combination with the use of a static spatial pattern: a 1D wave pattern, like those commonly used in phase-shifting approaches. Unlike such spatial approaches however, the function of this pattern is not to label camera-projector correspondences. Instead, it is used to encode the structured viewpoint changes: the illumination reflected by any given scene point, observed from any given angle, forms an amplitude-modulated wave pattern. This modulated wave pattern \((a)\) is directly correlated with the distance between the scene point and the image plane of the projector, encoded by the wave frequency; and \((b)\) forms a direct sampling of (a one-dimensional slice of) the BRDF, encoded by the amplitude modulation. As such, we are the first to have casted depth estimation as a signal processing problem.

There are several advantages to the proposed approach. Due to the large data redundancy associated with the many different viewpoints, our method is robust in the presence of many factors that commonly form a problem for active shape acquisition methods (non-Lambertian surfaces, interreflections, occlusions, specular highlights, etc...), as long as there is a minimal amount of diffuse/Lambertian reflection present. Furthermore, because of the methods independence from the observer (the camera), all computations can be applied in parallel on a per pixel basis. This allows for the preservation of sharp edges.

Finally, we have discussed and illustrated the duality of the proposed method to the its passive stereo counterpart, epipolar plane image analysis, and as such have shown potential future directions for additional research.

1.3 Overview of the Dissertation

In this section we will provide a short chapter by chapter overview of the content within this dissertation.
In chapters 2 and 3, we will provide the reader with an overview of the related work in the field of shape acquisition in general, followed by a more in-depth discussion of the camera/projector 3D scanning systems in particular. We will provide the reader with a taxonomy of available techniques, and how our own work fits within this structure.

In chapter 4 we will provide the reader with an overview of the related work in the field of signal processing. As the technique proposed in this dissertation casts depth estimation as a signal processing problem, it is useful to provide a short overview of the related problems in this domain.

In chapters 5 and 6 we will introduce our own technique. Starting from the basic algorithm in chapter 5, as we have initially presented it to the computer vision community [Hermans 09b], we will improve on the initial idea by making use of the fact that we have cast depth estimation as a signal processing problem. As such, we will make use of the material covered in chapter 4. Finally, we will remove unwanted restrictions to our setup in chapter 6, by introducing an additional illumination pattern to the setup.

In chapter 7, we will analyze the dual relation between the techniques proposed in this dissertation and epipolar image plane analysis, and what we can learn from the future directions taken in its dual counterpart.
Chapter 2

Related Work - Shape Acquisition

Contents

2.1 Contact Scanners ........................................ 9
2.2 Non-contact Scanners, Passive ........................ 9
2.3 Non-contact Scanners, Active ........................ 11
Many problems in computer vision are characterized by a single simple question, e.g., for object recognition algorithms, the produced output is generally an answer to the question: "What is the object?", or just how likely it is that the observed object is part of a certain category; motion analysis (egomotion, optical flow, tracking) deals with question: "Where is the object / where am I going?"; event recognition systems tell us: "When did it happen?". Shape acquisition is no exception to this rule, as it deals with solving the 3D reconstruction problem: "What is the exact shape of the object?".

There are many different types of shape acquisition systems in existence today, but regardless of the method of choice, the final output of the algorithm is usually a point cloud of geometric samples representing the surface of the object. These points can then be used to extrapolate the shape of the object, converting the point cloud into a more continuous surface representation like parametric surfaces (NURBS or splines), subdivision surfaces, or triangle meshes [Alexa 01]. If color information was collected at each point, then the colors on the surface of the subject can also be determined. However, before we reach this discrete set of sample points, 3D scanners commonly first acquire the scanned surface information in the form of a range or depth map. This is the equivalent of a picture taken by a camera, if we replace the color information captured by the sensor by distance information acquired by the scanning device, as illustrated in Figure 2.1. This means that like cameras, such scanning devices have a limited field of view, and are only able to collect information about surfaces that are not occluded. As a result, a single scan will usually not produce a complete model of the subject. Multiple scans, from many different directions are required to obtain sufficient information about all sides of the subject. These scans have to be brought in a common reference system, a process that is usually called alignment, registration, or extrinsic calibration, and then merged to create a complete model. This whole process, going from a single depth map to an entire model, is known as the 3D scanning pipeline [Bernardini 02].

Depending on the required speed and accuracy of the application, there are many different possible shape acquisition systems which the user can choose from. At the highest level, there are two types of scanners: contact and non-contact scanners. Non-contact 3D scanners can then still be divided further into two categories: active and passive scanners. Passive scanners do not alter the scene as it is being captured, while active methods do influence the scene by controlling its illumination. In this chapter, we will provide a brief overview of the available options.
2.1 Contact Scanners

Contact 3D scanners, such as the coordinate measuring machines with a physical probe used in manufacturing, or the hand driven touch probes used to digitize clay models in computer animation industry, probe the object through physical touch. Even though such scanners can be very precise, by definition they require contact with the object being scanned. As a result, the act of scanning the object might modify or even damage it. This poses a very significant disadvantage when scanning delicate or valuable objects such as historical artifacts. In addition, the time required for a physical robot or human arm to perform a complete scan is relatively long, compared to other scanning methods. As a result, these systems are mainly used in practice when time constraints are non-critical, and very high precision is required.

2.2 Non-contact Scanners, Passive

Because of the apparent disadvantages of contact scanners, other approaches based on computer vision techniques instead of robotics have also been developed. We will begin our overview of these non-contact scanners by a discussion of the passive methods. These are 3D scanners that do not emit any kind of radiation themselves, but instead rely on measuring reflected ambient radiation (e.g. visible light). The most common passive 3D scanning setup is the stereo rig: two cameras facing the same direction, but placed slightly apart. By analyzing the minor differences between the images seen by each camera, it becomes possible to determine the distance at each point in the images [Scharstein 02]. More elaborate approaches, such as using a multi-camera setup [Seitz 06] or a moving camera with a known or unknown path [Bolles 87] can also be employed.
Feature point matches are not the only means available for reconstructing an object’s shape. In 1994, Laurentini [Laurentini 94] introduced the first shape-from-silhouette reconstruction technique, and its associated geometric entity, the visual hull. This class of passive reconstruction techniques assumes that for all images, the foreground object can be separated from the background, producing a binary silhouette image. Each of these silhouettes represents a 2D projection of a corresponding 3D foreground object and, assuming that all camera parameters are known, each silhouette defines a back-projected generalized cone that contains the actual object. The intersection of these cones is called a visual hull, a bounding geometry of the actual 3D object. In practice, visual hulls by themselves are rarely used as a means of creating 3D object scans. Using individual pixel matches, they are augmented by combining the classical passive stereo techniques with the boundary volume of the visual hull. As the most common matching method is a simple photo-consistency check, the product of these approaches are commonly referred to as photo hulls [Seitz 97].
2.3 Non-contact Scanners, Active

In a controlled environment, the use of active non-contact scanners becomes an increasingly interesting option. The term *active* refers to the fact that these scanners actively emit one or more kinds of radiation. By detecting the reflections, this enables them to probe an object or environment. Possible types of emissions used include light (visible or infrared), ultrasound or x-ray. There are various ways the reflected radiation can be used to determine depth, and each method gives rise to a different type of scanner.

The most basic of all active scanners is the *triangulation* based [Jalkio 85] laser scanner (see Figure 2.3). Using a single stationary camera and a laser emitter, with known positions and orientations, the depth of a point can easily be derived from its observed position in the camera screen. This technique is called triangulation because the laser dot, the camera, and the laser form a triangle. The length of one side of the triangle, the camera-to-laser distance, is known. The angle of the laser emitter corner is also known. The angle of the camera corner can be determined by looking at the location of the laser dot in the camera’s field of view. From this information, we can compute the intersection of the two other sides of the triangle, providing us with a depth estimate for the illuminated point. In most practical systems a laser stripe, instead of a single laser dot, is swept across the object to speed up the acquisition process. By using rotation and translation stages, depth estimates can be acquired for most of the points on a sufficiently diffuse object surface.

A *time-of-flight* [Cracknell 07] laser scanner (see Figure 2.4) is an active scanner that is based on timing the *round-trip time* of a single emitted pulse of light. As the speed of light is a known quantity, simply multiplying it with the round-trip time gives us twice the distance traveled between the point of origin and the scanned surface. The accuracy of a time-of-flight scanner depends on how precisely it is able to measure this round-trip time: it only takes \( \approx 3.3 \cdot 10^{-12} \) seconds for light to travel 1 millimeter. As a result, time-of-flight scanners are relatively inaccurate compared to other methods, with an accuracy in the order of several millimeters. Despite this disadvantage, time-of-flight sensors still have practical use, as they are capable of operating over very long distances, on the order of kilometers. As such, these scanners are suitable for scanning large structures like buildings or geographic features. At a rate of 10,000 sample points per second, low resolution scans can take less than a second, but high resolution scans - requiring millions of samples - can take up to several minutes. In case of a non-static scene, this will introduce distortions in the collected data, as each data point is sampled at a different time.
Figure 2.3: In triangulation based laser scanning, a single laser dot or stripe is used to probe the environment. Based on the known position and orientation of the laser emitter and the camera, it is possible to derive the depth of the illuminated point(s), depending on its location on the camera screen [Teutsch 07].

(a) LRB 20000C Long Range LRF  (b) Leica HDS-3000 Terrestrial LIDAR

Figure 2.4: Two examples of devices based on the principle of time-of-flight. (left) A laser range finder, used for geological and engineering surveying, … (right) A terrestrial LIDAR scanner, used to scan buildings, rock formations, …
X-ray computed tomography (CT) [Kak 01, Suetens 09] is an imaging method primarily used in medical applications. It produces a set of cross-sectional images, representing the X-ray attenuation properties of a scanned object from many different directions. Image formation of a cross-section is based on the following procedure: (a) production of the X-rays by an X-ray tube; (b) attenuation of the emissions by the scanned object; (c) measurement of the remaining emissions by a set of detectors on the opposing side. The object is scanned using a set of thin X-ray beams, covering the entire field of view (see Figure 2.5). This process is repeated for a large number of angles, yielding line attenuation measurements for all possible angles and for all possible distances from the center. Based on all these measurements, the actual attenuation at each point of the scanned slice can be reconstructed.

Consider the 2D parallel-beam geometry in Figure 2.5a, in which $\mu(x,y)$ represents the distribution of the linear attenuation coefficient of the object slice in the xy-plane. The X-ray beams make an angle $\theta$ with the y-axis. The unattenuated intensity of the X-ray beams is $I_0$. If we define a new coordinate system $(r,s)$ by rotating $(x,y)$ over the angle $\theta$, the corresponding measured intensity profile as a function of $r$ is given by:

$$I_\theta(r) = I_0 \cdot e^{-\int_{L_{r,q}} \mu(x,y) \, ds} = I_0 \cdot e^{-\int_{L_{r,q}} \mu(r \cdot \cos \theta - s \cdot \sin \theta, r \cdot \sin \theta + s \cdot \cos \theta) \, ds} \quad (2.1)$$

where $L_{r,q}$ is the line that makes an angle $\theta$ with the y-axis at distance $r$ from the origin. Subsequently, each intensity profile is transformed into an attenuation profile:

$$p_\theta(r) = -\ln \frac{I_\theta(r)}{I_0} = \int_{L_{r,q}} \mu(r \cdot \cos \theta - s \cdot \sin \theta, r \cdot \sin \theta + s \cdot \cos \theta) \, ds \quad (2.2)$$

where $p_\theta(r)$ is the projection of the function $\mu(x,y)$ along the angle $\theta$. Stacking all these projections results in a 2D dataset $p(r,\theta)$, called a sinogram (see Figure 2.5d). In mathematics, such a transformation of a function $f(x,y)$ into its sinogram $p(r,\theta)$ is called the Radon transform:

$$p(r,\theta) = R[f(x,y)] = \int_{-\infty}^{\infty} f(r \cdot \cos \theta - s \cdot \sin \theta, r \cdot \sin \theta + s \cdot \cos \theta) \, ds \quad (2.3)$$

Given the sinogram $p(r,\theta)$, the question is how to reconstruct the original attenuation distribution $\mu(x,y)$. The algorithm that is currently being used in most [Kak 01] applications of straight ray tomography is the so called (filtered) backprojection algorithm. Intuitively, this corresponds to the following algorithm: for a particular
Figure 2.5: The basic scanning procedure in X-ray computed tomography (CT). A set of lines is scanned covering the entire field of view. (a) The corresponding parallel-beam geometry with coordinate systems. The X-ray beams make an angle \( \theta \) with the y-axis, and are at distance \( r \) from the origin. (b) An intensity profile \( I_0(r) \) is measured for every view, defined by their angle \( \theta \). \( I_0 \) is the unattenuated intensity. (c) The attenuation profiles \( p_\theta(r) \), obtained by log-converting the intensity profiles \( I_\theta(r) \), are the projections of the function \( \mu(x, y) \) along the angle \( \theta \). (d) Stacking the set of 1D projections \( p_\theta(r) \) gives us a 2D dataset \( p(r, \theta) \), known as a sinogram [Suetens 09].

For discrete computations, this becomes:

\[
\begin{align*}
    b(x_i, y_j) &= B[p(r_n, \theta_m)] \\
    &= \sum_{m=1}^{M} p(x_i \cdot \cos \theta_m + y_j \cdot \sin \theta_m, \theta_m) \delta \theta 
\end{align*}
\]

The values \( (x_i \cdot \cos \theta_m + y_j \cdot \sin \theta_m) \) generally do not coincide with the discrete positions \( r_n \), thus interpolation is required. For each view, a projection line through each pixel is drawn. The intersection of this line with the detector array is then computed, and the corresponding projection value is calculated by interpolating the neighboring measured values. This procedure is called pixel-driven or voxel-driven backprojection with linear interpolation.
2.3 Non-contact Scanners, Active

Figure 2.6: **Functional magnetic resonance imaging (fMRI)** is a type of specialized radiological scan, which measures the hemodynamic response (change in blood flow) related to neural activity in the brain or spinal cord.

Although most common in medicine, CT is also used in other fields, such as nondestructive materials testing, reverse engineering, or to study biological and paleontological specimens. In some cases, these novel application areas require a finer resolution\(^1\) at which to operate, introducing the field of **microtomography**\(^2\). These scanners are typically used for small animals, biomedical samples, foams, composites, foods, microfossils, and other studies for which a very high level of detail is required. In recent years, the concept of Nanotomography (Nano-CT) has been introduced, allowing for scans in the nanometer range.

**Magnetic resonance imaging (MRI)** is primarily a medical imaging technique, most commonly used in radiology to visualize the internal structure and function of the body [Suetens 09]. MRI provides much greater contrast between the different soft tissues of the body than computed tomography does, making it especially useful in neurological, musculoskeletal, cardiovascular, and oncological imaging. Unlike CT, it uses no ionizing radiation, but uses a powerful magnetic field to align the nuclear magnetization of (usually) hydrogen atoms in water in the body. **Functional magnetic resonance imaging (fMRI)** is a type of specialized MRI scan. It measures the hemodynamic response (change in blood flow) related to neural activity in the brain or spinal cord of humans or animals (see Figure 2.6). Since the early 1990s, fMRI has come to dominate the brain mapping field due to its relatively low invasiveness, absence of radiation exposure, and relatively wide availability.

---

\(^1\)The required pixel sizes of the cross-sections are in the micrometer range.

\(^2\)Also known as Micro-computed tomography, Micro-CT, X-ray tomographic microscopy, XMT, etc. All of these names generally represent the same class of instruments.
Ultrasound imaging has been used in clinical practice (see Figure 2.7) for over more than half a century. It is noninvasive, relatively inexpensive, portable, and has an excellent temporal resolution. Imaging by means of acoustic waves is not restricted to medical imaging. It is used in several other applications, such as in the field of nondestructive testing of materials to check for microscopic cracks (e.g. in airplane wings or bridges), in sound navigation ranging (SONAR) to locate fish, in the study of the seabed, to detect submarines, and in seismology to locate gas fields. The basic principle of ultrasound imaging is simple. A propagating wave partially reflects at the interface between different tissues. If these reflections are measured as a function over time, information is obtained on the position of the tissue if the velocity of the wave in the medium is known. However, besides reflection, other phenomena such as diffraction, refraction, attenuation, dispersion, and scattering also play a role when ultrasound propagates through matter. Just like in the case of the more conventional light waves, these phenomena can be a source of difficulties, and require adequate compensation. Ultrasound imaging is used not only to visualize morphology or anatomy, but also to visualize function by means of blood and myocardial velocities. The principle of velocity imaging was originally based on the Doppler effect and is therefore often referred to as Doppler imaging. A well-known example of the Doppler effect is the sudden pitch change of a whistling train when passing a static observer. Based on the observed pitch change, the velocity of the train can be calculated.
An alternative approach to probing the scene by emitting a single ray or stripe of coherent laser light consists of making use of a so called structured light method (see Figure 2.8). These methods [Jordi 04] employ a projector or a multi-stripe laser (instead of a one-dimensional stripe laser) to illuminate the scene, which significantly increases the total potential volume that can be scanned. By applying a smart codification of the emitted pattern, it becomes possible to create a one-to-one mapping between projector pixel locations and camera pixel locations. Using these point correspondences, and the available calibration data, it becomes possible to triangulate the corresponding 3D position of the illuminated surface point.

There are many different structured light methods available today. The main difference between all these methods is the choice of the used pattern codification strategy. The most important spatial, temporal, and viewpoint codification strategies will be the subject of the next chapter, in which we will discuss their strong and weak points, and compare them to our own method.

---

3In case the reader is unfamiliar with the concepts of intrinsic and extrinsic camera calibration, we refer to the excellent book by Hartley and Zisserman [Hartley 04].
Chapter 3

Related Work - Camera/Projector 3D Scanning Systems

Contents

3.1 Why use active illumination? .......................... 20
3.2 Temporal Encoding ................................. 22
  3.2.1 Binary Codes ................................. 22
  3.2.2 N-ary Codes ................................. 25
  3.2.3 Phase shifting ............................... 28
3.3 Spatial Encoding ................................. 31
  3.3.1 De Bruijn Sequences and M-Sequences .......................... 32
  3.3.2 M-Arrays ................................. 35
3.4 Spatio-Temporal Encoding .......................... 37
  3.4.1 Spacetime Stereo .......................... 37
  3.4.2 Unstructured Light .......................... 38
3.5 Direct Encoding ................................. 39
  3.5.1 Grey Levels ................................. 39
  3.5.2 Color Coding ............................... 41
3.6 Viewpoint Encoding ............................... 42
3.7 Focal Encoding ................................. 44
  3.7.1 Depth from Camera (De)Focus .................. 44
  3.7.2 Depth from Projector Defocus ................. 45
Related Work - Camera/Projector 3D Scanning Systems

Figure 3.1: Camera/projector 3D scanning. (a) Structured light setup, showing the digital camera mounted on a translation stage and the video projector; (b) One of the scenes being acquired; (c) An example of one of the scenes, through the camera lens; (d) The corresponding depth map. [Scharstein 03]

In the previous chapter, we have provided an overview of the many different shape acquisition systems in existence today. We have discussed both the advantages and the disadvantages, together with their usual target applications. In this chapter, we will focus on one particular subset of methods: camera/projector based 3D scanning systems, also know as structured light.

3.1 Why use active illumination?

The classical approach to multi-view stereo - known as structure from motion - is a passive feature-based approach. Such methods typically match a sparse set of image features\(^1\), observed by two or more cameras [Scharstein 02, Seitz 06], a moving camera with a known camera path [Bolles 87], or a moving camera with an unknown camera path [Sturm 96, Pollefeys 04]. Using the estimated calibration data, it becomes possible to triangulate\(^2\) the depth information for each set of feature points. There are many different instances of such stereo algorithms in existence today, yet they all typically suffer from similar problems, caused by feature matching difficulties. Typical difficulties for this class of algorithms include uniform regions, depth discontinuities, specular highlights and other more complex BRDFs.

---

\(^1\)For the remainder of this dissertation, we shall assume that the image features come in the form of feature points. The main advantage of using points is our ability to attach a descriptor to each point. During the matching stage, these descriptors are used to compare points from different views to establish feature correspondences. The most basic descriptor of a point is a simple \(N \times N\) window around the pixel location, which is matched by minimizing the sum of squared differences (SSD) over all pixels. More complex descriptors, such as SIFT [Lowe 99] were developed for wide-baseline setups, in which the descriptor of a chosen point needs to be invariant to rotations, translations and scale.

\(^2\)In practice, some form of error function is minimized, e.g. the reprojection error, or the distance of the point to each backprojected ray, or a computationally more efficient cost function such as the Sampson error. For more information, we refer to the book of Hartley and Zisserman [Hartley 04].
3.1 Why use active illumination?

Feature-based stereo algorithms are hugely dependent on their ability to (a) find sufficient features with good descriptors, and (b) match these features over two or more frames. Regardless of the method of choice, the output of such an algorithm will always be a (relatively) sparse set of feature points on the object’s surface. For some applications however, it is desirable to recover a dense depth map of the scene. By providing the algorithm with a discrete set of possible depth labels, it is able to evaluate these depths on a per pixel basis using an implicit or explicit cost function. For each pixel, the depth label with the minimal cost is assigned. The resulting depth map is commonly refined using some global optimization algorithm. If we have a calibrated stereo rig, which provides us with rectified stereo pairs, the set of possible depth labels is defined in disparity space. In case of multi-view stereo however, it is more common to take a plane-sweep approach. In these approaches, named after the seminal work of Collins [Collins 96], a discrete set of depth planes is defined at specified distances from one central camera for which the depth map must be computed. For each possible depth value, the observed camera intensities in each camera are projected back to corresponding depth plane in space. For each pixel of the virtual camera, the color correspondence forms the basis for the cost function: the depth value for which the color dissimilarity is minimal is assigned to the pixel. Throughout the years, many extensions to this approach have been presented and implemented. Because of the per pixel nature of the computations, it lends well to parallelization, and as such it has been implemented for real-time applications on commodity hardware [Yang 02, Yang 03, Dumont 08]. Interesting surveys on this material are provided by Scharstein et al. [Scharstein 02] and Seitz et al. [Seitz 06].

An alternative to detecting corresponding image features is creating easily distinguishable features using controlled illumination (usually in the form of a projector, see Figure 3.1), assigning a unique codeword to each projector pixel (or in some cases: sub-pixel) location. This allows for a more robust labeling, especially within previously uniform regions. This labeling comes in a variety of forms [Jordi 04], from projecting a single spatially encoded pattern [Boyer 87, Batlle 98] such as De Bruijn codes [Vuylsteke 90], to a temporal encoding scheme such as binary [Posdamer 82] or Gray codes [Inokuchi 84, Scharstein 03], a combination of both spatial and temporal encoding [Davis 03, Davis 05], or viewpoint encoding [Young 07]. Methods using a single pattern typically yield very fast - but lower quality - approximations, whereas slower methods that employ multiple patterns produce higher quality acquisitions. However, both types are generally sensitive to specular outliers, have problems with global light transport (i.e. subsurface scattering and interreflections), or encounter difficulties at depth discontinuities. Problems due to the global light transport can be reduced,
by employing high frequency illumination modulation [Nayar 06, Zhang 06b, Chen 08], light polarization [Wolff 89, Chen 07] or immersing the scene in a fluorescent dye [Hullin 08]. Recently, Gupta et al. [Gupta 09] have proposed a method to estimate the global illumination, assuming the absence of camera defocus. We will inspect these different approaches in the following sections.

