Combined 3D, multispectral, and fluorescence imaging through design of an integrated structural light scanner

Kristian Ryder Thomsen

Kongens Lyngby 2016

Technical University of Denmark
Department of Applied Mathematics and Computer Science
Richard Petersens Plads, building 324,
2800 Kongens Lyngby, Denmark
Phone +45 4525 3031
compute@compute.dtu.dk
www.compute.dtu.dk

Summary (English)

The goal of the thesis is to design a system for measuring 3D simultaneously with spectral image recording in the VideometerLab4 instrument, develop analysis algorithms to exploit the combined 3D and spectral information, and to demonstrate that this can be utilized efficiently in 1-3 applications.

The possible approaches for designing such an integrated 3D measurement system are discussed and evaluated and an in-depth analysis made of selected types of structured light solutions and time-of-flight technology. A variant of phase shifting profilometry based on Fourier analysis is selected as the most suitable method given the system specifications. The problem of phase unwrapping is also studied and a dual-wavelength solution selected. Algorithms for triangulation of points in 3D space are discussed and a computationally effective algorithm is derived. The extended acquisition time for additional 3D measurements are just ≈ 0.6 seconds. The accuracy of the systems 3D reconstructions are analysed and the height error found to be normally distributed around zero with a standard deviation of just 34.7 micrometers. The lateral uncertainty is 25.9 micrometers. An accurate and robust stereo calibration is explained and performed with sub-pixel accuracy for both the camera and the projector. Lastly two specific applications of combined 3D and spectral data are introduced and evaluated. First a novel algorithm is presented for classification of grains orientation into the categories of either dorsal or ventral and it is shown to be statistically significantly outperforming the 2D alternative. Segmentation of granular products, such as rice, grains or seeds, are also studied and a modification to the existing 2D approach presented that are expected to increase the number of correctly segmented grains by $\approx 1.5\%$.

Summary (Danish)

Målet for denne afhandling er at designe et system til 3D måling samtidig med spektral billedoptagelse i VideometerLab4 instrumentet, udvikle analysealgoritmer til at udnytte den kombinerede 3D og spektrale information samt at vise, at dette kan udnyttes effektivt i 1-3 applikationer.

De mulige metoder til at designe et sådant integreret system til 3D måling diskuteres og evalueres og der udføres en grundig analyse af udvalgte metoder inden for struktureret lys og time-of-flight teknologi. En variant af phase shifting profilometry baseret på Fourier-analyse er udvalgt som den mest egnede metode i betragtning af systemets specifikationer. Metoder til at udføre phase unwrapping er også studeret og en dobbelt bølgelængde løsning er valgt. Algoritmer til triangulering af punkter i 3D gennemgås og en beregningsmæssig effektiv algoritme udledes. Det tager kun $\approx$ 0,6 sekund at udføre de supplerende 3D opmålinger. Nøjagtigheden af systemets 3D rekonstruktioner er analyseret og højdefejlen er fundet normalt fordelt omkring nul med en standard afvigelse på kun 34,7 mikrometer. Den laterale usikkerhed er 25,9 mikrometer. En nøjagtig og robust stereo kalibrering er gennemgået og udført med sub-pixel præcision for både kameraet og projektoren. Til slut er to specifikke anvendelser af den kombinerede 3D og spektrale data introduceret og evalueret. Først gennemgås en ny algoritme til klassifikation af orienteringen af korn som enten værende ventrale eller dorsale. Det vises at denne nye metode klarer sig statistisk signifikant bedre end 2D alternativet. Segmentering af granulære produkter, såsom ris, korn eller frø, studeres også og en modifikation af den eksisterende 2D metode præsenteres. Det forventes at denne modificerede metode kan øge antallet af korrekt segmenterede korn med $\approx$ 1,5%.

Preface

This thesis was prepared at the Technical University of Denmark, Department of Applied Mathematics and Computer Science, in fulfilment of the requirements for acquiring a Master of Science in Engineering in Mathematical Modelling and Computing.

The work was undertaken partly at the Department of Applied Mathematics and Computer Science and partly at the industrial cooperator Videometer A/S.

An electronic version of this thesis can be found online in the IMM Publication Database at www.imm.dtu.dk/pubdb.

Kongens Lyngby, 01-February-2016

Kristian Ryder Thomsen

Kristian Ryder Thomsen

Acknowledgements

I would like to thank all employees at Videometer for a cozy and comfortable working environment. This made it possible to arrive with a smile every morning and go home in the afternoon with a sense of accomplishment.
I would especially like to thank Tobias Have for help with mounting the projector on the mock-up and modifying the instruments electronics, Karsten Hartelius for advice and guidance during the development of the application on grains, and Christian Gleerup for help with software related issues.

A special thanks goes to PhD student Jakob Wilm (DTU) for answering detailed questions, giving suggestions of literature and introducing me to the terms and concepts of phase shifting profilometry.

I would also like to thank my supervisors Jens Michael Carstensen (DTU) and Anders Bjorholm Dahl (DTU) for always being available and for help, advice and guidance in connection with the weekly guidance meetings, and for quickly answering questions by e-mail.

Abbreviations

CCD	Charged couple device
CDA	Canonical discriminant analysis
CDF	Cumulative distribution function
DLP	Digital light processing
DMD	Digital micro-mirror device
EV	Exposure value
LED	Light emitting diode
NaN	Not a number
PS	Phase shifting
PSI	Phase shifting interferometry
PSP	Phase shifting profilometry
R&D	Research and development
RBF	Radical basis functions
TOF	Time of flight
TWPU	Two wavelength phase unwrapping

Contents

Chapter 1

Introduction

This thesis begins with an introduction of the industrial cooperator Videometer in section 1.1 followed by a description of their multispectral imaging system the VideometerLab in section 1.2. The addition of an integrated 3D measurement system for simultaneous 3D and multispectral imaging