3.2 Temporal Encoding

The most commonly applied strategies in active acquisition are based on temporal encoding. In these algorithms, a set of patterns is successively projected onto the scanned surface. The codeword for a given projector pixel is formed by the sequence of intensity values for that pixel across the projected patterns. Thus, because the bits of the codewords are multiplexed in time, this codification scheme is commonly referred to as temporal encoding. These algorithms generally produce high accuracy measurements. This is mainly due to two factors: (a) since multiple patterns are projected, the codeword basis tends to be small and therefore a small set of primitives is used, being easily distinguishable among each other; (b) a coarse-to-fine paradigm is followed, as the position of a pixel can be increasingly refined as more patterns are successively projected.

During the last thirty years many techniques based on time-multiplexing have appeared. We have classified these techniques as follows: (a) techniques based on binary codes: a sequence of binary patterns is used in order to generate binary codewords; (b) techniques based on n-ary codes: a basis of n primitives is used to generate the codewords; (c) phase shifting: the same pattern (usually a sine wave) is projected several times, shifted it in a certain direction. In the remainder of this section, we shall describe these techniques in more detail.

3.2.1 Binary Codes

In this category of techniques only two illumination levels are used. Every pixel of the pattern has its own codeword, defined by a sequence of zeros and ones, corresponding to its location in each of the projected patterns. Thus, a codeword for an observed pixel is obtained once the entire sequence has been projected. In case the calibration information of both the camera and projector are known, it is sufficient to only encode a single projector axis, as the corresponding depth information can be derived from the epipolar geometry constraints. Otherwise, the x and y positions need to be estimated separately, by emitting a sequence of horizontal and vertical stripe patterns.
3.2 Temporal Encoding

Figure 3.2: **Gray code patterns** (left) versus **binary patterns** (right). In case of Gray code patterns, an observed intensity reflects a pattern edge for only a single pattern refinement level, as the black/white edges will vary at each level. This is not the case for binary patterns. Once the observed intensity lies on a binary pattern edge, it will fall on that edge for every subsequent pattern [Francken 10].

In 1982, Posdamer and Altschuler [Posdamer 82] proposed the projection of a sequence of \( m \) patterns to encode \( 2^m \) horizontal pixel positions (stripes), using a straightforward binary code. Thus, the codeword associated with each pixel is the sequence of zeros and ones obtained from the \( m \) patterns, in which a lack of intensity corresponds to a 0, and a fully illuminated pixel corresponds to a 1. The number of stripes increases by a factor of two at every consecutive pattern, and defines the number of unique binary codewords. The maximum number of patterns that can be projected is limited by the resolution of the projector. However, it is generally not recommended to use the maximal amount of available stripes, as the camera might no longer be able to distinguish between illuminated and non-illuminated pixels\(^3\). As such, stripes are commonly encoded with a width of several pixels. As all pixels belonging to a single stripe in the highest frequency pattern all share the same codeword, it is necessary to calculate either the center of every stripe, or the edge between two consecutive stripes, before triangulation is performed. Practical experience has shown the latter to be the best choice [Jordi 04].

\(^3\)This could be due to a variety of reasons: (a) because the highest frequency pattern might get blurred by defocus; (b) the resolution of the camera is insufficient to distinguish between two different stripes, resulting in a single blurred pixel; (c) aliasing effects occur.
Inokuchi et al. [Inokuchi 84] improved the codification scheme of Posdamer and Altschuler by introducing Gray codes instead of the plain binary codification scheme. The advantage of using Gray codes over their binary counterparts lies in the fact that codewords of neighboring projector pixels belonging to different stripes\(^4\) have a Hamming distance\(^5\) of one, making the technique more robust against noise. This means that a surface point illuminated by a series of Gray code patterns risks of being illuminated by a stripe edge only once, whereas if it were illuminated by a series of binary patterns, it would fall on an edge for every subsequent pattern after the first refinement level containing the edge. This is illustrated in Figures 3.2 and 3.3.

The concept of using binary or Gray code patterns has not been restricted to the use of structured light using projectors. In related work, not part of this dissertation, we have investigated the use of a computer display - or any other rectangular grid of point light sources - as a planar illuminant, for the purpose of performing meso-structure acquisition. While the main goal of most structured light systems (including the work in this dissertation) is to acquire the macro-structure of the scanned object, meso-structure acquisition aims at recovering the fine detail of the scanned surface. Binary patterns can be used (a) to calibrate such a camera-display setup [Francken 09], and (b) as a means of encoding surface normals [Francken 08a]. However, the range of suitable patterns for this application is not restricted to binary patterns only, and the choice of the used pattern depends on the nature of the scanned material[Francken 07, Francken 08b, Francken 08c, Tarini 05].

\(^4\)More specifically, neighboring stripes at the highest refinement level.
\(^5\)In information theory, the Hamming distance between two strings of equal length is the number of positions at which the corresponding symbols are different.
3.2 Temporal Encoding

3.2.2 N-ary Codes

In binary coding, the fact that only two intensities are used facilitates the segmentation process on the capturing side. However, the main drawback of these schemes is the relatively large number of required patterns. There has been some work done on solving this problem, by means of increasing the number of intensity levels used to encode the stripes. In this section, we will present the two different types of approaches that have developed over the years.

Color Encoding

Caspi et al. [Caspi 95] have proposed a multi-level Gray code based on color encoding. This variant on the Gray code scheme employs an alphabet of \( n \) symbols, where every symbol is associated with a certain (RGB) color. As the alphabet size increases, the number of required patterns is reduced. E.g. while \( m \) patterns are necessary to encode \( 2^m \) stripes for binary codes, the use of \( n \)-ary (Gray) codes allows for the codification of \( n^m \) different stripes using the same number of patterns.

It is important to note that this approach is effectively a generalization of the widely used binary codes. E.g. like their binary counterparts, \( n \)-ary codes have a fixed Hamming distance of one between consecutive codewords. Caspi et al. not only developed the mathematical basis for generating such \( n \)-ary codes, but they also presented an illumination model that could be used when applying such a structured light system. This model takes into account the light spectrum of the LCD projector, the spectral response of a 3-CCD camera, and the reflectance properties of the scanned surface:

\[
\begin{bmatrix}
R \\
G \\
B \\
\end{bmatrix}
= 
\begin{bmatrix}
\begin{bmatrix}
ar_r & a_{rg} & a_{rb} \\
ag_r & ag_g & a_{gb} \\
ab_r & ab_g & ab_b \\
\end{bmatrix} & k_r & 0 & 0 \\
0 & k_g & 0 \\
0 & 0 & k_b \\
\end{bmatrix}
\begin{bmatrix}
r \\
g \\
b \\
\end{bmatrix}
+ 
\begin{bmatrix}
R_0 \\
G_0 \\
B_0 \\
\end{bmatrix}
\]

(3.1)

where \( \vec{c} \) is the projected intensity value for a given stripe, \( P \) is the non-linear transformation from projection instruction to actually projected intensities for every RGB channel. \( A \) is the projector-camera coupling matrix. Every element \( a_{ij} \) of matrix \( A \) is the convolution of the camera spectral response for channel \( i \) with the spectrum of the light projected in channel \( j \). Matrix \( A \) shows the crosstalk between color channels. \( \vec{C} \) is the vector containing the RGB camera readings of a certain pixel. \( \vec{C}_0 \) is the camera readings corresponding to the scene under ambient lighting. Finally, \( K \) is the surface reflectance matrix specific for every scene point projected into a camera pixel. This matrix contains a reflectance constant for every RGB channel.
The main contribution of this model is that it incorporates a constant reflectance value for every scene point in the three RGB channels. This was the first work not to assume color neutrality of the scene, an assumption which was pretty common in most earlier systems dealing with color coding schemes [Boyer 87, Hgli 88, Chen 97]. In the case we are assuming color neutrality, matrix $K$ becomes the identity matrix for every pixel.

In order to compute the different terms of eq. (3.1), it is necessary to perform an initial color calibration step. Doing this will provide us with the values of $A$ and $P$. $P$ is a non-linear function, but it is invertible, so it can be implemented by a look up table for each color channel. From this information, we can compute the matrix $K$ by taking a reference image under white illumination ($\forall i: \vec{c}_i = [1, 1, 1]^T$) and solving for eq. (3.1). The proposed illumination model allows for the projected color to be estimated directly from camera readings, and it is generally applicable to any system dealing with color.

**Intensity Encoding**

Another technique that encodes adjacent stripes with $n$-ary codewords, proposed by Horn and Kiryati [Horn 99], uses multiple grey levels instead of a binary alphabet as the basis of their codes. The aim of their work was to find the smallest set of patterns that meet the accuracy requirements of a certain application, producing the best performance under certain noise conditions.

Given an alphabet of $n$ symbols, in which all elements correspond to a set of equidistant grey levels, their technique produces a code in which consecutive codewords have a Hamming distance of one. By increasing the basis of the code and maintaining the length, more codewords can be generated at the cost of decreasing the intensity distance between consecutive grey levels. The authors proposed the use of space filling curves such as Hillbert or Peano curves [Sagan 94] as a basis for generating such a code. These curves represent a path in an $n$-dimensional space, passing through a point set in which consecutive points are joined by straight segments of a equal length. When generating a codification scheme using such curves, there exists a trade off between the number of stripes, the number of projected patterns, and the distance between codewords (which is proportional to noise immunity factor of the system). For a given interpoint distance, if the number of stripes must be increased, then the length of the space filling curve must also grow.

This can be achieved by either increasing the curve order, which has the problem of reducing the distance between consecutive codewords (since the distance between consecutive points on the curve also reduces), or increasing the dimension of the curve, i.e. increasing the number of projecting patterns.
Figure 3.4: **Intensity encoding using Hillbert curves.** (a) A 3-D 2\textsuperscript{nd} order Hillbert curve with 128 codewords placed on it for n-ary codification; (b) Intensity profiles of the patterns encoded with the codewords extracted from the Hillbert curve; (c) The resulting n-ary patterns [Horn 99].

Horn and Kiryati tested their system with a 3D Hillbert 2nd order curve, using 256 different codewords. This configuration consisted of 3 patterns and 13 grey levels, and outperformed a binary Gray coding scheme using 9 patterns with 512 encoded stripes, both in terms of speed and accuracy.
Figure 3.5: **Phase shifting.** (a) Stereo camera setup proposed by Lilienblum and Michaelis [Lilienblum 07]. (b) Each recorded camera image of the projected sinusoidal pattern can be converted into a corresponding (relative) phase image. (c) By a careful choice of projected fringe patterns, it is possible to combine multiple relative phase images into an absolute phase image. (d) Schematic representation of how to perform sub-pixel 3D measurements using known phase information.

### 3.2.3 Phase shifting

Patterns based on binary or n-ary codes have the advantage that the pixel codification process occurs at a per pixel basis, in which no spatial neighborhood has to be considered. However, the discrete nature of these patterns limits the potential resolution of their range. **Phase shifting** methods on the other hand work at a higher spatial resolution [Strutz 93, Quan 01, Wiora 01, Wang 06]. In these approaches, a periodic intensity pattern (usually a one-dimensional sinusoidal fringe pattern) is repeatedly projected onto the scene, slightly shifted at every projection with a known constant. By determining the relative phase of the observed intensity values, based on their M-step\(^6\) relationships [Surrel 96], we are able to pinpoint their corresponding location on to the observed wave period. This is illustrated in Figure 3.5b.

\(^6\)M is a algorithm-dependent constant that indicates the number of samples that are required by the technique to accurately determine the phase shift, and thus the corresponding depth value.
3.2 Temporal Encoding

Phase Unwrapping

There is however the clear problem of inherent ambiguity related to this class of methods. The periodicity of the observed fringe pattern is directly related to the periodicity of the computed phase values. In essence, from a set of \( m \) shifted fringe patterns\(^7\), we are only able to recover the observed point’s relative phase (see Figure 3.5b). Thus we still need to disambiguate the mapping of the observed phase onto a (sub-)pixel fringe pattern position. This is a process commonly referred to as phase unwrapping.

There are many ways in which people have tried to tackle this problem. If we already have approximate depth estimates [Wahl 84, Sato 85, Bergmann 95], we might be able to restrict the range of potential depth estimates to values within a single wave period of the projected fringe pattern. A disadvantage of combined solutions based on an additional method is that these approximate scans are (a) a source for additional errors, and (b) generally increase the measuring time, but not the measuring accuracy. An alternative to making use of approximate values is the calculation of the absolute phase measurements. The easiest method to achieve this, called spatial phase unwrapping [Baldi 02, Su 04], removes discontinuities in neighboring phase values by the addition or subtraction of suitable multiples of 2. This approach works well, but only for smooth surfaces. Surfaces with strong discontinuities propose a more difficult challenge [Huntley 93, Huntley 97]. A more general method is the projection of multiple phase shift sequences, which differ only in their local period. There are two categories: (a) the hierarchical approach, or (b) the number-theoretical approach. The hierarchical approaches [Nadeborn 96, Chen 07] have problems similar to those of the gray code solution, as the phase shift sequences with low frequency do not increase the accuracy of the measurement. The number-theoretical approaches [Gushov 91, Zhong 01, Lofdahl 01] avoid this problem through the use of equivalent frequencies. Recent works [Lilienblum 07, Weise 07] have focused on increasing the robustness of these approaches, while trying to reduce the complexity of the algorithm. Using compound phase coding\(^8\) [Albarelli 09], number-theoretical currently achieve interactive speeds.

---

\(^7\)The choice of the number of used patterns \( m \) initially gave rise to a wide variety of methods [Creath 86, Schmit 95, Zhang 99], until more general approaches dealing with an arbitrary number of images were developed [Surrel 96].

\(^8\)The main idea behind the Compound Phase Coding strategy is to project several fringe patterns in a single spatiotemporal pattern. This is obtained by encoding the phases of the fringe vector as phases of a Fourier term at different frequencies.
Surface Reconstruction

The actual surface reconstruction can then take place using triangulation methods [Faugeras 93, Jarvis 93]. In classical phase shifting, a setup with one fringe projector and one camera device is commonly used. However, because inaccuracies in the camera/projector calibration are an unwanted extra potential source of error, Lilienblum and Michaelis [Lilienblum 07] recommend a setup with a projector and two cameras instead (see Figure 3.5a). In case of two cameras the phase values are used to solve the correspondence problem [Batlle 98, Hartley 04]. This can be done at sub-pixel accuracy, as illustrated in Figure 3.5d.

The Influence of Projector Blur

One rarely mentioned in literature [Jordi 04], but nevertheless very interesting, note about phase shifting methods is the fact that it does not suffer from problems due to projector blur. To be more precise, the accuracy of the phase estimation actually increases when we emit a blurred fringe pattern!

To illustrate this, we refer to Figure 3.6, in which we can see the influence of a Gaussian blur kernel on the projection of the discrete emitted fringe pattern. The illumination from a projector device is a discrete sinusoidal curve (blue). It is obviously an approximation of the intended continuous sinusoidal curve (black). If we blur the projector however, this is approximately equivalent to convolving the discrete curve with a Gaussian blur kernel. In the illustrated example, we have discrete projector pixels with a width of 20 units, convolved with a Gaussian blur kernel of 49 units, approximately two and a half pixels. As we can see, the convolved curve is clearly a better approximation of the continuous curve than its discrete counterpart. In order to see just how much better, we also plotted the corresponding error curves. We can draw several conclusions from the analysis of Figure 3.6:

- Blurring a discrete sine wave does not influence its phase or period.
- Blurring a discrete sine wave results in a better approximation of the continuous sine wave it represents.

Both properties are important for phase shifting algorithms, who actually benefit from a slight blurring of the projected pattern. The first property is also important to our own method, as we also project a discrete sine wave, but are more interested in recovering its period. The only downside to blurring the pattern is the resulting loss of contrast: the amplitude of the sine wave decreases as the size of the blur kernel increases.
3.3 Spatial Encoding

As we have mentioned earlier, temporal codification is commonly used when high precision is required by the application. However, sometimes it is desirable that we are able to label each point in the scene, based on a single recorded image. Thus, the need arises to concentrate all the codification in a single unique pattern. The codeword that corresponds to a certain point of the pattern is obtained from the spatial neighborhood of points around it. As a result, we refer to this class of algorithms as spatial encoding techniques.

Figure 3.6: The influence of projector blur. (a) The illumination from a projector device is a discrete sinusoidal curve (blue). It is obviously an approximation of the intended continuous sinusoidal curve (black). If we blur the projector however, this is approximately equivalent to convolving the discrete curve with a Gaussian blur kernel. As we can see, the convolved curve is a better approximation of the continues curve than its discrete counterpart. (b) In order to see just how much better, we also plotted the corresponding error curves.
3.3.1 De Bruijn Sequences and M-Sequences

In combinatorial mathematics, a \( n \)-ary De Bruijn sequence \( B(n,m) \) of order \( m \) is a cyclic sequence of a given alphabet \( A \) of size \( n \), for which every possible subsequence of length \( m \) in \( A \) appears as a sequence of consecutive characters exactly once \([\text{de Bruijn 46}]\). As such, every position in a \( B(2,m) \) De Bruijn sequence is directly linked to a unique binary subsequence of length \( m \). A *pseudorandom sequence* or *M-sequence* is similar to a De Bruijn sequence, except that it does not contain not contain the substring formed by only zeros \([\text{MacWilliams 76}]\), and thus has length of \( n^m - 1 \).
3.3 Spatial Encoding

![De Bruijn patterns used for spatial encoding](image)

(a) Pattern proposed by Zhang et al. [Zhang 02].  (b) Pattern proposed by Monks and Carter [Monks 93].  (c) Pattern proposed by Sagawa et al. [Sagawa 09].  (d-e) Pattern proposed by Pages et al. [Pages 04, Pages 05]. The color pattern (d) is suitable for locating edges in the image, while the luminance channel of the same image (e) is suitable for locating intensity peaks.

**De Bruijn Sequence Generation**

A De Bruijn sequence can be obtained by searching Eulerian or Hamiltonian circuits\(^9\) over different kinds of De Bruijn graphs\(^10\) [Fredricksen 82]. Gathering the edge labels of an Eulerian circuit, a De Bruijn sequence of order \(m\) can be obtained (e.g. 10001011101001). If a Hamiltonian circuit is followed over the same graph, we obtain a De Bruijn sequence of order \(m - 1\) (e.g. 00101110). Unfortunately, obtaining a Hamiltonian or Eulerian circuit from a given graph is an NP complete problem, and as such we want to look for alternatives that run in linear or polynomial time instead.

---

\(^9\)An Eulerian circuit is a path which both starts and ends in the same vertex, passing through all the edges exactly once. A Hamiltonian circuit is a path which both starts and ends in the same vertex, passing through all the vertices exactly once.

\(^10\)In a De Bruijn graph, all the words of length \(m - 1\) over a given alphabet \(A\) correspond to a single vertex in the graph. An example of such a graph (in which \(m = 4\)) is shown in Figure 3.7.
Pseudorandom or M-Sequence Generation

Whilst it is possible to generate higher order De Bruijn sequences using the methods described in the previous section, in practice it is often desirable to make use of M-sequences instead. The motivation behind this is that these pseudorandom sequences can be generated using very fast and easy algorithms, known as Linear Feedback Shift Registers [Golomb 81]. Starting from a random $m$-bit pattern, excluding the state in which all bits are equal to zero, each next bit can be generated as a linear function of its previous state. We can continue this process until we have cycled through all states except the all-zero state.

Algorithms

A major group of spatial encoding techniques define multi-slit [Maruyama 93] or stripe [Monks 93, Zhang 02] patterns in order to obtain dense reconstructions. The former type of patterns is suitable for locating intensity peaks\(^{11}\) in the image while the latter is aimed to locate edges. In 2004, Pages et al. [Pages 04, Pages 05] introduced an interesting new way to design colored stripe patterns so that both intensity peaks and edges can be located without loss of accuracy, and reducing the number of hue levels included in the pattern.

Recently, Sagawa et al. [Sagawa 09] proposed a technique based on a one-shot scanning method, using a dense and simple grid pattern. Their approach aims to avoid instabilities due to the use of multiple colors, and the influence of object color and texture. In order to achieve this, they minimize the number of used colors in the De Bruijn pattern by weakening the uniqueness constraint to a local area. After performing a robust line detection step, based on belief propagation, their method is able to perform a fairly good reconstruction of the scene, using both the geometric constraints of line patterns with structures of pencils of planes, and the information from the De Bruijn sequence. The main strength of this algorithm is its high speed. In their experiments, they were able to successfully captured the shapes of an exploding balloon, and a breaking ceramic dish at 300 - 1000 fps.

\(^{11}\) Based on the observation that intensity profile of imaged lines commonly follows a Gaussian distribution, one commonly used approach is to use the peak detector of Blais and Rious [Blais 86]. This detector convolves a linear derivative filter at each pixel of every image row (when treating vertical multistripe patterns). For each row, a set of local maxima and minima indicating the transitions from black to white regions and vice versa is obtained. Afterwards, for each pair of consecutive maxima and minima, the zerocrossing of the linear interpolation between them is calculated, and as such the sub-pixel position of the intensity peak is obtained. For a comparison with other peak detection algorithms, we refer to Fisher and Naidu [Fisher 96].
3.3 Spatial Encoding

3.3.2 M-Arrays

The idea of using a pseudorandom spatial codification of the projected pattern onto the scene is not restricted to a single dimension. The idea behind M-sequences can be extended to the 2D case, introducing the concept of m-arrays.

Let $M$ be a matrix of dimensions $r \times v$, in which each element is part of a given alphabet $A$ of size $k$. If $M$ has the window property, i.e. each different submatrix of dimensions $n \times m$ appears exactly once, then $M$ is defined as a perfect map. If $M$ contains all submatrices of $n \times m$, except the one filled by zeros, then $M$ is called a pseudorandom array or an M-array\textsuperscript{12}. For more information on pseudorandom sequences and arrays, we refer to the works of MacWilliams [MacWilliams 76] and Etzion [Etzion 88]. These arrays have been widely used in pattern codification, as the window property relates every different submatrix to an absolute position in the array.

Algorithms

The main differences between the techniques in this category is the way in which the elements of the M-array are represented in the projected pattern (see Figure 3.9). Some authors [Davies 98, Morano 98] prefer to define the pattern as an array of colored spots, in which each color represents one of the symbols of the coding alphabet. Other authors [Griffin 92, Spoelder 00] prefer to assign different shapes to each symbol. Since a spatial neighborhood is used, there will be parts of the projected pattern that will not be visible to the camera due to shadows and occlusions. The robustness of these methods depends greatly on the correct decodification of the visible parts, taking advantage of the properties of the M-arrays. Some authors prefer to increase the robustness of their method by projecting additional patterns onto the scene, in order to ease the segmentation stage of their algorithm, or to perform some kind of intensity or color normalization. However, we feel that while this increases their robustness, it defeats the purpose of using a purely spatial pattern like a De Bruijn pattern in the first place, as it limits the use of these methods to static scenes.

The usefulness of an M-array codification scheme is not restricted to applications in structured light. Recently, Scholz et al. [Scholz 05, Scholz 06] used an M-array codification scheme to label individual texture coordinates on garments. While observing the scene by one [Scholz 06] or more [Scholz 05] cameras, their method is able to associate each patch of the encoded piece of clothing with a given texture coordinate, enabling the user to cover the encoded surface with an arbitrary texture, as is shown in Figure 3.10.

\textsuperscript{12}Notice the analogy with De Bruijn sequences, which only differ from M-sequences by the fact that they also include the all-zeros state.
Figure 3.9: **M-array representations.** (a) An M-array based on 3 symbols, represented by three shape primitives [Griffin 92]; (b) Generation of M-arrays represented by an array of colored spots [Morano 98].

Figure 3.10: **M-Arrays used for spatial encoding.** Each window of colored dots represents a unique codeword, corresponding to a single texture coordinate. This enables the user to cover the encoded garment with an arbitrary texture [Scholz 06].
Figure 3.11: Spacetime stereo. By using spatio-temporal matching windows, a calibrated stereo rig is able to create feature correspondences from which to triangulate depth [Davis 03].

3.4 Spatio-Temporal Encoding

So far, we have only considered techniques that assign either a spatial or temporal codeword to each individual camera pixel, after which triangulation gives us a corresponding depth value. However, nothing prevents us from using both the spatial and the temporal dimension to create a codebook for our scene.

3.4.1 Spacetime Stereo

The spacetime stereo framework [Davis 03, Davis 05, Zhang 03] was introduced as an extension of the classical passive stereo algorithms, in which a spatial $W \times H \times 1$ matching window is used to compute feature correspondences between two camera frames, after which triangulation takes place to recover the depth. Alternatively, it can also be seen as an extension of classical structured light systems, if we look at the pattern projector in these systems as a dual camera projecting $1 \times 1 \times T$ matching windows\(^\text{13}\). In spacetime stereo, $W \times H \times T$ matching windows are used instead to establish correspondences. Using this expanded the matchable information space, authors [Davis 03, Davis 05, Zhang 02, Zhang 03] have tried expanding the range of their applications, e.g. to include the sampling of dynamic scenes [Rusinkiewicz 02, Davis 03, Zhang 03] and animation [Zhang 04].

\(^{13}\)This is of course in case of a classical temporal encoding scheme. In the case of a spatial encoding scheme, such as De Bruijn codes, the window is defined in the spatial domain, similar to the passive stereo techniques.
3.4.2 Unstructured Light

In the previous techniques, the produced light patterns were all projected onto the scene by a projector, part of a carefully calibrated camera/projector stereo rig. Alternatively, it is possible to make use of the naturally occurring illumination in the scene. One recent example in an underwater environment can be found in the work of Swirski et al. [Y. Swirski 09], which make use of the naturally occurring variation in refracted sunlight known as underwater caustics or flicker. The temporal radiance variations due to flicker are unique to each object point, thus providing us with a unique codeword. However, since in this application the illumination source is not a controllable projector part of a camera/projector stereo rig, the use of another camera is required to perform the triangulation step.

Figure 3.12: Stereo from flickering caustics. (a,b) Left and right frames at one instance in the sequence. (c) Temporal intensity plots of the marked pixels in the left and right frame. (d) The estimated disparity map. (e) The scene, generated from a different viewpoint using texture mapping [Y. Swirski 09].
3.5 Direct Encoding

In the previous sections, we have introduced methods that encode the projected pixels either in the temporal (over a sequence of images) and/or the spatial dimensions (over a unique neighborhood). Alternatively, some methods attempt at containing the codeword in a single unique pixel, creating a *direct* one-on-one mapping between the observed codeword and the corresponding pixel within the projected pattern.

In order to achieve this, it is necessary to either use a large range of color values, or to introduce some form of periodicity. In theory, this would allow for a high resolution of 3D information. However, because the distance between codewords of nearby pixels is very close to zero, these systems are very sensitive to noise. Furthermore, the observed colors do not only depend on the projected colors, but also on the reflection properties of the measured surface. Because of all these difficulties, these techniques are generally not suitable for dynamic scenes, and usually constrained to pale or neutrally colored objects. In the remainder of this section, we will discuss two possible direct codification strategies: (a) codification based on grey levels; and (b) codification based on a large spectrum of colors.

3.5.1 Grey Levels

Early work in this area was done by Carrihill and Hummel [Carrihill 85], who developed a system which they called an intensity ratio depth sensor. Their approach makes use of two projected patterns: (a) a linear gradient, spread over several vertical columns; and (b) a fully lit pattern, providing a constant illumination. By computing the ratio for each pixel between the registered intensities under both illuminations, it was possible to identify the vertical column illuminating the scene point. Since two patterns must be projected, their system did not consider dynamic scenes. In their seminal work, the authors used a slide projector and an 8-bit monochrome camera, but even after careful color calibration, their method achieved poor accuracy, with a mean error of about 1 cm. This was due to the high sensitivity to noise and non-linearities of the projector device. In 2000, Miyasaka *et al.* [Miyasaka 00] reproduced the intensity ratio depth sensor by using an LCD projector and a 3CCD camera. With this setup, they were able to obtain more accurate results, by taking into account that the reflectance of the surface points is not constant for all the light frequencies.

In 1995, Chazan and Kiryati [G. Chazan 95] extended Carrihill and Hummel’s method by introducing a hierarchical refinement scheme, with the intention of reducing the high noise sensitivity of the original method. Their new method, the *pyramidal intensity-ratio depth sensor*, or *sawtooth sensor*, consisted of consecutively projecting the linear gradient, but gradually increasing its period.
Thus, the first projected pattern consisted of the original pattern projected by Carrihill and Hummel, a simple black-to-white gradient. The second patterns contains two neighboring linear wedges, the third contains four, and so on. The final ($n^{th}$) pattern will contain $2^n$ gradients. Since every period contains a discrete gradient, the number of used gray levels will decrease as the pattern refinement proceeds. As a result, adjacent grey levels will become gradually less similar and more easily distinguishable. The proposed strategy is quite similar to the time-multiplexing techniques we have discussed in section 3.2, except that in this case the exact codewords are not recovered. Furthermore, because they are so similar, they share similar difficulties with scene points illuminted by sharp pattern transitions (recall Figure 3.2). In order to combat these potential errors, every periodic pattern is projected twice, shifting it by half a period. Thus, when reading grey levels close to a sharp transition, the corresponding shifted pattern is used to avoid errors. The sawtooth sensor is more accurate than its predecessor, reducing an error of 1cm on a 80cm distance to the 1mm range, at the cost of increasing the number of patterns to $2^{n+1} + 1$, where $n$ represents the final resolution.

In 1993, Hung proposed the use of a grey level sinusoidal pattern [Hung 93]. Their work is based on the same observation that lies at the basis of this work: the period of the observed pattern increases proportionally to the distance between the projector and the object. Their idea was to estimate the instant frequency in every pixel of the camera image, and then computing the corresponding depth value for every pixel. Preliminary results using synthetic images looked promising, but never got followed up on with the real-world experiments needed for validation. Also, due to their dependence on a spatial neighborhood, they suffer from the typical problems associated with spatial encoding schemes (e.g. their system is restricted to smooth surfaces). In our work, we allow for the projector to change its viewpoint as a means to overcome the need for spatial neighborhoods and the associated problems.
3.5 Direct Encoding

3.5.2 Color Coding

Several authors have extended the ideas from the previous section into the color domain. In 1990, Tajima and Iwakawa [Tajima 90] replaced Carrhill and Hummel’s gradient pattern with a rainbow pattern, in which a large set of vertical narrow slits were encoded with different wavelengths, describing a spectrum from red to blue. They achieved this by diffracting white light through a nematic liquid crystal. The images were grabbed by a monochromatic camera, under two different color filters. Like in the case of grey level encoding, the same approach was repeated after a few years with a new setup. In 1996, Geng [Geng 96] improved the results of this approach by using a CCD camera with a linear variable wavelength filter, reducing the amount of required images to only a single image. A few years later, in 1999, Sato updated the technique once more, proposing a novel variant which only required an LCD projector and a CCD camera [Sato 99]. Moreover, this last incarnation was able to operate largely independent of the spectral reflectance properties of the surface. The technique consisted of projecting a periodic rainbow pattern 3 times, shifting the hue phase 1/3 of its period in every projection. An additional image was synthesized based on a certain linear combination of the three inputs. Sato demonstrated that the hue value of every pixel of this synthesized image was equal to the projected hue value in the first pattern. This served as a basis for the creation of correspondences between synthesized image pixels and the projected rainbow columns.

In 1991, Wust and Capson presented a three-step phase shift method [Wust 91], in which the shifted sinusoidal patterns were distributed over the three different RGB channels, displaced respectively by 0, π/4, and π/2 radians. Using only this single pattern, they recovered the observed phase for each pixel using the following equation:

$$\Phi(x, y) = \arctan\left(\frac{I_r - I_g}{I_g - I_b}\right)$$

(3.2)

where $\Phi(x, y)$ represents the phase in a given pixel $(x, y)$, and $I_r$, $I_g$ and $I_b$ denoted the observed intensities of the different color channels. As this method only requires a single unique pattern, it is able to measure moving surfaces. However, their technique is limited to surfaces that are predominantly color neutral and do not contain large discontinuities. Also, due to the repetitive nature of the pattern, ambiguities might arise.
3.6 Viewpoint Encoding

In 2007, Young et al. [Young 07] were the first to introduce a theoretical framework for replacing temporally encoded structured light patterns with a viewpoint encoding scheme, in the form of additional camera locations. More specifically, by using a high frequency stripe pattern and placing cameras in carefully selected locations, they observed that the epipolar projection in each camera can be made to mimic the binary encoding patterns (see section 3.2.1) normally projected over time.

One important observation made by Young et al. that allows for camera viewpoints to be used as codes is that the epipolar line traditionally used in stereo vision is far too general. The working volume constrains the range of possible depths for the existing epipolar geometry. Thus, the authors focus only on the segment of the epipolar line corresponding to the working volume, which they call the epipolar segment (see Figure 3.14).
3.6 Viewpoint Encoding

Figure 3.15: **Viewpoint code example.** The pixel intensities along the epipolar segment as observed in each camera view are stacked together. Each row can be thought of as a particular code. By carefully choosing the location of the cameras, a unique spatial coding resembling binary temporal schemes is possible. (a) If we place our cameras at equal distances, ambiguities might arise. (b) However, by placing our cameras at well defined locations, we can uniquely identify the intensity correspondence among all cameras.

By restricting the search for stereo correspondences to a single epipolar segment, the matching ambiguity is substantially reduced. By changing the viewpoint of the camera, both the observed part of the code and the projection size of the epipolar segment change. We illustrate this with Figure 3.15, in which we show the pixel intensities along the epipolar segment in each camera view stacked together. Each column of this stacked image corresponds to a single possible surface depth, and each row represents the code provided by a single camera. The codes produced by cameras with smaller epipolar segments (C1,C2) are mostly low-frequency in nature, as the entire depth volume of the segment is occupied by only a small number of stripes. The codes for further cameras (C3-C5) are higher in frequency, as there are many more stripes within the epipolar segment. This is comparable to the different spatial frequencies typically used for temporal binary codes in structured light.

It should be noted that in order to come to a consensus about the pixel’s depth, the camera configuration cannot be chosen at random. E.g. in case of a simple camera configuration arrangement, in which the cameras are located at equal distances, the recovered intensity values do not allow for unambiguous recovery of depth.

Young *et al.* were the first to propose a viewpoint encoded structured light method. However, we shall refer to any method that explicitly exploits its proposed camera configuration as a viewpoint encoded algorithm. One such example in the passive stereo domain is epipolar-plane image analysis, which shows many similarities to the method proposed in this dissertation. We shall elaborate on its relation with our method in chapter 7, after we have introduced our method.
3.7 Focal Encoding

In previous sections we have mentioned several possible encoding mechanisms, exploiting different aspects of the camera/projector setup. Focal encoding, more commonly referred to as depth from defocus, makes use of yet another possible depth cue: projector defocus. Because projector devices were originally designed to illuminate fronto-parallel planar surfaces like a wall or a projection screen, using large apertures in order to produce very bright images, the projected image is only in focus for a very limited depth range. Whereas the occurring projector blur is commonly a source of error for most camera/projector scanning systems, the algorithms in this category exploit it as a depth cue. The main strength of focal encoding techniques is that they are able to produce a depth estimate on a per pixel basis, making them suitable for accurate dense reconstruction even near depth discontinuities. Aside from the method we propose in this dissertation, this is the only\textsuperscript{14} method that allows for such a complete dense reconstruction.

3.7.1 Depth from Camera (De)Focus

There are two types of methods that are referred to as depth from (de)focus algorithms. The oldest methods exploit depth from camera (de)focus [Pentland 87, Ens 93, Nayar 94, Nayar 96, Schechner 00, Favaro 02, Favaro 05]. By taking two or more images of the scene using different camera focal lengths, the blur kernel at each focal length can be estimated and used to determine at which moment the observed pixel was in focus. Such methods have the potential to recover depth at every pixel, regardless of the scene complexity and occlusions.

In order to resolve the focus ambiguity of textureless surfaces, patterns can be projected onto the scene providing additional scene texture [Girod 89]. However, camera defocus kernels depend on local surface geometry. In order to simplify the kernel analysis, most of these works make the so-called equifocal assumption, which states that within a small spatial window surface depth is constant. Unfortunately, this assumption smears shape details and is invalid at depth discontinuities.

To alleviate this problem, Rajagopalan and Chaudhuri [Rajagopalan 97], and Jin et al. [Jin 02] suggested estimating the depths for all pixels simultaneously, based on an energy minimization approach. Unfortunately, these approaches are computationally expensive and like all global minimization strategies prone to getting stuck in local minima.

\textsuperscript{14}To be more specific, this is the only camera/projector shape acquisition system able to produce such a reconstruction. Time-of-flight scanners, such as we have discussed in section 2.3, are also able to build a complete dense reconstruction, but the produced results are rather limited in terms of accuracy.
3.7.2 Depth from Projector Defocus

Most recent methods [Zhang 06a, Gupta 09] make use of projector defocus instead. The key advantage of using projector defocus is that, unlike camera defocus, the kernel is scene independent for most scene surfaces. More specifically, when a 3D scene point is illuminated by the entire projector aperture, its defocus kernel depends only on its distance to the projector lens and not on the neighboring surface geometry. This important difference with camera defocus is due to the fact that projector defocus convolution happens on the projector’s image plane, while camera defocus convolution happens on the scene surface. Recently, several techniques have emerged that exploit this observation.

Zhang et al. [Zhang 06a] projected a shifting illumination pattern with a wide range of frequencies across the scene. For each surface point, the observed radiance over time form the response of its defocus kernel to its excitation by the illuminating pattern. As the pattern is shifted, points at different distances to the projector will exhibit different amounts of blur in their temporal radiance profile. It is this temporal blur profile that is used by the authors as a cue for depth recovery.

Moreno et al. [Moreno-Noguer 07] presented a system for refocusing images and videos of dynamic scenes using a single-view depth estimation method, based on the defocus of a sparse set of removable dots projected onto the scene (see Figure 3.16). The depths corresponding to the projected dots and a color segmentation of the image are used to compute an approximate depth map of the scene while maintaining clean region boundaries. The depth map is used to refocus the acquired image after the dots are removed, simulating realistic depth of field effects.

The previous techniques perform very well on Lambertian and slightly specular materials. However, global illumination effects due to inter-reflections, sub-surface scattering and volumetric scattering introduce strong biases in the recovered scene shape. Recently, Gupta et al. [Gupta 09] have studied the interplay between global illumination and the depth cue of illumination defocus (see Figure 3.17). By following the spatial domain approach of Nayar et al. [Nayar 06], they derived an approximate defocus-invariant measure of global illumination, expressing both effects as low pass filters and separating them without explicitly modeling the light transport. This enables them to perform depth recovery in the presence of global illumination, and factoring out the effects of defocus for direct/global separation in large depth scenes. They provide two algorithms: (a) the first algorithm requires a sweep of the projector focal plane across the scene and is essentially dual to shape-from-camera-focus techniques; (b) the second algorithm requires only two focal plane settings and is similar to shape-from-camera-defocus methods. Finally, a comparison between single and multi-focal methods is given in Figure 3.18.
Figure 3.16: Depth from (de)focus. Moreno et al. [Moreno-Noguer 07] presented a system for refocusing images and videos of dynamic scenes. (a) The acquired image, with a sparse set of dots projected onto the scene. (b) By computing the defocus of these dots, an approximate depth map can be computed. (c-d) Using the approximate depths, the image can be refocused on either the foreground or the background subject.

Figure 3.17: Depth from (de)focus and global light transport. Scene recovery results using the technique of Gupta et al. [Gupta 09]. This scene consists of organic materials: a few peppers, two pumpkins, the green plant, some marbles and eggs and a flower-pot, in this depth order.

Figure 3.18: Comparison between single/multi-focal depth from projector defocus. (left) The scene to be reconstructed. (center) The recovered depth map using a single projector focal plane [Zhang 06a]. (right) The recovered depth map using multiple projector focal planes [Gupta 09].
Chapter 4

Related Work - Signal Processing

Contents

4.1 Single Tone Estimation ........................................ 48
  4.1.1 Problem Statement ......................................... 48
  4.1.2 Macleod’s Method ........................................ 49

4.2 Sub-Nyquist Sampling of Periodic Signals .............. 51
  4.2.1 Problem Statement ......................................... 51
  4.2.2 Compressed Sensing ...................................... 51
  4.2.3 Coded Strobing Photography ............................. 51
In the two previous chapters, we have provided an overview of the different types of shape acquisition methods that are in existence today, with a more narrowed focus on camera/projector based 3D scanning systems. Unlike any of these approaches, the method presented in this dissertation casts depth estimation as a signal processing problem. Because of this, we will also cover the relevant signal processing literature in this chapter. However, as signal processing itself is not the focus of this dissertation, we will not be as exhaustive in our discussion as in the previous chapters. Two topics will be covered in this chapter: (a) the signal processing problem that is relevant to our method: single tone estimation; (b) the sub-Nyquist sampling of periodic signals.

4.1 Single Tone Estimation

4.1.1 Problem Statement

The single tone estimation problem consists of determining the frequency - and sometimes also the phase - of a single pure tone (blue signal) from a discrete signal corrupted by noise (red samples).

![Figure 4.1: The single tone estimation problem consists of determining the frequency - and sometimes also the phase - of a single pure tone (blue signal) from a discrete signal corrupted by noise (red samples).](image)

In the two previous chapters, we have provided an overview of the different types of shape acquisition methods that are in existence today, with a more narrowed focus on camera/projector based 3D scanning systems. Unlike any of these approaches, the method presented in this dissertation casts depth estimation as a signal processing problem. Because of this, we will also cover the relevant signal processing literature in this chapter. However, as signal processing itself is not the focus of this dissertation, we will not be as exhaustive in our discussion as in the previous chapters. Two topics will be covered in this chapter: (a) the signal processing problem that is relevant to our method: single tone estimation; (b) the sub-Nyquist sampling of periodic signals.

4.1 Single Tone Estimation

4.1.1 Problem Statement

The single tone estimation problem consists of determining the frequency - and sometimes also the phase - of a single pure tone from a discrete signal corrupted by noise. This is essentially a parameter estimation problem, where we want to fit a sinusoidal curve $\hat{I}(t)$ to a discrete set of data points $f(t), t \in [0, T - 1]$:

$$f(t) = \mu + A \sin(\omega t + \phi) + \epsilon_t$$  \hspace{1cm} (4.1)

with additive noise $\epsilon_t$, sampled at $T$ points in time. In general, the noise is assumed to be Gaussian white noise with $E[\epsilon_t] = 0$ and $\text{Var}[\epsilon_t] = \sigma^2$, although some works generalize upon this. Even though it is possible to apply a simple least-squares approach to estimating the sinusoidal curve $f$, more accurate methods have been developed within the last decades [Quinn 94, Quinn 97, Macleod 98].
4.1 Single Tone Estimation

Figure 4.2: Single tone estimation by inspecting the Fourier power spectrum. In this power spectrum (of the signal in Figure 4.1), we see a clear peak at frequency component $k = 4$.

As it is not the purpose of this work to provide a complete survey of the work done on this problem, we restrict ourselves to a single well-performing technique by Macleod [Macleod 98], based on the interpolation of Fourier coefficients. This technique approaches the Cramér-Rao bound, a theoretical limit and benchmark for the performance of algorithms of this type, and has proven to be well-suited for our purposes.

4.1.2 Macleod’s Method

The single tone estimation technique of Macleod [Macleod 98] is based on the interpolation of Fourier coefficients. In this section, we will first introduce a naive Fourier based frequency estimation approach, followed by the proposed interpolation algorithm.

Naive Fourier Method

Fourier analysis deals with the decomposition (and synthesis) of a complex signal into a series of much simpler trigonometric functions. In case the complex signal is a single continuous sinusoidal function, finding its frequency is equivalent to determining the corresponding component in the decomposed signal. Unfortunately, in practice we are never dealing with a continuous signal, but with discrete measurements of the continuous sinusoidal signal.

$$f(t) = \mu + A\sin(\omega t + \phi) + \epsilon_t, \quad t \in [0, T - 1]$$

with additive noise $\epsilon_t$, sampled at $T$ different points in time
Like its continuous counterpart, discrete complex signals $f(t)$ can also be decomposed into a set of discrete trigonometric functions, by converting the signal into an equivalent Fourier representation $F(k)$, by applying the Discrete Fourier Transform (DFT):

$$F(k) = \sum_{t=0}^{T-1} f(t) e^{-2\pi i kt} \quad k = 0, \ldots, T - 1 \quad (4.3)$$

This Fourier representation can be further decomposed into two spectra: (a) the phase spectrum, which holds all information regarding the phase shifts of the individual trigonometric components, and (b) the power spectrum, which represents the relative amplitudes of all frequency components. As our goal is frequency estimation, we are mainly interested in the power spectrum $P(k)$:

$$P(k) = F(k) F^*(k) \quad (4.4)$$

where $F^*(k)$ is the complex conjugate of $F(k)$. After this conversion, we locate the global maximum $k_{max}$ in the power spectrum. This maximum should be an acceptable approximation of the frequency of the original signal. This is illustrated in Figure 4.2, where the global maximum is located at $k_{max} = 4$.

**Fourier Interpolation**

If we take a look at the power spectrum in Figure 4.2, we observe a nice clean peak at coefficient $k = 4$. However, in practice this will rarely be the case due to aliasing effects. The 'true' continuous peak could be located at $\hat{k}_{max} = 4.2$, and we would be unable to detect it due to the limited resolution of the observed signal. Thus, if we want to achieve sub-coefficient precision, some form of interpolation will have to be applied.

$$\hat{k}_{max} = k_{max} + \delta \quad \delta \in \left[-\frac{1}{2}, +\frac{1}{2}\right] \quad (4.5)$$

In his work, Macleod proposed the following algorithm for determining $\delta$. Assuming we have already determined $k_{max}$ as described above, the first step consists of correcting the phase components of frequency components $k_{max} - 1, k_{max},$ and $k_{max} + 1$ in order assert that the corrected values $G(k) \in \mathbb{R}$:

$$G(k_{max} - 1) = R(k_{max}) R(k_{max} - 1) + I(k_{max}) I(k_{max} - 1) \quad (4.6)$$

$$G(k_{max}) = R(k_{max}) R(k_{max}) + I(k_{max}) I(k_{max}) \quad (4.7)$$

$$G(k_{max} + 1) = R(k_{max}) R(k_{max} + 1) + I(k_{max}) I(k_{max} + 1) \quad (4.8)$$

where $R(k)$ and $I(k)$ are respectively the real and imaginary components of the complex number $F(k) \in \mathbb{C}$. 
4.2 Sub-Nyquist Sampling of Periodic Signals

Once the assertion is made that we are dealing with real numbers, we are ready to calculate the offset $\delta$:

$$\delta = \frac{\sqrt{1 + 8\gamma^2} - 1}{4\gamma}$$

$$\gamma = \frac{G(k_{\text{max}} - 1) - G(k_{\text{max}} + 1)}{2G(k_{\text{max}}) + G(k_{\text{max}} - 1) + G(k_{\text{max}} + 1)}$$ (4.9)

For more details on the exact derivation of these formulas, we refer to the paper [Macleod 98].

4.2 Sub-Nyquist Sampling of Periodic Signals

The second topic of this chapter concerns the sub-Nyquist sampling of periodic (video) signals, by applying recent advances in the domains of compressed sensing and computational photography.

4.2.1 Problem Statement

The Nyquist sampling theorem states that, if a continuous signal $f(t)$ contains no frequencies higher than $N$ hertz, it is completely determined by its values at discrete set of points spaced $\frac{1}{2N}$ seconds apart. In this section, we will discuss the reconstruction of continuous signals with maximum frequencies of $N$ hertz, from discrete sampling rates lower than the corresponding Nyquist rate $2N$.

4.2.2 Compressed Sensing

The sampling theorem provides a sufficient condition, but not a necessary one, for perfect reconstruction. Recently, the field of compressed sensing [Donoho 06, Candès 06] has provided stricter sampling conditions in cases where the underlying signal is known to be sparse, allowing for sub-Nyquist sampling of periodic signals.

4.2.3 Coded Strobing Photography

This new approach by Veeraraghavan et al. [Veeraraghavan 10] shows that, by applying temporal modulation, one can observe a high-speed periodic event well beyond the abilities of a low-frame rate camera. Their key result is that for such signals the Nyquist rate constraint can be imposed on the strobe-rate rather than the sensor-rate. The technique is based on intentional aliasing of the frequency components of the periodic signal, while the reconstruction algorithm exploits recent advances in sparse representations and compressed sensing. For a visual representation of their approach, we refer to Figure 4.3.
Periodic phenomenon with unknown period $P$ (say 16 ms)

Video camera with frame rate $f_s = 25$fps

Every frame is modulated $U = 80$ times with a unique binary code by opening & closing the shutter

Capture $M = 125$ frames in 5s

Structured sparse recovery

Computationally reconstruct $N = 10000$ frames

Figure 4.3: **Coded strobing.** (a) Coded strobing schematic: Capture periodic phenomena by coding the shutter of a low frame-rate camera during every frame and reconstruct the phenomena at a higher rate by exploiting sparsity of periodic signals. (b) Observation and signal model: At every pixel, the signal is periodic in time and has sparse Fourier coefficients. Observed values are a linear combination of the Fourier coefficients. The resulting under-determined system of equations are solved by enforcing sparsity of the coefficients [Veeraraghavan 10].
4.2 Sub-Nyquist Sampling of Periodic Signals

Camera Observation Model

Consider a luminance signal \( x(t) \), band-limited to the range \([ -f_{\text{Max}}, f_{\text{Max}} ]\). In order to accurately recover the continuous signal, we only need to sample the signal at a temporal resolution of \( \delta t = \frac{2}{2f_{\text{Max}}} \). If the total time of observing the signal is \( N \cdot \delta t \), then the \( N \) samples can be represented in an \( N \)-dimensional vector \( x \).

In a normal camera, the radiance collected at a single pixel is integrated over the exposure time, and the sum is recorded as the observed intensity at that pixel. Instead of integrating during the entire frame duration, amplitude modulation of the incoming radiance values is performed, strobing the incoming light, before integration. Thus, the observed intensity values at a given pixel can be represented by a \( M \times 1 \) vector \( y \):

\[
y = Cx + \eta \quad (4.10)
\]

where \( C \) is an \( M \times N \) matrix, which performs both the modulation and integration for frame duration, and \( \eta \) represents the signal noise. If the camera observes a frame every \( T_s \) seconds, the total number of frames/observations would be \( M = \frac{N \delta t}{T_s} \). The camera sampling time \( T_s \) is far larger than the time resolution we would like to achieve (\( \delta t \)), which means \( M \ll N \). The resulting upsampling factor \( U \) (or decimation ratio) can be defined as

\[
U = \frac{N}{M} = \frac{2f_{\text{Max}}}{f_s} \quad (4.11)
\]

Signal Model

If \( x \), the incoming luminance at a pixel is band limited, it can be represented by a linear system

\[
x = Bs \quad (4.12)
\]

where the columns of \( B \) contain Fourier basis elements. Furthermore, as we assume that the signal \( x(t) \) is periodic, the basis coefficient vector \( s \) has to be sparse.

Mixing Model

Putting together the signal and observation model, the intensities in the observed frames are related to the basis coefficients by

\[
y = Cx + \eta = CBs + \eta = As + \eta \quad (4.13)
\]

where \( A \) becomes the effective mixing matrix of the forward process. Recovery of the periodic signal \( x \) thus comes down to solving this linear system of equations, assuming \( A \) has the restricted isometry property\(^1\) due to a well-chosen code matrix \( C \).

\(^1\)In linear algebra, the restricted isometry property characterizes matrices which are nearly orthonormal, at least when operating on sparse vectors [Candés 05].
Reconstruction

In eq.(4.13), the number of unknowns exceeds the number of known variables by a factor $U$ and as a result the system of equations (4.13) is severely underdetermined ($M \ll N$). This knowledge by itself however is insufficient. In order to obtain a robust solution, further knowledge about the signal must be used. Since the Fourier coefficients $s$ of a periodic signal $x$ are sparse, a reconstruction technique enforcing sparsity of the vector $s$ could still hope to recover the periodic signal $x$.

Veeraraghavan et al. [Veeraraghavan 10] have presented two reconstruction algorithms. Their first method enforces the sparsity of the Fourier coefficients, by applying a minimization technique similar to those used in compressed sensing: estimating a sparse vector $s$ with $K$ non-zero entries, that satisfies $y = As + \eta$, can be formulated as an $\ell_0$ optimization problem:

$$
\min ||s||_0 \quad s.t. \quad ||y - As||_2 \leq \varepsilon \quad (4.14)
$$

which can be solved using a greedy algorithm (CoSaMP [Needell 08]), or by solving the equivalent (for a sufficiently small $K$) $\ell_1$-norm minimization

$$
\min ||s||_1 \quad s.t. \quad ||y - As||_2 \leq \varepsilon \quad (4.15)
$$

Unfortunately, both minimizations mentioned above do not take into account the specific structure of the sparse coefficients $s$, and as such the authors developed another technique that increased the robustness of the recovered signal.

Their second algorithm exploits the specific structure of a period signal with a period $P = \frac{1}{f_p}$, as the only frequency components such a signal has should be located in the the small bands at the harmonics $j \cdot f_p$, where $j$ is an integer value. After estimating the fundamental frequency $f_p$, a set $S_{f_p}$ with basis elements $[j \cdot f_p - \Delta f_p, j \cdot f_p + \Delta f_p]$ for $j \in \{Q, \ldots, 0, 1, \ldots, Q\}$ is established such that all sparse Fourier coefficients will lie in this smaller set. This allows for a new, more narrowly defined minimization problem:

$$
\min ||s||_0 \quad s.t. \quad ||y - As||_2 \leq \varepsilon \land nonZero(s) \in S_{f_p} \quad (4.16)
$$

where $nonZero(s)$ is a set containing all the non-zero elements in the reconstructed vector $s$. For more algorithms which exploit further statistical structure in support of the sparse coefficients, we refer to the work of Baraniuk et al. [Baraniuk 10] on model-based compressed sensing.
Chapter 5

Depth from Frequency Analysis

Contents

5.1 Depth Estimation as a Signal Processing Problem . . . . 56
  5.1.1 Composition of the Observed Temporal Intensities . . . . 59
5.2 The Reflectance Function . . . . . . . . . . . . . . . . . . 59
  5.2.1 Recovering the Reflectance Function . . . . . . . . . . 61
5.3 The Visibility Function . . . . . . . . . . . . . . . . . . . 62
  5.3.1 Recovering the Visibility Function . . . . . . . . . . . 62
5.4 The Dominant Frequency . . . . . . . . . . . . . . . . . . . 64
  5.4.1 Recovering the Dominant Frequency . . . . . . . . . . 64
5.5 Our Setup Prototype . . . . . . . . . . . . . . . . . . . . . 64
5.6 Influence of Scene Factors . . . . . . . . . . . . . . . . . 65
  5.6.1 Influence of the Reflectance & Visibility Function . . . . 65
  5.6.2 Influence of Interreflections and Stray Light? . . . . . 67
5.7 Practical Considerations . . . . . . . . . . . . . . . . . . . 70
5.8 Projector Depth vs. Camera Depth . . . . . . . . . . . . . 71
5.9 Discussion . . . . . . . . . . . . . . . . . . . . . . . . . . . 72
  5.9.1 Results . . . . . . . . . . . . . . . . . . . . . . . . . . 72
  5.9.2 Comparison to the State of the Art . . . . . . . . . . . 78
  5.9.3 Limitations and Future Work . . . . . . . . . . . . . . 81
5.10 Conclusions . . . . . . . . . . . . . . . . . . . . . . . . . . 82
In the previous chapters, we have provided an overview of the broad spectrum of shape acquisition methods that are in use today, followed by a brief introduction to signal processing problems that are relevant to our work. Like the approaches presented in chapter 3, the method presented in this dissertation makes use of active illumination provided by a projector. However, unlike any of the methods discussed, our method makes use of a mobile projector, i.e., the viewpoint of the projector changes over time, emitting a static illumination pattern. As such, it can be classified as a new type of viewpoint encoding structured light\(^1\). In addition, our approach produces a depth estimate for each\(^2\) individual camera pixel, based solely on the intensities observed for that particular pixel. As no additional spatial information is used, our algorithm lends itself well to parallelization.

The remainder of this chapter is structured as follows: (a) an introduction to the actual method, illustrating how our approach casts depth estimation as a signal processing problem; (b) an overview of the many different factors that have an influence on the observed signal; (c) a discussion of the proposed method, concerning its place within the taxonomy presented in chapter 3.

### 5.1 Depth Estimation as a Signal Processing Problem

An overview of our approach is shown in Figure 5.1. In the left image we see a projector translating at a constant velocity, in the direction of the projector’s horizontal axis. If we project a periodic\(^3\) pattern onto the scene, the illumination arriving at a random scene point will be dependent on the distance between the point and the principal plane of the projector. During the remainder of this chapter, when we refer to the depth of an observed scene point, we are talking about this distance. It is important to note that at this stage, the depth of a scene point is defined in terms of the projector, and not in terms of the camera\(^4\).

---

1. Unlike the previous viewpoint encoding methods however, in our case it is the projector that illuminates the scene from multiple viewpoints, instead of static illumination observed from multiple predefined viewpoints.
2. The proposed method is a dense depth estimation method, producing a depth estimate for each camera pixel. This is an improvement over the classical temporal structured light methods, which also provide a dense estimation, but only for each projector pixel.
3. At this stage, the choice between stripe or wave patterns is purely an aesthetic one. Only at the frequency refinement stage will we observe a difference between the use of stripe or wave patterns.
4. Recovering the depth of the scene point in terms of the camera is possible under special camera/projector configurations, or in general configurations using an additional pattern. How to solve the general camera-projector depth conversion will be the topic of chapter 6.
5.1 Depth Estimation as a Signal Processing Problem

Figure 5.1: Method overview. (left) A projector emitting a periodic pattern is translated at a constant speed, according to its horizontal axis. (center) As a result, observed scene points (indicated by red circles) at different depths \( \{\mu, 2\mu, 3\mu\} \) will be illuminated by the pattern at different speeds. If we take a closer look at the recorded intensities in time, the signals clearly have different periods \( \{p, 2p, 3p\} \), linearly related to the depth values. (right) Converting the observed changes in intensity over time into an equivalent frequency domain representation allows us to find a dominant frequency, which can be directly converted into a corresponding wave period and depth value.

If we take a closer look at the illumination arriving at the different scene points in Figure 5.1, it becomes clear that there exists a linear relationship between the depth of the points and the period of their observed patterns. Converting the observed changes in intensity over time into an equivalent frequency domain representation will allow us to find the dominant frequency \( f \), which can be directly converted into a corresponding depth value \( \mu \):

\[
\mu = \frac{s}{f}. \tag{5.1}
\]

The scaling factor \( s \) is uniquely defined by the parameters of the projector (translating at \( v \) meters per second, emitting a pattern of \( N \) stripes, with a field-of-view of \( \alpha \) radians), and the camera (recording at \( r \) frames per second). The distance traveled (in meters) by a wave in a single period \( p = 1/f \) at depth \( \mu \) is equal to \( (v \cdot p)/r \). Using basic trigonometry, we can show this to be equal to \( (2\mu \cdot \tan(\alpha/2))/N \), which leads to a uniquely defined linear relationship between depth and observed frequency:

\[
s = \frac{N \cdot v}{2 \tan(\alpha/2) \cdot r}. \tag{5.2}
\]

Equations (5.1) and (5.2) are illustrated in Figure 5.2.
A linear relationship exists between the depth $\mu$ (in meters) of the scene points and the period $p$ (in frames, $p = 1/f$) of their observed periodic patterns. If a projector with a field of view of $\alpha$ radians, translating at $v$ meters per second, emits $N$ wave periods, and the camera captures these at a rate of $r$ frames per second, then $\mu = \left( \frac{N \cdot v}{\alpha \cdot \tan(\alpha/2) \cdot r} \right) \cdot p$.

The composition of the modulated intensity signal. The observed intensities $I(t)$ at an unoccluded scene point are the product of the reflected wave pattern emitted by the projector, $P(t)$, and the reflectance function $R(t)$.
5.2 The Reflectance Function

5.1.1 Composition of the Observed Temporal Intensities

In this section we will describe the influence of the different scene properties on the wave pattern of the discrete temporal intensity function $I(t)$, where $t \in [0, T-1]$ represents the sample time. We will discuss the influence of material properties of the scanned objects, as well as the influence of scene geometry.

These observed intensities $I(t)$ of a fixed object surface point are the result of a moving stripe pattern behind a moving point light source (the intensities $P(t)$ emitted by the projector, reflected by the scene point), multiplied with the reflectance function $R(t)$ representing the material properties of the observed scene point at the sampled angles, and a visibility function $V(t)$ representing the influence of the scene geometry:

$$I(t) = P(t) \cdot R(t) \cdot V(t)$$

(5.3)

If we manage to estimate $\tilde{R}(t) = R(t) \cdot V(t)$, the reflectance function of the unoccluded time steps $t \in [0, t_{\text{max}}]$, multiplying the observed intensities $I(t)$ with the function $\frac{1}{\tilde{R}(t)}$ removes its influence on the signal. What remains is an approximately sinusoidal pattern $\tilde{P}(t) \cdot V(t)$ from which we can accurately determine the depth.

In the following sections, we will take a closer look at how we can compute both an estimate $\tilde{R}(t)$ for the reflectance function, and an estimate $\tilde{V}(t)$ for the visibility function, in addition to what these functions physically represent, and what their influence is on the retrieval of the dominant frequency $f$ from incoming signal.

5.2 The Reflectance Function

In this section, we will first relate the classical bidirectional reflectance distribution function to the reflectance function as we have defined it in the previous section.

The bidirectional reflectance distribution function (BRDF) is a four-dimensional function $f_r(\omega_i \rightarrow \omega_o)$ that defines how light is reflected at an opaque surface. The function takes an incoming light direction $\omega_i$, and an outgoing direction $\omega_o$, both defined with respect to the surface normal $\eta$, and returns the ratio of reflected radiance exiting along $\omega_o$ to the irradiance incident on the surface from direction $\omega_i$.

The reflectance function $R(t)$, as defined in the previous section, is a one-dimensional function that defines how much light is reflected by a fixed object surface point $x$, observed from a fixed angle. The function takes a time step $t$, defining the position $p_i(t)$ of the illumination source, and returns the ratio of reflected radiance exiting along the fixed angle of observation to the irradiance incident on the surface from the direction $p_i(t) - x$, multiplied by a distance attenuation factor $d(p_i(t), x)$. 
Comparing $f_r(\omega_i \rightarrow \omega_o)$ to $R(t)$ provides us with the following observations:

- The BRDF $f_r(\omega_i \rightarrow \omega_o)$ and our reflectance function $R(t)$ can only be related if we are able to establish each position $p_i(t)$ of the illumination source at each time step $t$, with respect to the point cloud $\{x_j\}$. This will be covered later, in section 6.2.

- After removing the influence of the distance attenuation factor $d_i(p_i(t), x)$, the remainder of $R(t)$ constitutes a discrete sampling of a one-dimensional (horizontal) slice of the four-dimensional BRDF. If our projector is translated by a robotic arm, we have the option of repeating the process for multiple vertical ordinates, adding an additional dimension to the BRDF sampling space.

**Approximation of BRDF by Phong Reflection Model**

Even though BRDF acquisition is not the focus of this work, we might still be able to estimate the reflectance properties of materials that can be approximated by simplified reflection models, such as the Phong reflectance model:

$$f_r(x, \omega_i \leftrightarrow \omega_o) = k_a i_a + \sum_{m \in \text{lights}} \left( k_d (\omega_{im} \cdot \eta) i_d + k_s (\omega_{rm} \cdot \omega_o)^n i_s \right)$$  \hspace{1cm} (5.4)

where $i_s$ and $i_d$ represent the intensities of the specular and diffuse components of the light sources, $i_a$ controls the ambient lighting, $k_s$ is the specular reflection constant$^5$, $k_d$ the diffuse reflection constant$^6$, $k_a$ the ambient reflection constant$^7$, $n$ is the shininess fact, $\eta$ is the surface normal, and $\omega_{rm}$ is the direction of perfect reflection$^8$. All vectors are unit vectors. For our controlled setup, in which no ambient light is present, and only a single moving white light source is used, equation (5.4) further simplifies to:

$$f_r(x, \omega_i \leftrightarrow \omega_o) = k_d (\omega_i \cdot \eta) + k_s (\omega_r \cdot \omega_o)^n$$  \hspace{1cm} (5.5)

$$= k_d \cos(\hat{\omega}_i \eta) + k_s (\cos(\hat{\omega}_r \omega_o))^n$$  \hspace{1cm} (5.6)

Assuming we have already performed the shape acquisition steps described in this work, we should have both the scene structure and camera motion at each time step $t$, providing us with the values of $\omega_i, \omega_r$ and $\omega_o$. Additionally, we know that the influence of the diffuse component spans the entire sampling space, while the specular component will be centered around a smaller area, if present at all.

$^5$The ratio of reflection of the specular term of incoming light
$^6$The ratio of reflection of the diffuse term of incoming light.
$^7$The ratio of reflection of the ambient term present in all points in the scene rendered.
$^8$The law of perfect reflection: $\omega_r = \frac{2(\eta \cdot \omega_o)\eta - \omega_o}{\| \omega_o \|^2}$
5.2 The Reflectance Function

5.2.1 Recovering the Reflectance Function

In this section we will describe the process of recovering the reflectance function $R(t)$ for the visible part of the reflected wave pattern. If the fundamental frequency of the signal was known, we could apply standard amplitude demodulation techniques in order to recover $R(t)$. Unfortunately, we are recovering the reflectance function as a means to finding this frequency. Thus, we propose the following two-step process: (a) accurately recovering the local extrema; and (b) interpolating values inbetween subsequent local maxima, producing an approximation of the reflection function.
Depth from Frequency Analysis

Estimating Local Extrema
Locating the local extrema is a matter of scale, which we define in terms of minimal required contrast value $C$. If we want to verify if a data point at time step $t$ represents a local extremum, we first compute the minimal interval $[t_{\text{begin}}, t_{\text{end}}]$ around $t$, for which $\|t - t_{\text{begin}}\| \geq C$ and $\|t - t_{\text{end}}\| \geq C$. Once this interval is known, we can verify if $I(t)$ is maximal/minimal for all values $I(s), s \in [t_{\text{begin}}, t_{\text{end}}]$.

Estimating the Reflectance Function
Based on the local maxima estimated in the previous step, we are able to create a piecewise linear function $\bar{R}(t)$ which approximates the true reflectance function $R(t)$ by connecting and interpolating between each pair of subsequent maxima.

5.3 The Visibility Function
The visibility function is a binary function $\bar{V}(t) : [0, t_{\text{max}}] \mapsto \{0, 1\}$ which indicates if the observed scene point was illuminated by the projector at time $t$. Thus, it encodes scene structure and the associated occlusions.

5.3.1 Recovering the Visibility Function
In this section we will describe the process of recovering the visibility function $V(t)$ from the reflected wave pattern $I(t)$. This is a four-step process: (a) accurately recovering the local extrema; (b) computing a rough estimate of the wave period; (c) removing false local extrema from the initial step; and (d) based on the remaining local extrema, computing the regions for which we have confirmed pattern visibility, thus providing us with an estimate $\bar{V}(t)$ of the visibility function.

Estimating the Period
Given a set of local maxima, we can now compute an initial estimate of the wave period $p$. Theoretically, for each local maximum $I(t_i)$, the distance $d_i = \|t_i - t_{i-1}\|$ should approximately equal to the wave period $p$. Based on this notion, we define our period estimate $\tilde{p}$ as the median value of the set of computed $\{d_i\}, \forall i \in [1, N - 1]$, where $N$ is the number of local maxima. However, in practice it occurs that local maxima are not uniquely defined, but are located over two - or in some cases more - time steps. In order to avoid outliers due to this scenario, we remove all $d_i < 3$ from the set $\{d_i\}$, prior to median estimation. Also, as the distance between subsequent local minima is similar to those of the local maxima, we are able to extend the set $\{d_i\}$ with additional period samples. Finally, it is important to note that this estimate of
5.3 The Visibility Function

Estimating the Visibility Function

For a given set of observed temporal intensities $I(t)$, we compute an estimate of the underlying visibility function $V(t)$ by estimating the local extrema of $I(t)$, computing an estimate $\hat{p}$ of the wave period, and locating the intervals for which we have missing local extrema.

the wave period is restricted to integer values, making it virtually useless as a direct depth measure because of the limited amount of potential depth planes.

Removing Outliers from Set of Local Extrema

As mentioned in the previous paragraph, it occurs in practice that local maxima are not uniquely defined, but rather spread out over two or more time steps due to a combination of a very high sampling frequency and noise. In order to combat this effect, we need to group these time steps together, allocating a mean time step $\bar{t}$ for a set of subsequent $\{t_{i-1}, t_i\}$’s for which $d_i < \frac{3}{2} p$. For future computations, we will use this time step instead of the timesteps it represents.

Estimating the Visibility Function

Once we have removed the outliers from the ordered set of local maxima, we are ready to identify all visible segments $[t_{\text{begin}}, t_{\text{end}}]$, with $t_{\text{begin}}, t_{\text{end}} \in [0, t_{\text{max}}]$. We define a visible segment $[t_{\text{begin}}, t_{\text{end}}]$ in function of these local maxima:

$$[t_{\text{begin}}, t_{\text{end}}] = \left[ t_{\text{first}} - \frac{p}{2}, t_{\text{first}} \right] \cup \left( \bigcup [t_{i-1}, t_i] \right) \cup \left[ t_{\text{last}}, t_{\text{last}} + \frac{p}{2} \right]$$ (5.7)

where all $I(t_i)$ are local maxima for which $d_i < \frac{3}{2} p$ and $i \in [\text{first} + 1, \text{last}]$. Combining all detected visible segments provides us with a binary function $V(t) : [0, t_{\text{max}}] \mapsto \{0, 1\}$, as is illustrated in Figure 5.5.
5.4 The Dominant Frequency

The dominant frequency $f$ of the discrete temporal intensity function $I(t)$, where $t \in [0, T - 1]$ represents the sample time, is linearly related to the depth of the fixed observed scene point. In the previous sections, we have discussed how to estimate the influence of material properties and scene geometry, by their respective functions $R(t)$ and $V(t)$. If we undo their influence on the incoming signal, we retain an approximately sinusoidal reflected pattern $\bar{P}(t)$:

$$\bar{P}(t) = I(t) \cdot \left( R(t) \cdot V(t) \right)^{-1}$$  \hspace{1cm} (5.8)

$$\approx \mu + A \sin(\omega t + \phi)$$  \hspace{1cm} (5.9)

5.4.1 Recovering the Dominant Frequency

Once we recover an approximated sine wave, it becomes possible to apply known signal processing techniques in order to recover the dominant frequency. For more information on how to recover this frequency from the signal, we refer to our discussion of the single tone estimation problem in section 4.1. We will illustrate the difference between the use of both methods in our discussion section.

5.5 Our Setup Prototype

An image of our setup can be found in Figure 5.6. The bulk of our setup is made using off-the-shelf Lego Technic [Leg]. A light-weight pocket projector (Samsung Pocket Imager SP-P310ME Projector) is placed on a platform. This platform is translated in the direction of the projector’s horizontal axis, by applying rotational movement on the rail attached to the platform. Activating the motor on the right results in a slow translational movement in the direction of choice. A single camera (Point Grey Research Grasshopper [PGR]) is encapsulated into our Lego [Leg] framework, observing the scene. Our results were generated with variable camera resolutions at 60 frames per second. We used stripe patterns with variable stripe sizes, varying from 2 to 8 pixels, depending on the required depth range. The native resolution of the used projector was $800 \times 600$.

\footnote{While it is technically impossible to undo the influence of the visibility function $V(t)$, we will argue in section 5.6.1 that its influence on the recovery of $f$ from the signal is marginal.}
5.6 Influence of Scene Factors

In this section, we will discuss the influence of the many different scene factors on our observed illumination pattern, such as the material properties (represented by the reflectance function), scene geometry (occlusions represented by the visibility functions), interreflections, and subsurface scattering.

5.6.1 Influence of the Reflectance & Visibility Function

Due to the convolution theorem, which states that a multiplication (∙) in the time domain is equal to a convolution (⊗) in the frequency domain, eq. (5.3) is equal to:

\[
\mathcal{F} \{ I(t) \} = \mathcal{F} \{ P(t) \cdot V(t) \cdot R(t) \} = \mathcal{F} \{ P(t) \cdot V(t) \} \otimes \mathcal{F} \{ R(t) \} \]

(5.10)

where \( \mathcal{F} \) indicates the Fourier transformed of the original function. In this form, assuming \( P(t) \) is a perfect sinusoidal wave pattern, we can see that \( \mathcal{F} \{ P(t) \} \) is the sum of an impulse function with \( \delta \) equal to zero, and an impulse function with \( \delta \) equal to the dominant frequency \( f \) of the stripe pattern. As it is this \( f \) we are interested in, we neglect the lower end of the frequency spectrum in order to avoid false positives. If we have a prior estimate the wave period \( \bar{p} \), such as from the computation of the visibility function in section 5.3.1, we can even further restrict our search of \( f \) to the interval \( \left[ \frac{1}{\bar{p}^2}, \frac{1}{\bar{p}^2} \right] \).

Figure 5.6: Our shape acquisition setup. A projector emitting a static illumination pattern is translated in the direction of its horizontal axis, while a static camera observes the scene.
Figure 5.7: Influence of the visibility function on the signal power spectrum. The observed intensities $I(t)$ at an occluded scene point are the product of the reflected wave pattern emitted by the projector, $P(t)$, a binary visibility function $V(t)$ which dictates when the projector is able to illuminate the scene point, and the reflectance function, $R(t)$. (left) The temporal functions $I(t)$, $R(t)$, $P(t) \cdot V(t)$ and $V(t)$. (right) The corresponding Fourier power spectrum.

The Form of the Transformed Reflectance Function $\hat{R}(t)$

In many practical cases, it is not necessary to perform a prior estimate of the reflectance function $R(t)$ in order to recover a good depth estimate. E.g. this is commonly the case for objects made of primarily diffuse materials. For such materials, the transformed reflectance function $\hat{R}(t)$ will consist mainly of a few low frequency components. Because of this, if $f$ was the dominant frequency in $\hat{R}(t)$, eq.(5.11) dictates it should remain the dominant frequency in $\hat{R}(t) \cdot V(t)$. Only when $\hat{R}(t)$ takes on more exotic shapes, e.g. the transformed reflectance function of translucent materials, $\hat{R}(t)$ might amplify low-amplitude noise to a sufficient level for $f$ to no longer be the dominant frequency. Also, in case the choice of $f$ is ambiguous due to frequency sampling artifacts, the reflectance function might affect the choice of the dominant frequency between two neighboring frequencies.

Influence of the Scene Geometry

Because of occlusions by other scene points, it is plausible that not every point in the scene is illuminated by the projector with the wave pattern for the entire duration of the scene capture. However, in most cases this incomplete wave pattern $P(t) \cdot V(t)$ will still produce a familiar $\hat{R}(P(t)) = \hat{R}(P(t)) \odot \hat{V}(t)$, from which we can robustly retrieve the dominant frequency. With the addition of a few minor low frequency components due to the missing stripes, the Fourier power spectrum largely remains the same. This is illustrated in Figure 5.7.
5.6 Influence of Scene Factors

Observed Intensities $I(t)$, Reflectance Function $R(t)$, Occluded Stripe Pattern $P(t)$, Visibility Function $V(t)$

Figure 5.8: *Influence of a specular highlight on the signal power spectrum.* The occurrence of a specular highlight is treated similar to an occlusion. As a result of the lack of uniquely defined local maxima in the highlight region, our visibility function $V(t)$ will label this region as ‘occluded’, $V(t) = 0$.

**Specular Highlights**

For many structured light methods, specular highlights are a source of errors, requiring special attention. In contrast, our method is inherently robust against the occurrences, because it treats these specular highlights as *inverse occlusions*: instead of receiving no light at all, it keeps receiving light, even if the projected wave pattern was in a local minimum. This occurs when the frequency of the projected wave patterns is too high, resulting in enhanced colorbleeding artefacts by the specular highlight, or due to an oversaturation effect on the recorded camera signal.

As no local maxima are found in the region of the specular highlight, the computed visibility function $\overline{V}(t)$ will label this region as *occluded*. Subsequently, the values of the reflection function $R(t)$ in this region will be computed by interpolating between the two local maxima neighboring the highlight interval. This is illustrated in Figure 5.8. Thus, for all practical purposes we do not consider the highlight to be part of the computed reflection function, but rather as a special form of occlusion. From that point on in the pipeline, there is no difference between a specular highlight and a minor occlusion.

5.6.2 Influence of Interreflections and Stray Light?

In the preceding analysis, we have assumed a scenario in which all light arriving at the observed scene point originated from the translating projector. In other words: eq. (5.3) only accounted for direct illumination. In reality, additional terms $IR(t)$ and $S(t)$ should be added to this equation, describing the intensities related to interreflections and subsurface scattering:
\[ \mathcal{I}(t) = P(t) \cdot R(t) \cdot V(t) + IR(t) + S(t) \]  \hspace{1cm} (5.13)

However, these terms are relatively small compared to the main term described in eq. (5.3). As a result, it is justified to ignore them in a practical implementation.

Interreflections

Interreflections, second order reflections produced by another stripe pattern with another dominant frequency, are a potential source of noise in the frequency domain. However, we argue that their overall contribution to the observed signal is usually negligible compared to contribution of first order reflections, as long as the surface produces sufficient\(^{10}\) diffuse reflection. We base this argument on two observations.

Most importantly, the light intensities arriving at a fixed object surface point are structured in nature. To be more precise: the direct illumination arriving at the scene point, producing the first order reflections, is structured in nature: as time progresses, the horizontal ordinate of the illuminating projector pixel translates in a linear fashion. It is because of this structure that we are able to derive its depth, as the intensities produce a single dominant frequency. In contrast, the indirect illumination producing second and higher order reflections is scene-dependent. If we would trace back the horizontal ordinate of the illuminating projector pixel trough time, we will usually not find a linear progression similar to the one associated with first order reflections. This is due to the fact that under most circumstances, for \(N^{th}\) order reflections, the position of the \((N - 1)^{th}\) scene intersection does not move in a structured manner similar to the linear translation of our first order illuminator, the translating projector. Instead, this position, as well as the corresponding projector pixel ordinate, becomes randomized and scene-dependent. As such, their influence on the frequency spectrum can usually be regarded as little more than signal noise.

The second observation is the distance traveled by light of second order reflections is greater than the distance traveled by light from direct illumination. Furthermore, light reflected by a surface is only a fraction\(^{11}\) of the light that arrived at the surface. As such, the reflected intensities from second order reflections are commonly significantly lower than their first order counterparts\(^{12}\).

\(^{10}\)This excludes ‘perfect mirrors’ and transparent objects, such as glass.
\(^{11}\)For diffuse light, this fraction is also dependent on the angle between the surface normal and the direction of the incoming light.
\(^{12}\)This statement is true for opaque surfaces, with a minimal guaranteed diffuse component. For transparent objects and fluids, the statement becomes false due to effects such as caustics.
5.6 Influence of Scene Factors

In practice, we have noticed very few instances where the scene composition had a sufficient impact to produce false depth estimates due to second order reflections. The only significant occurrence of this was observed for scene points that were not part of the object surface, but were part of the floor on which the object resided. In this case, the angle between the floor’s surface normal $\eta$ and the direction of the incoming light $\omega_i$ from direct illumination was too steep, reducing the contribution of first order reflections to practically nil. At the same time, approximately planar surfaces from the scanned object produced sufficiently structured second order reflections to produce (somewhat consistent) false depth values for patches of the floor surface. However, it is important to note that the primary source for errors in this example was not the presence of second order reflections, but the lack of direct illumination due to a large angle $\eta \omega_i$.

The conclusion from this experiment was that, even though the vertical angle of inclination remains constant for direct illumination during horizontal movement of the projector, its orientation with respect to the object surface normal can still have an impact.

Subsurface Scattering

Subsurface scattering is a phenomenon in light transport, in which light enters the surface of a translucent object at one point, and exits the surface at a different point. Light will penetrate the surface and be reflected a number of times at irregular angles inside the material, after which it exits the material at an angle other than the angle it would have had if it had been reflected directly off the surface. Subsurface scattering occurs in many everyday materials, such as marble, skin, fruit, and milk.

Once again, we argue that the contribution from indirect illumination to the intensities due to subsurface scattering can be ignored. The reasoning behind this is similar to the interreflection case. While the influence of the indirect illumination to the total illumination reflected by the object surface is significant, it is not structured. By definition, the reflection paths by which light is transported within a scattering surface are highly irregular, preventing structured light transport from one or more entrance points to a single exit point. As a result, the influence of subsurface scattering on the frequency spectrum can be regarded as noise distributed over many different frequencies. For an example of a depth map obtained for objects with the subsurface scattering properties, we refer to Figure 5.11 (third row), in which we have scanned a selection of pieces of fruit.

---

13This adds additional degrees of freedom to the reflectance properties described by the bidirectional reflectance function described in section 5.2, which had only two parameters: the direction of the incoming light $\omega_i$, and the direction of the outgoing light $\omega_o$. 
5.7 Practical Considerations

The observed intensities function $I(t)$ discussed in the preceding analysis represents a discrete sampling of a continuous function $I(t)$, just like $\mathfrak{F}[I(t)]$ is a discrete sampling of its frequency domain representation, obtained using the Discrete Fourier Transform. Because we are dealing with these discrete functions instead of their continuous counterparts, we have to take several considerations into account if we want to maximize our depth resolution.

Spatial Resolution of our Stripe Pattern

The spatial resolution of our pattern needs to be maximized, under the following constraints:

- Projector defocus has to stay within reasonable limits for the depth range of the scanned scene.
- Projector translation speed has to be low enough, in order to stay above the Nyquist sampling frequency: we do not want to miss a stripe in our projected pattern, at any depth.

Translation Speed & Acquisition Speed

The speed of the projector translation needs to be minimized and constant, as it is directly responsible for the sampling frequency of the observed scene. Halving the speed of the projector results in doubling the number of captured frames over the same scanning volume, doubling the sampling frequency. Increasing the capturing speed of our camera results in a similar improvement.

Projection Defocus

Adding projector defocus to equation (5.3) introduces a smoothing kernel, depending on the points depth, to the wave pattern $P(t)$. As a result, the amplitude of the dominant frequency $f$ in $\mathfrak{F}[P(t)]$ becomes less pronounced as the level of blur increases. At a certain point, we will no longer be able to distinguish between the amplitude of the dominant frequency and random noise. Thus, projection defocus poses a physical limit on the range of depths we can measure using the described technique.
5.8 Projector Depth vs. Camera Depth

We have defined the depth \( \mu_i \) of an observed scene point \( i \) as its distance to the principal plane of the translating projector. This depth measure is very different from the conventional camera pixel’s depth \( \lambda_i \). In this section, we will describe how the two are related, and can be derived each other.

**Degrees of Freedom**

The first thing to note is that unless any initial information is known about the principal plane \( \pi(a,b,c,d) \) of the sliding projector, we cannot derive the set \( \{ \lambda_i \} \) from \( \{ \mu_i \} \), because the number of unknowns \( \{ \{ \lambda_i \}, a,b,c,d \} \) will always outnumber the number of equations (1 per \( \mu_i \)) by 4. On the other hand, if these plane coordinates \( (a,b,c,d) \) with respect to the camera are known, the camera pixel depths \( \lambda_i \) are uniquely defined by the point-plane distance equation:

\[
\lambda_i = \frac{\mu_i \cdot \sqrt{a^2 + b^2 + c^2} - d}{ax_i + by_i + c} \tag{5.14}
\]

where \((x_i,y_i)\) represents the pixel’s normalized camera coordinates, assuming the camera’s intrinsic parameters are known.
Consequences for the Setup

In practice, this means that in order to obtain camera depth values instead of projector depth values, we have to either (a) perfectly align the camera’s principal plane with that of the sliding projector, or (b) restrict our setup to the use of an orthonormal camera, and align its principal plane with a plane parallel to the principal plane of the sliding projector. In the latter case, $\lambda$ becomes equal to $\mu$, plus an unknown constant. However, for the purpose of reconstructing a point cloud, this constant is irrelevant and can be ignored.

5.9 Discussion

In this section, we will first illustrate some of the results produced by our technique, followed by a comparison of the proposed method to the state of the art as described in chapter 3.

5.9.1 Results

Results without Interpolation

Results from our method are shown in Figure 5.11. Each row contains information of the same scene. In the first column, we show a color image of the maximal intensity observed at each pixel. The second column shows a depth map produced by our method. The third column provides a magnified zoom, focusing on certain details provided in the second column. And finally, in the fourth column we rerender the original scene in a variety of ways.
All depth maps in Figure 5.11 were computed using the naive Fourier frequency estimation method, as described in section 4.1.2. As no Fourier interpolation is applied, this results in a finite set of depth planes. Still, due to the large number of frames (±2000), the depth resolution is more than sufficient in these cases.

The first two rows show a dense scanning of approximately Lambertian surfaces. In the first example we have used a wide stripe pattern and a high resolution, in order to capture a large range of depths. Alternatively, the second example shows a close-up scan of the middle figurine, in which we have applied a finer stripe pattern. Because of the increased spatial resolution of the pattern, the corresponding discrete set of depth planes is now concentrated within a smaller depth range.

The third row shows a selection of fruits, objects with strong specular and translucent properties. The smooth gradient progression and dense coverage of each object surface illustrate our technique’s potential for scanning materials with the subsurface scattering property. Notice the captured level of fine detail, as well as the low amount of minimally required diffuse reflection in the crown of the apple, where the stem has a color\(^\text{14}\) approaching black and a thickness of only a few pixels.

Finally, in the fourth row, we display a fan segment with a very fine grid structure. While the blade itself is fairly diffuse, it can be seen in the bottom right of the color image that the outer layer is actually fairly specular\(^\text{15}\). Even though it is difficult to discern from this image, there is also an outer grid behind the fan blade. In contrast, its presence is perfectly clear in the depth map computed by our method. This example illustrates the proposed algorithm’s ability to preserve sharp edges, its robustness in the presence of occlusions\(^\text{16}\), as well as its ability to cope with specular materials.

**Comparison of Results without Interpolation to Ground Truth**

For a comparison of our technique against a ground truth, we refer to Figures 5.12 and 5.13, in which we compare our obtained depth values against the values returned by our synthetic dataset. The first dataset was generated with Blender using a simple diffuse shading model, producing 2000 frames. The second dataset was generated at a later date with Maya, containing a more complex scene, containing subsurface scattering (marble sphere), highly specular surfaces with large highlights (the cylinder), partial transparency (the cone), and specular mesostructures (bump mapped cube). This latter example was generated with only 200 frames, and analyzed both with and without interpolation enabled. In both case, we have placed

\(^{14}\)The maximal intensity value of its pixels throughout the entire captured sequence.

\(^{15}\)The white blob in the color image indicates the presence of a fairly large specular lobe.

\(^{16}\)Each part of the grid occludes the fan blade for a short period of time.
the projector above the projector translation trajectory, and aligned its principal plane with that of the projector. As this is a synthetic dataset, there are no precise units assigned to the graph, but it should be clear that the recovered shape very closely approximates the actual synthetic z-buffer depth values, and shows no problems handling difficult materials. Furthermore, it should be noted that doubling the number of frames in these synthetic datasets always increases the accuracy of the results twofold\textsuperscript{17}.

\textsuperscript{17}As synthetic datasets allow us to always perfectly define the path, without any jitter on the horizontal coordinate of the projection center, we can always achieve the theoretical 'perfect' acquisition limit by simply adding more frames. Not that this is not possible with classical structured light methods, but rather a feature of our method.
5.9 Discussion

Figure 5.12: Comparing our results on a synthetic dataset with ground truth. (top) The (reconstructed) fully lit image. (middle) The depth map produced by our technique. (bottom) The depth map of the corresponding ground truth. (right) The plot of a single scanline at the same vertical coordinates in both images.

Processing Speed

In order to process the results shown in Figure 5.11, our unoptimized Matlab application required about 5 to 10 minutes on average. The bulk of this processing time was allocated to swapping data between hard drive and memory. However, we believe that with an efficient C++ implementation, which would retain the images in memory during capture, and should contain a per pixel GPU-based CUDA processing of the observed intensities, processing speed can be approximately reduced to approximately half the minute\textsuperscript{18} it takes for the projector to perform its translation.

\textsuperscript{18}With a precalibrated, custom-made setup consisting of large number of point lights behind a single slide, the recording speed itself could also be reduced to a fraction of its current state. Thus theoretically, our proposed technique should be able to reach interactive speeds of a few seconds.
Figure 5.13: **Comparing our results on another synthetic dataset with ground truth.** *(top)* One of the 200 frames in the sequence. *(middle top)* The depth map produced by our technique, without interpolation. *(middle bottom)* The depth map produced by our technique, with interpolation. *(bottom)* The depth map of the corresponding ground truth, according to the z-buffer provided by Maya. *(right top)* The plot of a single scanline at the same vertical coordinates in both images. *(right bottom)* The corresponding error function. We believe the large peaks at the object edges are most likely caused by anti-aliasing effects in the rendering process.
5.9 Discussion

Results with Interpolation

In the previous section, we have illustrated the minimal level of accuracy that can be attained with our technique, as well its ability to cope with a variety of materials and scene conditions. In this section, we will discuss the potential improvements gained by applying Fourier interpolation to the recovered signal, illustrated by Figure 5.14.

For each example, we have provided a color (maximal intensity) image, and the depth maps computed by our method without and with interpolation. Each example uses fewer samples than it would need to guarantee a sufficient depth resolution without Fourier interpolation, in order to illustrate the improvements. To summarize what we have discussed in sections 4.1.2 and 5.4: depth maps without Fourier interpolation produce a discrete set of depth values

\[ d = \frac{s}{f} \]  

where \( f \) is a single integer frequency from the power spectrum of the recovered signal. In the case of Fourier interpolation, eq. (5.15) instead becomes

\[ d = \frac{s}{f + \delta} \quad \delta \in \left[ -\frac{1}{2}, \frac{1}{2} \right] \]

As a result, the entire continuous depth spectrum is now covered, enabling us to recover much finer detail.

In the first example, we have rendered 1000 frames of the synthetic Stanford bunny dataset, and provided depth maps computed with and without interpolation. The increased accuracy is most easily observed when looking at the fine detail on the ears and the patches of fur on the chest, captured in the refined depth map.

In the second example, we have used 1000 frames of our house dataset to render the preliminary depth map, as shown in the center image. As this is an object with many fine details, the low number of frames/samples results in a discrete set of clearly distinguishable depth planes. In comparison, by using a significantly lower amount of frames (150 \( \ll \) 1000), we are able to capture the vast majority of fine detail, such as the relief on the door and windows, and the tiles on the roof, on a scale of a few tenths of a millimeter. For a close-up view of these depth maps, we refer to Figures 5.15 and 5.16.

---

\[ \text{After undoing the influence of the reflectance function.} \]
Figure 5.14: **Influence of Fourier interpolation on the accuracy of our depth estimates.**

*Left* Color image of the set; *center* Unrefined depth map, with discrete depth planes; *right* Refined depth map, using frequency interpolation.

### 5.9.2 Comparison to the State of the Art

**Situating our Method**

We will initiate this section by situating the method presented in this chapter within the broad spectrum of methods that we have discussed in chapter 3. Like all those approaches, our method makes use of active illumination, provided in the form of a static periodic wave or stripe pattern emitted by a single projector.

We do not use multiple patterns over time, and as such our method cannot be classified as temporal structured light. Furthermore, the computations performed by our algorithm occur strictly on a per pixel basis. Additionally, as our projected pattern is periodic in nature, not a single projector pixel is uniquely defined by its spatial neighborhood. Thus, our method does not fall under the category of spatial structured light methods either.
If the new algorithm was to be classified under a single category in the presented
taxonomy, it should be labeled as a viewpoint encoding scheme. One of the key
components of the method presented in this work, mobile structured light, is the
fact that the projector is mobile. Depending on the recording time, the observed
intensities were projected from a different viewpoint, and the progression of this
change in viewpoint happens in a structured manner. For this reason, we classify our
method as a dual viewpoint encoding scheme.

Comparison to Active Methods

Most active shape acquisition methods recover scene depth the same way their
passive counterparts do, i.e. by establishing feature point correspondences over two
or more views, and triangulating the corresponding scene point. The main difference
lies in the manner the projector pixels are encoded. If we compare the proposed
method against these static viewpoint approaches, we notice several advantages over
these classical approaches.

Because of the limited amount of viewpoints, feature-based methods will almost
never be able to provide a depth estimate for every pixel in the camera view. There
will always be points that are occluded in one or more views. As the number of views
increases, the number of occlusions will decrease. As in our technique the scene is
illuminated in a structured manner from many different viewpoints, the chance that an
observed scene point remains non-illuminated for the entire duration of the captured
sequence is very low.

This same data and viewpoint redundancy is responsible for our technique’s
robustness in the presence of interreflections and subsurface scattering effects.
E.g. in classical temporal encoding schemes, the illumination source’s position
remains static throughout the entire sequence, while the intensity changes in time. As
a result, the light rays that are scattered within the surface only change in intensity,
but the path (and thus, the exit points) remain the same. This produces erroneous
labellings, as the exit points are wrongly recognized as a directly illuminated scene
point. In our method, the illumination source’s position changes constantly over
time, distributing the produced noise over many different frequencies, rendering its
influence on the computed depth value negligible.

\(^{20}\)The original viewpoint coded structured light paper assumed the cameras to be located at different
specific viewpoint, and a single projector illuminating the scene. In our case, we have the dual
case: a single camera observing the scene, and a set of projectors at different viewpoints illuminating
the scene.

\(^{21}\)There are of course exceptions to this, e.g. depth from camera defocus methods.

\(^{22}\)E.g. Temporal structured light encodes each pixel by unique intensities over time.

\(^{23}\)The only exception here being the case where the scene is spherical, and no self-occlusions occur.
Another advantage of our method is that it all computations are performed on a per pixel basis. This makes it suitable for parallelization, and enables us to scan very thin structures, like the fan grid in Figure 5.11. This would not be possible with classical spatial encoding methods, that require a spatial neighborhood in order to recover the correct projector pixel label.

Our method has many advantages in common with the focal encoding schemes in section 3.7, which produces a similarly dense estimate of the depth map. Recent projector defocus methods, such as the work of Gupta et al. [Gupta 09], rely on a large amount of projector focal planes to recover an estimate of the scene. The accuracy of the depth estimates is largely dependent on the distance between the scene points and the closest projector focal plane. The method also introduces what can be referred to as striping artefacts, which appear to be related to the emitted stripe pattern during each focal stage. Furthermore, the recovered depth maps appear to be fairly planar in nature. In comparison, our proposed method is able to capture a finer level of detail, despite its significantly lower complexity.

Comparison to Passive Methods

As shall be the object of discussion in chapter 7, our method can be perceived as the active counterpart to the technique known in passive stereo by the name of epipolar-plane image analysis. Like EPI analysis, our method represents an alternative to sparse feature-based methods, as it is able to produced a depth estimate for each pixel, assuming it was illuminated for a sufficient amount of time. For both methods, the large amount of redundant data commonly assures us this to be the case. As such, our method can be classified as a dense shape acquisition algorithm.

Our Method ≠ Phase Shifting

At first sight, our technique could be mistaken for a phase shifting technique. However, there is fundamental difference between our method and phase shifting techniques: (a) phase shifting techniques recover depth by labeling projector pixel positions, based on the computed phase of the illuminated pixels, and triangulating the depth of the corresponding scene point; (b) mobile structured light performs no triangulation, but instead it exploits the linear relation between the observed dominant frequency and the depth of the scene point.

---

24This is most apparent in Figure 3.18, where both the depth maps of the single and multi-focal methods appear to consist of large planar regions, even though the corresponding rendering of the scene seems to suggest otherwise.
5.9 Discussion

5.9.3 Limitations and Future Work

In this section, we will discuss the limitations of our method, followed by a short investigation of the potential future directions research on this topic could take.

Limitations

Like many other active shape acquisition methods, the proposed approach is unable to capture the shape of transparent solids (e.g., glass) or fluids. In the case of a missing (or insufficient) diffuse component, there will be no detectable reflected pattern visible to the camera, and the corresponding depth cannot be established.

The other main limitation is the position of the camera, with respect to the translating projector. As we have discussed in section 5.8, we require specific projector/camera configurations in case we want to recover the depth values with respect to the camera, instead of with respect to the projector. Solving this limitation will be the central theme of our next chapter.

Any other limitations are mainly of practical nature, e.g., depth resolution is dependent on the number of samples. If the translation of the projector contains too much jitter, this will have an influence on the true resolution of the capture depth map. Therefore, it is desirable that we have an estimate of the size of the translation error, to estimate a maximum error bound. We have described the different factors that influence the practical setup in section 5.7.

Future Work

One recent interesting topic that lends itself well to the proposed method is the coded strobing approach of Veeraraghavan et al. [Veeraraghavan 10], which shows that, by applying temporal modulation, one can observe a high-speed periodic event well beyond the abilities of a low-frame rate camera. Their key result is that for such signals the Nyquist rate constraint can be imposed on the strobe-rate rather than the sensor-rate.

Coded strobing naturally extends to our own proposed acquisition method, as it addresses the main bottleneck of our system: camera frame acquisition speed. If we opt for a setup with a continuously moving projector, increasing this acquisition speed will increase the number of samples, which in turn increases the accuracy of the detected fundamental frequency and corresponding depth. The fact that coded strobing does not perform real-time processing is not an issue, as the produced interactive speeds are in the same category as our own method.

25Note that we are talking about applying coded strobing to the continuous case, which automatically assumes large (500+) sample sizes.
One minor difference between the goal of coded strobing and our own method is that we are really only interested in recovering the fundamental frequency $f_P$, which is related to point depth, and do not necessarily require an completely accurate reconstruction of the full signal $x(t)$. In other words, for our purposes it is sufficient to establish which are the non-zero elements in $s$, rather than actually recovering the exact values of these elements.

Employing coded strobing as an integral part of our setup poses some potential practical difficulties, mainly due to its incompatibility with the extensions described in the next chapter, where we introduce the use of an additional pattern.

### 5.10 Conclusions

In this chapter we have introduced a novel approach to active shape acquisition, using a mobile projector emitting a static periodic pattern. Our method is able to compute a dense depth map on a per pixel basis, by casting depth estimation as a signal processing problem, \textit{i.e.} frequency estimation. We have illustrated its accuracy and robustness in the presence of a wide variety of material and scene properties, as well as its practical and theoretical limitations.
Figure 5.15: **House dataset: unrefined depth map.** This depth map was acquired using 1000 samples, recovering the dominant frequency using the naive method.
Figure 5.16: **House dataset: refined depth map.** This depth map was acquired using only 150 samples, recovering the dominant frequency using the Fourier interpolation method. Many fine details, such as the tiles on the roof, the bars on the windows, and the lion in the bottom center, are now visible.
In the previous chapter, we have introduced a novel shape acquisition method, which utilizes a mobile projector emitting a static periodic pattern. By observing the reflected intensities of a fixed surface point over time, we were able to recover its depth with respect to the projector\(^1\). Furthermore, in section 5.8 we have shown these projector depth values to be equivalent to the more conventional camera depth values only under special camera configurations\(^2\). In practice, calibrating a setup to meet these configuration constraints has proven to be difficult. In order to relieve this constraint, we have extended our method by introducing an additional pattern for calibration purposes. This will facilitate the use of our method for more general camera/projector setups.

The remainder of this chapter will follow the structure of the algorithm, followed by some discussion: (a) introduction of the new pattern, which uniquely labels a sparse set of projector pixel coordinates; (b) performing a sparse reconstruction of the scene, based on the labels from the new pattern, and the projector depths from the original pattern; (c) establish camera depths for a dense reconstruction, computing the camera matrix from a sparse set of 2D-3D correspondences.

### 6.1 Labeling the Projector Pixels

#### 6.1.1 Temporal Encoding

In the previous chapter, we have discussed our method for recovering projector depth estimates for a dense set of scene points. In this chapter, we want to convert these into corresponding camera depth estimates, which will require extrinsic calibration of the setup from the observed data. Initially, we want to retrieve the projector’s position at each time step.

In order to achieve this, we need to have a set of projector points which we can track throughout the entire sequence. This requires uniquely encoding a sparse set of spatial positions in the illumination pattern in such a way that, based solely on the analysis of the temporal intensities, camera-projector correspondences can be inferred. In order to establish this mapping from a 1D temporal code to a 2D spatial position, we make use of one-dimensional De Bruijn sequences (see section 3.3.1).

\(^1\)To be more specific: projector depth is defined as the distance between the fixed surface point and the mobile projector’s principal plane. As the projector only performs a horizontal translation at a constant speed, its principal plane remains constant for the entire duration of the capture.

\(^2\)More specifically: for projective cameras, the camera’s principal plane and the projector’s principal plane must coincide; for affine cameras, the camera’s principal plane and the projector’s principal plane must be parallel to each other.
6.1 Labeling the Projector Pixels

Figure 6.1: Encoding 2D projector pixel positions with a 1D De Bruijn sequence. (left) 1D sine wave pattern used to recover point-projector depth estimates; (center) 2D spatially encoded pattern; (right) close-up of two patches of both patterns, in which it becomes clear the employed binary code is mapped onto the pre-existing wave pattern; (bottom) 1D De Bruijn sequence used to spatially encode a sparse set of 2D positions in the center image. Note that each of the encoded 2D positions is encoded by a unique $16 \times 1$ window which maps to a unique 16-bit subsequence in the De Bruijn sequence.

6.1.2 Signal-to-Code Conversion

In order to convert a temporal intensity signal into a corresponding binary code, we need to establish the beginning and end of each single bit in the observed code. To this end, we have opted to map the De Bruijn sequence onto the same wave pattern we have used to establish point-projector distances (see Figure 6.1). Each recorded wave has a known\(^3\) period $p$, directly related to point’s distance to the projector’s principal plane. Additionally, in order to avoid aliasing effects due to light bleeding from one line of code into another, we have opted to introduce horizontal interlacing into the pattern.

Assuming we are able to simultaneously observe both the wave pattern and the De Bruijn pattern, we are able to locate the timesteps at which the individual bits of the De Bruijn code are visible at the local maxima of each wave period. Comparing the intensity value of the observed De Bruin signal to the local extrema within the observed wave period provides us with a simple and robust method for resolving the De Bruijn code. If the observed De Bruijn signal is sufficiently close to the local maximum, we assign a value of 1, otherwise we assign a 0.

\(^3\)In this chapter, we assume that the steps described in the previous chapter have already been completed. Thus, in order to establish the depth of the observed surface point, the dominant frequency $f$ of the reflected wave pattern has already been established. As such, the period $p = \frac{1}{f}$ of the fitted De Bruijn signal is also known.
6.1.3 Time Multiplexing

In the previous section, we have assumed the simultaneous observation of both the wave pattern and the De Bruijn pattern. In practice, a projector can only emit one illumination pattern at a time. We simulate this by time multiplexing the patterns displayed by our projector, alternating between the wave and De Bruijn pattern at the same frequency as the camera, and demultiplexing the observed intensities into two separate signals. Missing values can then be interpolated from the available data if needed be. This is shown in Figure 6.2.

6.2 Recovering Structure and Projector Motion

Once we have a sparse set of camera-projector pixel correspondences, the next step is to perform a sparse reconstruction of the scene. For the remainder of this section, we will assume the intrinsic parameters of the projector are known\(^4\).

For each scene point \(X^i = [x^i, y^i, \mu^i]^T\), we have a set of projector pixels \([u_t, v_t, 1]^T\):

\[
P_tX^i = \begin{bmatrix} 1 & 0 & 0 & -\lambda t \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x^i \\ y^i \\ \mu^i \\ 1 \end{bmatrix} = \begin{bmatrix} x^i - \lambda t \\ y^i \\ \mu^i \end{bmatrix} \propto \begin{bmatrix} u_t \\ v_t \\ 1 \end{bmatrix} \tag{6.1}
\]

From eq. (6.1) one can see the only unknown parameter in this equation is \(\lambda\), related to the projector’s speed. An initial estimate of this parameter is computed by averaging over all \(\lambda_{irs}\) values, \(r\) and \(s\) being random timesteps at which point \(X^i\) registered known projector pixel positions \([u_r, v_r, 1]^T\) and \([u_s, v_s, 1]^T\), where

\[
\lambda_{irs} = \mu^i \frac{u_r - u_s}{s - r} \tag{6.2}
\]

Using the initial estimate, we employ Levenberg-Marquardt minimization to find the optimal \(\lambda\) which minimizes the overall reprojection error of all \(X^i\) for all \(P_t\). This results in a sparse reconstruction of the scene, e.g. as depicted in Figure 6.3.

\(^4\)During our experiments, we computed the intrinsic parameters of the projector by employing the multi-view reconstruction toolbox of Svoboda et al. [Svoboda 02, Svoboda 05]. By employing temporal encoding schemes, such as those described in section 3.2, we established camera-projector pixel correspondences, which serve as input for their algorithm.
6.2 Recovering Structure and Projector Motion

Figure 6.2: **Time multiplexing.** The observed intensities in the top sequence, contain information of two separate signals: the scene point illuminated by the wave pattern (bottom sequence, blue), and the scene point illuminated by the De Bruijn pattern (bottom sequence, red). Separating these signals, and interpolating the missing information allows for (1) frequency/period estimation on the wave pattern, and (2) recovering the binary code from both signals (e.g., the depicted sequence clearly encoded the sequence 100101011). In the example above, the De Bruijn pattern is projected at even time steps, while the wave pattern occurs at odd time steps.

Figure 6.3: **Recovering sparse structure and projector motion.** (left) A random image from our Stanford bunny dataset, with the superimposed sparse set of identified projector pixels at that time step. The color code represents the projector pixel’s horizontal position, from left (yellow) to right (red). (right) Visualization of the computed sparse reconstruction, based on the information from the projector pixel correspondences.
6.3 Recovering Camera Parameters & Dense Depth Map

For each point \(X^i\) of the sparse dataset, we have the corresponding camera pixel coordinates \([u^i, v^i, 1]^T\). From this set of correspondences, we can estimate the parameters of the camera matrix, using one of many available methods [Hartley 04].

Once the position and orientation of the camera matrix are known\(^3\), we can convert the dense map of point-projector distances into a corresponding camera depth map, by intersecting each camera pixel’s back-projected ray with the plane \(\pi(0,0,1,-\mu)\).

6.4 Calibrating the Setup

The initial per pixel depth computation, as briefly described the previous chapter, assumes that the scaling factor \(s\) in eq. (5.15) is known. However, in practice it is not always easy to accurately recover this constant, as it is dependent on many parameters of both the projector and the camera.

However, by scanning a single sphere of a known radius, our technique is able to accurately recover this scale factor. Each choice of the parameter \(s\) gives rise to a sparse set of 3D points \(X^i = [x^i, y^i, s]_i^T\). By fitting a sphere through these points, we can compute the overall error for each instance. Applying iterative minimization of this error allows us to quickly find the optimal scale factor.

6.5 Discussion

There are several aspects of to our method that require verification: (1) the robustness of the depth estimation component for different material properties; (2) the influence of time multiplexing to these depth estimation results; (3) the robustness of the signal-to-code conversion algorithm; and (4) the quality of the estimated reconstruction. As the first of these four items has already been the subject of discussion in chapter 5, we will focus in this chapter on the other three items. Results from our method are shown in Figure 6.4. All results were generated with 1500 frames or less, using patterns with a wave period of 8 projector pixels.

\(^3\)In this chapter, the camera’s position and orientation are computed with respect to known projector and surface point coordinates. This is a different reference frame from the coordinate framework used in section 5.8, where we define a projector-to-camera depth conversion in terms of a camera in the origin.
Figure 6.4: **Camera depth results.** A set of results produced by our method, each consisting of four columns: (1) the fully illuminated scene; (2) the estimated dense depth map, computed from the projector’s perspective; (3) the computed sparse reconstruction; (4) the converted depth map, with respect to the camera instead of the projector. All results were generated with 1500 frames or less, using patterns with a wave period of 8 projector pixels.
6.5.1 Time Multiplexing

Compared to previous work, time multiplexing effectively cuts our amount of potential depth planes in half\(^6\). In order to combat this issue, we could add color multiplexing to our method, alternating between the R/GB and GB/R channels for the wave/spatial code patterns. Theoretically, this could restore our depth resolution to previous levels under most reflectance functions. Practically, this just adds an additional level of complexity to the system. This may not be needed, as Fourier interpolation seems to produce good results, but we have yet to further explore this option.

6.5.2 Signal-to-Code Conversion

Our experiments have indicated that, in order to avoid mismatches, we need to use tri-state codes instead of binary codes, assigning 'unknown' bits in the code when there is no clear inclination towards a local minimum or maximum. A simple backtracking

---

\(^6\)For \(N\) captured frames, \(N/2\) observe the reflected wave pattern, and \(N/2\) observe the reflected De Bruijn pattern. Thus, the initial depth estimate (without applying Fourier interpolation) will only have a resolution of \(N/4\) integer frequency values, instead of \(N/2\) as in the previous chapter.


6.5 Discussion

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Bunny</th>
<th>4 Objects</th>
<th>Sphere</th>
<th>Flag</th>
<th>House</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean Error</td>
<td>0.997067</td>
<td>0.992447</td>
<td>1.092911</td>
<td>0.858623</td>
<td>1.042102</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>0.012627</td>
<td>0.016163</td>
<td>0.014703</td>
<td>0.014192</td>
<td>0.014237</td>
</tr>
</tbody>
</table>

Table 6.1: Projector pixel reprojection errors for the displayed datasets. Note that these errors are for superpixels consisting of $8 \times 8$ pixels, and should be interpreted accordingly.

algorithm suffices to uniquely identify the observed code. However, as in practice mismatches are impossible to avoid, it is still highly recommended to apply outlier rejection during the sparse reconstruction phase following this labeling step.

**Backtracking Algorithm Example**

In this section, we will illustrate our tri-state code detection backtracking algorithm by example. After comparing the intensity values of an observed wave and De Bruijn pattern, the value of bits 3 and 7 were unclear:

$$f(x) = 11?101?0$$  \hspace{1cm} (6.3)

where $x$ is the projector pixel position of the first bit, and $f$ is the mapping between position and De Bruijn code. Thus, recovering position $x$ is equivalent to finding $f^{-1}(11?101?0010100011)$. Unfortunately, the 3rd and 7th bit are missing, so we need to recover them. There are four possible candidates:

$$f(x) \in \{11110110, 11110100, 11010110, 11010100\}$$  \hspace{1cm} (6.4)

In most cases, the observed De Bruijn signal will contain more bits than the minimal window length. For each of the candidates in eq.(6.4), we can use these bits as control bits to verify if they correspond to true segments of our generated De Bruijn code. This can be done either in a breadth-first manner, or a depth-first manner (backtracking). We have opted for the latter, as in backtracking it is easier to deal with cases when the new bits does not contain a clear 1 or 0.

6.5.3 Estimated Reconstruction

The error metric we discuss in order to validate our estimated reconstruction is the (projector pixel) reprojection error. Using a projector stripe width of 8 pixels, we are able to encode $64 \times 4$ different projector pixel positions. After estimating the sparse reconstruction, these positions are reprojected back to the projector's image plane at each time step, with an average reprojection error of approximately a pixel for each block of $8 \times 8$ pixels per position (see table 6.1).

---

7 As illustrated in Figure 6.1, in our experiments we used a window length of 16.
6.6 Conclusion

In this chapter, we have lifted the largest practical constraint for the use of the method presented in the previous chapter. We have achieved this by introducing a new pattern to our setup, a 1D De Bruijn sequence, fitted onto our previous 2D wave pattern. This new pattern uniquely labels a sparse set of projector pixel locations, enabling us to establish at which point in time an observed surface point was illuminated by which projector pixel. Subsequently, from this sparse set of 2D-3D correspondences, we show how to recover the corresponding point cloud, together with the associated projector matrices. This establishes a clear coordinate system in which structure and motion are defined. Within this coordinate framework, we then proceed to compute the camera matrix using one of the many available methods known in literature. Finally, we convert the projector depth values of the dense depth map into corresponding camera depth values, based on the calibration information gathered in the previous steps. We have discussed the different components of this extension to our algorithm, and performed an evaluation our method on both synthetic and real-world data.
Chapter 7

Epipolar Plane Image Analysis & Mobile Structured Light

Contents

7.1 Related Work ........................................ 96
  7.1.1 Sparse Feature Based Methods ............... 96
  7.1.2 Dense Plane-Sweeping Methods ............... 97
  7.1.3 Triangulation vs. Plane-Sweeping in Active Vision ...... 98
  7.1.4 Viewpoint Encoding Equivalent in Passive Stereo? ...... 98
7.2 Epipolar Plane Image Analysis: A Dense Alternative ...... 98
  7.2.1 What is EPI Analysis? ........................... 98
  7.2.2 From Classical Stereo to EPI Analysis ............ 99
  7.2.3 Matching Criteria ................................. 100
  7.2.4 EPI Structure and Occlusions ................. 100
  7.2.5 Dense Reconstruction ......................... 101
7.3 From EPI Analysis to Mobile Structured Light .......... 105
7.4 Beyond Linear Camera (and Projector) Paths ........... 106
  7.4.1 Circular Camera Trajectories .................. 106
  7.4.2 Circular Projector Trajectories ............... 111
In this chapter we will discuss a dense reconstruction algorithm from the class of passive shape acquisition methods, known as epipolar plane image analysis. Furthermore, we will illustrate how this algorithm relates to the method we have presented in this dissertation. After we have discussed the basic algorithm, we will take a look at the extensions that have been made during the years that followed the initial algorithm, and how these extensions could influence potential future directions for our own work, mobile structured light.

7.1 Related Work

Passive shape acquisition methods, such as those based on stereo and structure from motion, have been the studied extensively by the computer vision community for decades, and still remain an active research topic today. In spite of the progress made in this area, 3D scene reconstruction from multiple images or video feeds still faces many difficulties, especially in dealing with occlusions, partial visibility, and textureless regions. In the remainder of this section, we will briefly discuss both the classical sparse triangulation-based methods, and the dense plane-sweeping methods, after which we conclude with the dense reconstruction method that is the topic of this chapter: epipolar plane image analysis.

7.1.1 Sparse Feature Based Methods

Like we have mentioned in section 3.1, the most frequently used approach to passive scene reconstruction comes in the form of multi-view stereo based on feature correspondences. Such feature based stereo algorithms are based on the assumption that the raw image or video data provided by the sensor is insufficiently structured, requiring one or more initial low-level image processing steps to produce a more meaningful representation for the high-level stereo algorithm to work with. By looking for salient image regions, such as corners [Mikolajczyk 04] or blobs [Lindeberg 98], these low-level region detectors collect a set of feature points, each represented by a descriptor commonly invariant to rotation and scale transformations [Lowe 99]. This set of interest points completely describes (the visible part of) the observed object. Once such a collection of salient features is acquired for each image of the scene, the associated descriptors are matched across images in order to establish feature correspondences. As not every correspondence

---

1This forms the basis of the bag-of-features metaphor commonly used in object classification [Csurka 04, Willamowski 04], unsupervised discovery of categories [Fei-Fei 05, Quelhas 05, Sivic 05], and video retrieval [Sivic 03], similar to the bag-of-words model used in text document classification [McCallum 98, Nigam 99]
from this initial matching step is a good one, one or more outlier rejection steps need to be included. The most commonly used outlier rejection step is based on random sampling consensus [Fischler 81], or one of the many specialized\footnote{Many RANSAC derivatives exist. Depending on special requirements (e.g. restricted computation time [Nistér 03, R. Raguram 08]) or partly known scene structure (e.g. the presence of lots of planar structures in the scene [Chum 05] or other potential degenerate configurations [Frahm 06]), specific algorithms have been developed. Others have aimed to simply improve the algorithm overall, e.g. by using spatial constraints to improved the sampling procedure [Ni 09, Sattler 09].} RANSAC derivatives. Based on the set of inliers, multi-view stereo algorithm simultaneously derive a set of camera positions and orientations and a consistent 3D point cloud of the scene, using iterative minimization techniques like Levenbergh-Marquardt. In case of extremely large scale reconstruction problems, e.g. city reconstruction based on a large photo-collection, specialized parallelizable large scale optimization techniques need to be derived to handle the amount of data [Agarwal 09].

### 7.1.2 Dense Plane-Sweeping Methods

There are two essentially two different ways for shape acquisition algorithms to improve the density of the recovered point cloud: (a) increase the amount of views of the scene, or (b) increase the number of matchable pixels per view. Whereas the trend for sparse methods like the ones we have discussed in the previous section seems to be leaning towards the first option, dense plane-sweeping methods aims at providing an alternative, by assigning a depth value to each pixel in a chosen viewpoint.

We loosely apply the term plane-sweep approaches to all algorithms that specify a discrete set of depth fields, which are evaluated on a per pixel basis, using an implicit or explicit cost function. After evaluating this cost function for every depth value, the plane with the minimal cost is assigned to the pixel. This class of passive stereo approaches is named after the seminal work of Collins [Collins 96], in which a discrete set of depth planes are defined at specified distances to one central camera for which the depth map must be computed. For each possible depth value, the observed camera intensities in each camera are projected back onto corresponding depth plane in space. For each pixel of the virtual camera, the color correspondence forms the basis for the cost function: the depth value for which the color dissimilarity is minimal is assigned to the pixel. Throughout the years, many extensions to this approach have been presented and implemented. Because of the per pixel nature of the computations, it lends well to parallelization, and as such it has been implemented for real-time applications on commodity hardware [Yang 02, Yang 03, Dumont 08].
7.1.3 Triangulation vs. Plane-Sweeping in Active Vision

It should be noted that the proposed grouping of methods for passive multi-view stereo in this chapter could also be applied to the taxonomy on active vision we have proposed in chapter 3. However, it would make less sense to use it there, as it provides less distinction between the different methods. Temporal and spatial encoding schemes, such as Gray codes, phase shifting, and De Bruijn codes are essentially triangulation-based methods, as they provide a unique labeling between different views\(^3\) which corresponds to the matching stage in the sparse multi-view stereo algorithms of section 7.1.1. On the other hand, most depth-from-(de)focus algorithms are essentially plane-sweep algorithms that compute the blur kernel at different stages, sweeping through the scene to find the sharpest image for each pixel.

7.1.4 Viewpoint Encoding Equivalent in Passive Stereo?

When converting the active shape acquisition taxonomy into an passive equivalent, so far one class of approaches has remained undiscussed: viewpoint encoding schemes. Such methods employ known camera positions as an additional cue for depth reconstruction, e.g. Young \textit{et al.} [Young 07] use known camera configurations as a substitution for temporal patterns, whereas our method uses the linear structure of the projector path to cast depth estimation as a signal processing problem. In passive stereo, similar methods exist for moving cameras with a known camera paths. These will be the topic of the remainder of this chapter.

7.2 Epipolar Plane Image Analysis: A Dense Alternative

7.2.1 What is EPI Analysis?

Epipolar plane image (EPI) analysis is the name of a dense structure from motion technique proposed by Bolles \textit{et al.} [Bolles 87] in 1987. Their algorithm, in which a camera is translated horizontally in a straight line at a constant speed, is based based upon the analysis of the discrete rectified spatio-temporal volume constructed by concatenating all recorded images into a single entity. This volume, referred to as the EPI volume [Criminisi 05] or image cube [Feldmann 03a], has a large amount of inherent structure\(^4\). This structure becomes apparent as we investigate the \textit{epipolar plane images}, horizontal slices of the EPI volume with a fixed height value. The structure of the EPI volume is illustrated in Figure 7.1.

\(^3\)In this case, the projector functions as an inverse camera.

\(^4\)This volume is equivalent to an orthogonal 3D slice through the 4D lightfield of a scene [Sloan 97].
7.2 Epipolar Plane Image Analysis: A Dense Alternative

Figure 7.1: **The epipolar plane image volume.** (left) The EPI volume is created by concatenating all recorded images into a single spatio-temporal volume. So by taking a time slice of the EPI volume, a surface with a fixed time value $t$, we recover the image taken at a given time $t$. (right) If we take a horizontal slice, a surface with a fixed height value $y$, we recover an epipolar plane image [Feldmann 03a].

7.2.2 From Classical Stereo to EPI Analysis

In the classical (rectified) stereo problem, we have two parallel cameras observing the scene. For a single point correspondence between both views, the depth of the imaged scene point is determined by the pixel disparity between both views. Scene points that were located closer to the camera will have undergone a larger translation than points that were further away. If we extrapolate this observation for two cameras to a setup with multiple equidistant parallel cameras\(^5\), the pixel disparity for a scene point between each pair of neighboring cameras will be the same. In case we concatenate the images captured by all consecutive cameras into a single image cube, we are effectively creating an EPI volume. Each scene point corresponds to a single straight line in the EPI volume, called an *EPI trail*. Furthermore, each of these lines lies within a plane with a constant height value $y$, directly observable in the corresponding epipolar plane image. Like the pixel disparity in classical stereo, the *slope* of the line determines the *depth* of the corresponding scene point.

\(^5\)This is of course equivalent to observing a static scene with a moving camera, translating horizontally at a constant speed, capturing the scene at a fixed number of frames per second.
7.2.3 Matching Criteria

In classical stereo, pixelwise matching is performed by assigning a descriptor to each point, and finding pairwise correspondences across each image pair. In epipolar image plane analysis, the use of an advanced descriptor is not common practice. Instead, EPI analysis usually simply relies on the color consistency assumption, which is often used by real-time plane-sweeping algorithms [Yang 03], or the detection of edges that are propagated throughout the EPI volume. Even though it would certainly be possible to use a more complicated dense descriptor for each point in the image, these algorithms fall back on the large amount of data redundancy and inherent structure present in the EPI volume in order to establish good matches.

7.2.4 EPI Structure and Occlusions

If $x_i^0$ is the (horizontal) coordinate of imaged scene point $X_i$ at time step $t = 0$, then its corresponding position at time step $t$ will be dependent on the slope $s_i'$, the disparity per unit of time:

$$x_i^d(t) = x_i^0 + t \cdot s_i' \quad (7.1)$$

Assuming scene point $X^1$ is closer to the camera than scene point $X^2$, there will be a certain point in time $t_{occ}$ where $X^1$ occludes $X^2$:

$$t_{occ} = \frac{x_0^1 - x_0^1}{s_1^d - s_2^d} \quad (7.2)$$

---

6Recall that not all descriptors are fit for arbitrary image points, e.g. applying SIFT [Lowe 99] only makes sense for interest points that can be detected at various scales and orientations.

7This assumption states that, for different views, the same scene point will have roughly similar intensity values. While this is approximately true for highly Lambertian objects, it obviously fails for specular objects or those with more complex reflectance functions.
7.2 Epipolar Plane Image Analysis: A Dense Alternative

The key observation here is that for each pair of EPI trails, the one with the highest slope will occlude the other. This inherent structure can be utilized if we recover and remove each trail from the EPI volume, starting with the points closest to the camera. Exploiting this structure is what separates EPI analysis and its derivatives from other stereo and structure from motion algorithms.

7.2.5 Dense Reconstruction

In their seminal work, Bolles et al. [Bolles 87] showed how the volume could be analyzed by finding paths and surfaces in the spatio-temporal volume. However, no fully dense 3D reconstruction was ever demonstrated. Almost two decades after Bolles had laid the foundations for sparse reconstruction from EPI volumes, Criminisi et al. [Criminisi 05] expanded their use to dense reconstruction, including the analysis of surfaces with (limited amounts of) specular reflections. In order to achieve this, they introduced two additional concepts: the EPI-strip and the EPI-tube.

EPI-strip

An EPI-strip is defined as a quadrilateral on the epipolar plane image, with two sides aligned with the bottom and top edges of the epipolar plane image, corresponding to the first and last frame of the sequence, respectively. This is illustrated in Figure 7.2.

EPI-tube

A collection of EPI-strips constitutes an EPI-tube, i.e. a volumetric primitive with a ruled surface boundary that represents a coherently moving set of pixels, free of occlusions occurring within the volume. This is illustrated in Figure 7.3.
Segmentation and Reconstruction

Criminisi et al. [Criminisi 05] proposed two algorithms for the recovery of EPI strips from a given image sequence. Their first algorithm analyzes the problem in disparity space [Dev 74, Yang 93], while the second one directly analyzes the color data contained in the EPI volume.

In their initial approach, the authors tried extracting clearly defined EPI-trails, with the goal of merging adjacent trails into EPI-tubes. In order to understand the proposed method, it should be noted that each EPI-trail corresponds to a single disparity hypothesis \((x_0, y_0, d)\), where \(x_0\) and \(d\) define an EPI-trail coordinate according to eq. (7.1), and \(y_0\) determines its \(y\) coordinate. The set of all such possible hypotheses forms a disparity space, an old concept dating back to the early cooperative stereo correspondence algorithms [Dev 74]. Sampling this function on a regular \((x_0, y_0, d)\) grid provides us with a so called disparity space image (DSI) [Yang 93]. Within this space, a consistent set of feature correspondences is represented by a set of surfaces hidden within the volume. Because of the interesting duality properties\(^8\) between the EPI volume and the disparity space volume, illustrated in Figure 7.4, detecting strong EPI-trails and merging them into internally consistent EPI-tubes becomes possible within the dual disparity space.

The DSI is built by shearing the EPI volume at a large number of possible disparities and computing the intensity variances along the vertical direction. By computing the weighted\(^9\) temporal mean and variance\(^10\) for each \((x_0, y_0, d)\) in the DSI, it becomes possible to identify strong candidate EPI trails, connect them using morphological operations on the discrete DSI data, and iteratively extracting non-overlapping layers in a front to back order (see section 7.2.4). By applying this method, the data is gradually analyzed over multiple passes, recovering only the most certain data at each iteration. By applying this strategy, the algorithm exploits the large data redundancy and inherent structure within the EPI volume, instead of relying on a single threshold for photo-consistency.

\(^8\)In DSI space, EPI-trails are equivalent to points, while a single point in the EPI volume has a one-dimensional subspace of possible hypotheses associated with it in a corresponding DSI. A generalization of this concept to the full space of 3D rays is represented by the 4D Lightfield [Gortler 96, Levoy 96]

\(^9\)The weights in these functions come in the form of a binary visibility mask, which encodes occlusion constraints and tries to enforce a good representation of the EPI trail.

\(^10\)The temporal variance functions here as a color consistency value. See section 7.2.3 for more information on the color consistency assumption.
7.2 Epipolar Plane Image Analysis: A Dense Alternative

Figure 7.4: **The EPI/DSI duality**: lines in one space map to points in the other and vice-versa. *(a)* The red line in the EPI maps to the red point in the DSI and the blue line in the EPI maps to the blue line in the DSI. Furthermore, the EPI-strip (gray quadrilateral) in the EPI maps to a line segment in the DSI (shown in gray). *(b)* Dually, the green point in the EPI maps to the green line in the DSI, and the pencil of lines through a voxel in the EPI maps to a line in the DSI.

A second approach presented by the authors is to directly analyze the color information contained in the EPI volume. This volume consists of an ordered set of epipolar plane images, of which each is a collection of EPI-strips. All EPI-strips can be parameterized by two lines or four points\(^\text{11}\). Finding these strips can then be viewed as detecting and connecting each pair of EPI-trails that defines each strip. This is the essentially what the authors propose in their second algorithm. After detecting strong lines in each EPI image using an edge detector, the lines are sorted from the most slanted to the most vertical. Assuming there are \(N\) lines for a given epipolar image plane, an \(N \times N\) upper triangular matrix is created, whose rows and columns are both indexed by the lines. Each element of this matrix correspond to a potential EPI-strip. This matrix will contain the cost of assigning the area between two given EPI trails to a single EPI strip, based on the photo-consistency within the strip. By iteratively selecting and removing minimal cost EPI strips from the (visibility mask of the) epipolar image plane, and updating the cost matrix accordingly at each step, their algorithm extracts a set of EPI strips segmenting the epipolar plane image.

**Specularity Analysis**

After the observed epipolar plane image is segmented into a collection of EPI strips, it becomes possible to analyze the reflection properties of each strip, in order to detect and analyze any specular highlights present in the scene. Considering the fact that an explanation of the full classification and separation algorithm presented by Criminisi *et al.* [Criminisi 05] would be to large a deviation from the central topic of this work, we will limit ourselves to an illustrative example in Figure 7.5.

\(^{11}\)The intersections of the two lines with the top and bottom edges of the epipolar plane
Figure 7.5: Estimating diffuse and specular components for each EPI-strip. 
(a) Four frames from an input sequence, including some specular objects. (b) One of the input frames, with a superimposed scanline, and the corresponding epipolar plane image. (c) An EPI-strip selected on the epipolar plane image in the previous image. Notice the typical highlight pattern seen on convex specular surfaces. Chromatically, the highlight region seems to occlude the underlying texture of the surface. However, the orientation of the highlight is more vertical implying a farther depth. This confirms the bright pattern to be caused by a specularity. (d) Rectification of the marked EPI-strip. The diffuse component is now made vertical, while the specular component is oriented beyond 90 degrees. (e) Using photometric analysis along with geometric reasoning, the highlight is extracted and the diffuse component of the selected EPI-strip fully recovered [Criminisi 05].
7.3 From EPI Analysis to Mobile Structured Light

In the previous section, we have illustrated how epipolar plane image analysis exploits the massive amounts of data, and the inherent structure present in the image volume. In this section, we will present another way of looking at mobile structured light (MSL), by discussing the similarities between EPI analysis and our own work. These observations will present the foundation for future directions MSL could take.

The Dual EPI Image

In order to provide a smooth transition from epipolar plane image analysis to mobile structured light, we will investigate the dual EPI image, illustrated in Figure 7.6.

Assuming a regular setup for EPI analysis, a single horizontal line of the sensor of the translating camera will produce EPI images such as the one illustrated in Figure 7.6a. For the purpose of this discussion, we have marked two distinctive EPI trails, marked in the respective dominant colors of their corresponding scene point, green and orange. By observing this EPI image, it is clear that the green scene point is located closest to the camera’s image plane. Furthermore, the EPI trails intersect, indicating that for a certain period of time, the green point will occlude its orange counterpart. If we were to create a dual configuration of the EPI analysis setup, replacing cameras with projectors and vica versa, this would be equivalent to replacing the translating camera by a translating projector. If we were to illuminate the scene using a repetitive sinusoidal pattern, as illustrated by the overlayed pattern in Figure 7.6b, we are no longer observing the green and orange scene points at changing spatial coordinates through time, but rather labeling them with temporally varying illumination. As is apparent from this image, the green EPI trail crosses more wave periods than its green counterpart. Furthermore, because the EPI trail is linear, the relationship between the frequency of the observed intensities and its slant (and thus the associated depth) is linear as well.

The data redundancy that was present in EPI analysis is the same data redundancy that determines the accuracy of the observed frequency in mobile structured light. However, it is interesting to note that, while exploiting the inherent structure of the image volume was the key to solving the occlusion problems in EPI analysis, mobile structured light no longer needs this ordering constraint in order to reach a per-scene-point depth estimation. This is due to the fact that occluding points no longer influence each others cost function. In addition, because the color consistency assumption is no longer a necessary requirement for depth estimation, MSL allows for the acquisition of a much larger variety of materials, whereas EPI analysis was restricted to mainly Lambertian surfaces.
7.4 Beyond Linear Camera (and Projector) Paths

7.4.1 Circular Camera Trajectories

ICT Analysis: Generalized EPI Analysis

The ideas presented in the section 7.2 have been expanded upon over the years, using different camera paths. In 2003, Feldmann et al. [Feldmann 03a, Feldmann 03b] proposed a novel alternative to conventional EPI analysis, in which the image cube could be analyzed under general parametrized camera movements. Similar to EPI analysis, the proposed algorithm analyzes the path structure of the projected 3D points in the image cube, referred to as image cube trajectories (ICTs). In the remainder of this section, we will discuss the differences between EPI analysis and the proposed extensions by Feldmann, and how their generalization of camera movements could be related to our own work, mobile structured light.

Detection Stage

Unlike their conventional EPI counterpart, ICT analysis contains an alternate approach to the initial detection stage of the algorithm. Whereas in earlier work this step was most commonly performed by employing some kind of segmentation algorithm, followed up by an edge detection stage, the proposed algorithm takes an inverse approach by first computing a reference ICT for an assumed 3D point, and then verifying if this an assumption is valid by checking if such a path exists. If no such path exists, the depth is adjusted until the resulting ICT fits to the image cube.

Figure 7.6: The dual EPI image. (left) The epipolar plane image of Figure 7.2, with two intersecting EPI trails marked in green and orange. (right) The dual EPI of the same scene. As is apparent from this image, the periodic wave pattern emitted by the translating projector in the spatial dimension $x$ remains static throughout the entire temporal dimension $t$. As the scene is identical to the one shown in the left EPI image, the green and orange scene point have undergone the same trajectories. It can be seen that the speed at which each point traverses the emitted wave pattern is linearly related to its depth.
7.4 Beyond Linear Camera (and Projector) Paths

Figure 7.7: **ICT analysis of an orthographic camera, following a circular camera trajectory.** (left) Projection of points $a$, $b$ and $c$ onto the image plane. (right) The image cube trajectories of the points in the image volume. Depending on the quadrant (I-IV), occlusion handling rules can impose an order in which the points are to be estimated.

**Occlusion Handling**

One of the most interesting features of conventional EPI analysis is that the inherent structure within the image volume is clearly defined. Because of the known relation between the EPI trails of points at different depths (see section 7.2.4), occlusion handling becomes easy and efficient. In the case of generalized parametrized camera movements however, the question arises if such a convenient data structure can always be guaranteed. Fortunately, it seems that for circular camera movements ICTs also follow well defined paths within the image volume. Moreover, like in the linear camera movement case, it is possible to define an occlusion compatible ordering in order to efficiently handle occlusions.

**Case 1: An Orthographic Camera**

Our initial topic of discussion will be the ICT analysis of an orthographic camera, following a circular camera trajectory. Figure 7.7 shows the image plane of an orthographic camera, and three different scene point $a$, $b$ and $c$ rotating around a midpoint $m$ at the same rotational speed. The image illustrates the projection of these points into the image plane (*i.e.* their image cube trajectories) for a single scene rotation. Similar to EPI analysis, we will derive the depth of the scene points by analyzing all horizontal slices with a fixed $y$ coordinate.
If we take a closer look at Figure 7.7 we can see that the ICT of each scene point is defined by a sinusoidal function over $t$:

$$x(t) = r \cdot \sin(\phi(t) + \Delta\phi)$$

(7.3)

where $\phi(t) = \omega \cdot t$ represents the phase in the range $-\pi < \phi(t) \leq \pi$. This equation only has only two degrees of freedom: the radius $r$ and the initial phase shift $\Delta\phi$. The radius corresponds to the amplitude of the sine wave, whereas $\Delta\phi$ determines the phase of the curve. Based on these two parameters, it is possible to reconstruct the 3D position of any point from its image cube trajectory.

Now that we have described the parametrization of the image cube trajectories, the next task consists of handling occlusions. Due to the parameterized circular movement of the points, it is possible to derive rules for an occlusion compatible ordering of the ICTs. In order to discuss this matter, it is necessary to split each of the ICTs into four subsections, corresponding to the four quadrants of the circle.

- **Rule 1**: All points in quadrants I and IV will occlude those in the quadrants II and III if their ICTs intersect.
- **Rule 2**: In quadrants I and IV, all points with a larger radius will occlude those with a smaller radius, if they have the same phase and their projection rays are equal. On the other hand, in quadrants II and III points with a small radius occlude those with a larger one.

Due to both rules it is necessary to split the occlusion aware ICT detection phase into two steps. In the first step, only quadrants I and IV are considered. The algorithm starts with the maximum expected radius, iterating through all possible ICTs with that given radius$^{13}$. Like in the case of linear EPI analysis, points at equal depths/radii do not intersect/occlude each other, and can be identified during a single iteration, masked and excluded from further processing. Once the search for a given radius is completed, the algorithm moves on to the next smaller radius, and repeats the process.

**Case 2: A Perspective Camera**

We can perform a similar analysis in case of a perspective camera, with the addition of a few modifications. By applying a perspective projection, the ideal sinusoidal curves that we observed in the orthographic case are now slightly distorted. Furthermore, the trajectory of a particular 3D object point is no longer

$^{12}$Thus the radius $r$ of the circle trajectory corresponds to the amplitude of the signal $x(t)$.

$^{13}$This is achieved by varying one degree of freedom: the phase shift value $\Delta\phi$
7.4 Beyond Linear Camera (and Projector) Paths

Figure 7.8: Mapping a scene point onto a perspective camera, following a circular camera trajectory. (left) The process of mapping a scene point in the X-direction; (right) The process of mapping a scene point in the Y-direction.

restricted to a single horizontal plane in the image cube, but forms a curve within the three-dimensional image cube. Fortunately, the shape of this curve is still determined by the same two degrees of freedom that were as present in the orthographic case: the radius \( r \) and the initial phase shift \( \Delta \phi \). The basic algorithms for curve detection and occlusion handling from the previous section also remains the same, but the valid visible range of a point in a curve is slightly reduced.

We will now analyze the image cube trajectory\(^{14}\) of a single observed 3D point \( \mathbf{X} \):

\[
\begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix} = \begin{bmatrix}
r \cdot \sin(\phi) \\
Y \\
Z_0 + r \cdot \cos(\phi)
\end{bmatrix}
\]  
(7.4)

where \( \mathbf{X} \) performs a circular trajectory with radius \( r \) around the point \((0,0,Z_0)^T\). By applying a perspective distortion, assuming a focal length \( f \), the scene point is mapped onto the 2D image coordinate \( \mathbf{x} = (x,y,z)^T \):

\[
\begin{bmatrix}
x \\
y \\
z
\end{bmatrix} = \begin{bmatrix}
-\frac{L}{Z_0} \sin(\phi) \\
\frac{r}{Z_0} \\
1 + \frac{r}{Z_0} \cdot \cos(\phi)
\end{bmatrix}
\]  
(7.5)

\(^{14}\)With the intent of retaining an uncluttered set of equations, we do not mention the time parameter \( t \) in these equations. The reader should however keep in mind that all variables aside from the radius \( r \) and the initial phase shift \( \Delta \phi \) (such as the phase \( \phi(t) \rightarrow \phi \)) are actually defined in function of \( t \).
Figure 7.9: **Image cube trajectories using a perspective camera, which follows a circular trajectory.** *(left)* The ICT of the imaged point in the X-direction; *(right)* The ICT of the imaged point in the Y-direction. It is clear that the presence of the focal point distorts the trajectory of the imaged points.

It is clear that the presence of the focal point and the depth dependence has an influence on the produced curve. We illustrate this with several exemplary trajectories in Figure 7.9. These curves no longer represent a perfect sinus in the horizontal direction like they did in the orthographic case, but have become slightly distorted. Object points with a larger radius $r$ deviate more from the sinusoidal curve. The phase of the maximum amplitude $\phi_{\text{max}}$ is now determined by:

$$\cos(\phi_{\text{max}}) = -\frac{r}{Z_0}$$  \hspace{1cm} (7.6)

The image coordinate in the vertical direction is also influenced by the perspective projection: points that are closer to the camera are shifted towards the horizontal image borders. All points, except those which are located in the horizontal plane through the optical axis of the camera, have time-varying $y$ coordinates. Since the magnitude of these variations is also depth and radius dependent, there does not exist a common 2D surface through the video cube that contains multiple object points for the entire object rotation. Therefore, the ICT analysis for perspective projection has to be performed on the entire 3D image cube.

The general algorithm of the ICT analysis and the occlusion handling remains the same as for an orthographic camera. However, the separation between quadrants I and II, as well as between quadrants III and IV, is now determined by the phase of the maximum amplitude $\phi_{\text{max}}$, as specified in eq.(7.6).
7.4 Beyond Linear Camera (and Projector) Paths

7.4.2 Circular Projector Trajectories

The central question that now arises is how to transfer the observations made in ICT analysis to the domain of mobile structured light. In this section, we will discuss the potential of such a transition. It should be noted however that the remainder of this chapter merely consists of a preliminary theoretical analysis, and has not been converted into a robust acquisition system. It does however illustrate a clear possible direction for future work.

Observations

Whereas the original EPI analysis algorithm searched for structure in the image volume by segmenting the inputs, ICT analysis examines precomputed paths in the image volume, assigning a cost value to each path and retaining/removing minimal cost paths from the image cube. Conceptually, this is equivalent to a transition from a feature-based method to a plane-sweeping (or in this case: path-sweeping) approach.

Assuming we have a setup with a static camera observing the scene, illuminated by a projector with a known trajectory, it might be possible to apply a similar approach to mobile structured light. By assigning a depth value to a camera pixel, the position of the associated scene point is uniquely defined. Assuming we have a perfectly known projector path, and a known illumination pattern, it becomes possible to predict the reflected illumination pattern of the observed pixel.

Theoretical Analysis

We shall discuss the circular trajectory proposed in the previous section, in which the path has two degrees of freedom: the radius $r$ and the initial phase shift $\Delta \phi$. The radius determines the depth of the point, whereas the phase shift determines the starting position of the projector. Assuming we are employing an identical sine wave pattern like we did in the linear case, there is no clear one-to-one mapping between camera and projector pixels. Like in the linear case, MSL thus requires that we recover the depth directly from the signal. In order to keep the structure of the observed signal as simple as possible, we recommend the use of an orthonormal projector. This way, the base frequency will remain constant for all depth values, while any frequency alterations will be solely the product of the path followed by the observed point. In order to get an intuitive idea of the distortion introduced by this non-linear projector movement, we refer to Figure 7.10.

\footnote{Lines are also usable features in scene reconstruction. Thus, the edge detectors used in EPI analysis perform a comparable function to feature detectors in classical stereo.}
Figure 7.10: **Influence of the projector trajectory on the incoming illumination - circular, orthonormal projector case.** Note that the incoming illumination follows a well defined pattern, depending on the radius \( r \) (which defines the number of waves that are projected onto the point) and the initial phase shift \( \Delta \phi \) (which defines the starting point at which an (unoccluded) point starts to become illuminated by the pattern. As the point becomes illuminated by pixels closer to the horizontal center of the pattern, the frequency increases.
7.4 Beyond Linear Camera (and Projector) Paths

Upon closer inspection, it is possible to relate a point’s horizontal (Figure 7.10a) and vertical trajectories to the observed intensities by the camera (Figure 7.10b/c). To be more precise, Figure 7.10a describes a scene point becoming visible to the projector at time \( t_{\text{min}} = 0 \) at position \((-r,0)\), following (half) a circular path for \( n = 1000 \) timesteps, to become occluded once again once it reaches position \((r,0)\).

Thus, the horizontal trajectory follows the following function:

\[
x(t) = \cos(\pi + \pi \cdot (t - t_{\text{min}})/n) \cdot r
\]

(7.7)

Similarly, a point’s vertical trajectory can be described by:

\[
y(t) = \sin(\pi + \pi \cdot (t - t_{\text{min}})/n) \cdot r
\]

(7.8)

In case of an orthogonal projection, we can define the projected wave pattern intensities\(^{16}\), consisting of \(w\) different waves, in terms of the horizontal position of the illuminated scene points:

\[
I_p(t) = \sin \left( \frac{2\pi \cdot w \cdot x(t)}{n} - \frac{\pi}{2} \right)
\]

(7.9)

The result is a frequency modulated signal, which increases in frequency as the angle approaches \(\frac{\pi}{2}\), equally decreasing afterwards. In order to describe this change, we need to introduce a basic concept from signal processing literature called the **instantaneous frequency** \(f(t)\):

\[
f(t) = \frac{\Phi(t)}{2\pi} \cdot \frac{d}{dt}
\]

(7.10)

where \(\Phi(t)\) is the **instantaneous phase** of the signal at time step \(t\).

In the case of a point following a circular trajectory at a constant angular speed, the velocity at which the point transverses the x-direction is linearly related to the instantaneous frequency of the observed signal at time step \(t\).

\[
f(t) = a \cdot x(t) \cdot \frac{d}{dt} + c_a = b \cdot y(t) + c_b
\]

(7.11)

where \(a,b,c_a\), and \(c_b\) are constants for a single recording. Similarly, the global\(^{17}\) phase \(\Phi(t)\) is linearly related to the horizontal trajectory over time \(x(t)\).

\[
\Phi(t) = s \cdot x(t) + c_s
\]

(7.12)

\(^{16}\)In eq.(7.9), the intensity values are located in the interval \([-1,1]\). This is of course unrealistic, and we assume that in practice a preprocessing step will be applied to scale the signal to the appropriate intensity ranges. This will also be required to undo the amplitude modulation performed by the material properties, similar to our linear MSL approach (see section 5.2: The Reflectance Function).

\(^{17}\)In case of a discrete set of subsequent (local) instantaneous phase angles \(\phi(t)\), the global phase \(\Phi(t)\) is acquired by adding \(2\pi\) for each time the local phase has passed from \(-\pi\) to \(+\pi\). This process is called **phase unwrapping**, and is also used in structured light methods which apply phase shifting approaches.
Proposed Approach

In light of the equations and observations made in the previous section, we propose two possible complementary algorithms to recover depth from circular mobile structured light. Both algorithms would require a preprocessing stage similar to the one we have suggested in our linear mobile structured light setup, in order to remove influence from material properties (see section 5.2: The Reflectance Function). The resulting function should closely approximate (possibly part of) a frequency modulated sine wave, with function values within the range $[-1, 1]$.

The main difference between the two approaches constitutes which equation to use. Both approaches require the application of a curve-fitting algorithm on their respective data, in order to recover a sine wave corresponding to a scaled and translated version of either the horizontal or vertical trajectory over time.

The first approach, based on eq. (7.12), employs local and global phase values as data to fit the horizontal trajectory over time. This approach has the disadvantage that it cannot efficiently handle gaps larger than a single wave in the observed pattern, as this would ruin the phase unwrapping process. As a potential advantage, the accumulated global phase values create a large range in which the data $\Phi(t)$ would be located, which could potentially be beneficial in terms of accuracy.

The second approach, based on eq. (7.11), employs the instantaneous frequency measurements as data to fit the vertical trajectory over time. This has the advantage that the data is completely robust against occlusions and specularities, as the instantaneous frequency is a local measurement and does not accumulate like the global phase in the first approach does. However, accurately and robustly estimating the instantaneous frequency of a signal still remains an open problem in signal processing literature. Direct differencing $f(t) = \phi(t) - \phi(t - 1)$ is the most obvious solution, but can hardly be called robust in the presence of noise. Therefore, a more robust estimator would need to be used.

Preliminary Conclusions

Based on our observations, we dare not say with 100% certainty that it is possible to generalize our linear structured light method to any arbitrary trajectory. However, based on our initial theoretical analysis, we do believe this to be the case for orthogonal projectors, performing circular trajectories around the scanned object. We have proposed two approaches, based on the recovery of the horizontal and/or vertical component of the circular trajectory, both with their potential advantages and disadvantages. We believe this could be the basis for a practical circular extension of our mobile structured light approach, yet the attainable level of accuracy of the proposed methods remains to be seen.
Chapter 8

Conclusions

Contents

8.1 Summary ......................................................... 116
  8.1.1 Depth by Frequency Estimation ......................... 116
  8.1.2 Removing the Camera Constraint ....................... 117
  8.1.3 The EPI-MSL Duality ..................................... 117
  8.1.4 Comparison to State of the Art .......................... 118
8.2 Future Work ..................................................... 118
8.1 Summary

In this dissertation we have presented a novel method for 3D structure acquisition, which we have called mobile structured light. Unlike the more classical structured light methods, in which a static projector illuminates a (static) scene with a variety of time-varying illumination patterns, our proposed technique makes use of a mobile projector emitting a single static sinusoidal illumination pattern. This projector, which we refer to as a sliding projector, is translated at a constant velocity in the direction of the projector’s horizontal axis.

8.1.1 Depth by Frequency Estimation

Illuminating the object in this manner allows us to perform a fast per pixel analysis, in which we can decompose the illumination sequence observed by the camera into a corresponding set of frequency components. We have observed that there exist a linear relationship between the depth of a scene point, and the frequency of the observed illumination pattern in the corresponding pixel\(^1\). Thus, after establishing the dominant frequency in the Fourier power spectrum, this frequency can be directly converted into a corresponding depth value.

Frequency Refinement, using Signal Processing Techniques

By establishing this linear relationship between point depth and signal frequency, we have effectively casted depth estimation as a signal processing problem. Because of this feature, it is possible to apply known techniques from this domain in our method, \textit{i.e.} the single tone estimation problem, which deals with recovering a single tone (or frequency) from a discrete 1D audio signal, corrupted by noise. In our case, the noise corresponds to light reflected by the scene point that was not part of the point’s diffuse reflection component. After reviewing the work done on this topic by the signal processing community, we have opted for the very efficient and easy to implement technique of Macleod [Macleod 98], which is based on the interpolation of Fourier coefficients. This technique approaches the Cramér-Rao bound, a theoretical limit and benchmark for the performance of algorithms of this type, and has proven to be very well-suited for our purposes, significantly improving results compared to our initial fundamental frequency bin-searching algorithm.

\(^1\)We have assumed the intrinsic calibration of both the camera and projector to be known, otherwise it would be impossible to directly link the geometry of the scene to the observed illumination.
8.1 Summary

8.1.2 Removing the Camera Constraint

The shape acquisition method as initially proposed in this work has one important restriction. The depth estimates, as we had computed them until this point in the dissertation, were effectively *distances to the projector* instead of distances to the camera, such as is the case for most if not all depth estimation algorithms. Fortunately, it is possible to convert these projector depths into corresponding camera depth values, if one of the following three conditions is true:

(a) the (perspective) camera’s principal plane is perfectly aligned with the principal plane of the sliding projector; 
(b) the principal plane of the orthonormal camera is located on a plane parallel to the principal plane of the sliding projector; 
(c) the extrinsic calibration of the camera can be computed afterwards. In the first case, the projector depth are equal to the camera depths. In the second case, the camera depths are equal to the projector depths, up to an unknown but arbitrary constant.

It is the third case that is the most general. If the extrinsic calibration of the camera is known, it becomes possible to directly convert the projector depth value into a corresponding camera depth value. In order to obtain the extrinsic calibration data, we have made use of an additional illumination pattern in the form of a 1D De Bruijn sequence, tailored onto the existing 2D sine wave pattern. By time multiplexing both signals, and synchronizing our camera and projector emission and recording speeds, we are able to uniquely encode a sparse subset of projector pixels, while at the same time retaining the ability to compute projector depths values from the observed sine wave illumination. Based on this information, retrieved on a per (camera) pixel basis, we are able to estimate a sparse reconstruction of the scene. As this sparse set of 2D-3D camera-scene correspondences is sufficient to recover the camera’s location and orientation within the scene, we are able to convert the dense set of point-projector distances into a dense set of camera depths, effectively providing us with a dense reconstruction of the observed scene.

8.1.3 The EPI-MSL Duality

Finally, we have discussed the dual relationship between epipolar plane image (EPI) analysis, a passive shape acquisition technique which uses a linearly translating camera in order to capture the scene structure, and the proposed mobile structured light (MSL) method.

EPI analysis exploits the large amount of inherent structure in the recorded image volume to recover scene structure, relying on data redundancy and a color consensus matching criterium to establish cross-frame correspondences. It is this same data redundancy that determines the accuracy of the observed frequency in our proposed
dual technique. However, it is interesting to note that, while exploiting the inherent structure of the image volume was the key to solving the occlusion problems in EPI analysis, mobile structured light no longer needs the associated ordering constraints in order to reach a per-scene-point depth estimation. This is due to the fact that occluding points no longer influence each others cost function. In addition, because the color consistency assumption is no longer a necessary requirement for depth estimation, MSL allows for the acquisition of a much larger variety of materials, whereas EPI analysis was restricted to mainly Lambertertian surfaces.

8.1.4 Comparison to State of the Art

Because the proposed method performs depth estimation on a per pixel basis, it is able to preserve sharp edges in the produced depth image. Furthermore, unlike classical structured light methods, the quality of our results is not limited by projector or camera resolution, but is solely dependent on the temporal sampling density of the captured image sequence. Additional benefits include a significant robustness against common problems encountered with structured light methods, such as occlusions, specular reflections, subsurface scattering, interreflections, and to a certain extent projector defocus.

8.2 Future Work

Viewpoint encoded structured light methods, such as the mobile structured light (MSL) method proposed in this dissertation, have long remained unexplored in traditional 3D acquisition literature. As such, we believe there is still a lot of room for new research in this area, including on the topic presented in this work. Such potential research directions might include, but are not limited to:

- **EPI-MSL duality:** One clear further area of investigation is the duality between EPI analysis and MSL, with the potential of acquiring a full 3D model in a single scan, instead of having to align multiple point clouds resulting from combining multiple scans of the same object.

- **BRDF acquisition:** An interesting topic would be to research possible adaptations of our technique or setup to allow for a more extended acquisition of material properties.

- **Alternate hardware:** With the availability of high-dynamic range spatial gradient cameras, and ultra-high speed temporal gradient cameras, it would be interesting to determine the required modifications to our algorithms for the purpose of directly recovering depth maps from the output of such devices.
8.2 Future Work

- **Coded strobing:** In our related work, we have explored the recently developed technique of coded strobing photography. For the acquisition of sparse signals, this new technique replaces the hardware limit imposed by the camera recording speed by the strobing speed, significantly increasing the reconstructed signal’s sample size. As our intensity signal is a sparse signal, this technique shows much potential for being integrated into our method.

- **Sub-pixel edge detection / alpha matting:** Comparing the power spectra of border pixels to the power spectra of neighboring pixels, in particular near the dominant frequencies of the neighboring pixels, might help us locate the object borders at sub-pixel accuracy, or perform some sort of alpha matting.

- **Exploring different patterns:** We might be able to extend our use of time multiplexed patterns to color multiplexed patterns, but this might come at a cost in terms of accuracy for difficult materials.
Appendices
Appendix A

Scientific Contributions and Publications

The following list of publications, presented at international scientific conferences, contains work that is part of this dissertation:


The following list of publications contains work that was done during the duration of this PhD., but was not included in this dissertation:


Samenvatting (Dutch Summary)

Gedurende de laatste decennia hebben computers een steeds meer prominente plaats ingenomen binnen alle aspecten van ons leven. Of het nu gaat om de kleine zakcomputers die zich in je draagbare telefoon bevinden, of gespecialiseerde systemen die een industrieel proces controleren: de computer is bij de uitvoering van de meeste taken niet meer weg te denken. In deze dissertatie gaan we dieper in op één specifieke probleemstelling waarvoor computers een belangrijke component van vooruitgang hebben betekend, namelijk digitale 3D reconstructie of vormacquisitie.

De kunde van het bepalen en reconstrueren van de vorm van een origineel, om dit later te kunnen reproduceren met of zonder de nodige aanpassingen, heeft haar nut al bewezen gedurende vele eeuwen. Of het nu gaat om het creëren van gietvormen om bronzen werktuigen of wapens in massa’s te produceren, of om het opmeten van de structuur van kleefguurtjes om een animatiefilm te produceren: de kern van de taak is in meer dan tweeduizend jaar niet veranderd: het opmeten van de structuur van een origineel, opdat deze kan gereproduceerd of aangepast worden. Met de komst van de computer is dit proces niet alleen veel efficiënter en preciezer geworden, maar breidt de waaier van mogelijkheden tevens elke dag verder uit.

In dit werk richten we ons op één specifieke tak binnen het gamma aan vormacquisitiemethodes: de niet-destructieve, optische methodes gebaseerd op gestructureerd licht. Hierbij wordt er traditioneel gebruik gemaakt van een of meerdere camera’s en projectoren om een object of scène te belichten, zodat uit de geprojecteerde patronen de 3D-structuur van de door de camera’s geobserveerde scène kan gereconstrueerd worden. In tegenstelling tot de klassieke methodes van
dit domein, waarbij de scène vanuit vaste hoeken bekeken en beschenen wordt met lichtpatronen die evolueren in de tijd, hebben we in deze dissertatie voor een nieuwe aanpak gekozen. In de door ons voorgestelde methode, die *mobiel gestructureerd licht* gedoopt werd, wordt een statische scène met een vast sinusvdaal patroon beschenen, maar vanuit een vooraf gekend lineair pad. Door de projector aan een constante snelheid horizontaal beweegt met referentie tot zijn beeldvlak, kan er een lineair verband gelegd worden tussen de afstand van de projector tot elk 3D-punt in de scène, en de frequentie van het geobserveerde lichtpatroon in de overeenkomstige 2D-pixels. Aangezien we uiteraard geënteerstesseerd zijn in de afstand van de punten tot de camera in plaats van die tot de projector, is er vooraf een calibratiestap nodig, ofwel moet er aan online calibratie gedaan worden met een secundair gtn. De Bruijn lichtpatroon, geënt op het voorgaande golfpatroon.

De objecten op deze manier belichten laat ons toe om aan een hoge snelheid alle pixel berekeningen in parallel uit te voeren, waarbij het opgemeten lichtsignaal kan geanalyseerd worden aan de hand van bestaande vakliteratuur uit het gebied van de signaalverwerking. Op basis van Fourieranalyse kunnen we het opgemeten signaal voorstellen als een reeks frequentiecomponenten, waarmee we rechtstreeks de pixeldiepte kunnen afleiden. Om de accuraatheid van de methode te verbeteren, bespreken we tevens de link tussen het voorgestelde werk en signaalverwerkingsliteratuur binnen de domeinen van frequentie-interpolatie, en de relatief recente ontwikkelingen op het gebied van compressed sensing.

De voorgestelde techniek heeft verschillende voordelen ten opzichte van de klassieke gestructureerd licht methodes. Aangezien onze methode de diepteschatting uitvoert op een per pixel basis, is ze in staat om scherpe randen te bewaren. Verder, in tegenstelling tot de klassieke tegenhangers, wordt de kwaliteit van ons resultaat niet beperkt door de projector- of cameraresolutie, maar is de doorslaggevende factor echter de temporele resolutie van de bemonstering van het inkomende lichtsignaal. Hierbij moet gezegd worden dat bij een gelijkmatige bemonstering, de projectorresolutie wel degelijk een invloed heeft op de kwaliteit van het eindresultaat. Echter, waar bij de klassieke methodes deze spatiale resolutie een harde limiet stelt, wordt deze bij de voorgestelde methode slechts een schuivende parameter, gedomineerd door de temporele resolutie van de bemonstering. Tot slot willen we nog vermelden dat de voorgestelde methode zeer robuust is in de aanwezigheid van fenomenen die traditioneel een probleem vormen voor de klassieke methodes, zoals occlusies, speculaire reflecties, subsurface scattering, interreflecties, en tot op een bepaalde hoogte projector defocus. Deze laatste kan zelfs de kwaliteit van het signaal in sommige gevallen verbeteren.
Tot slot bespreken we in deze dissertatie nog de duale relatie tussen mobiel gestructureerd licht en *epipolar plane image analysis*, een passieve optische vormacquisitietecniek die geen gebruik maakt van een projector, maar enkel van een horizontaal transelerende projector. Deze oude techniek maakt gebruik van de grote hoeveelheid aan inherente structuur in het opgenomen beeldmateriaal en de overvloed aan redundante data om tot een consistente 3D-scènestructuur te komen. Het is dezelfde aanwezigheid van redundante data die in de door ons voorgestelde actieve tegenhanger voor robuustheid zorgt. Echter, onze techniek minder assumpties - zoals kleurconsistentie of de zgn. ordering constraint - en kan hierdoor overweg met een veel grotere variëteit aan materialen, waar EPI-analyse voornamelijk beperkt blijft tot het opmeten van strikt diffuse objecten, met een occasionele korte speculaire highlight.


International Conference on Computer Vision, Kyoto, Japan, September 2009.


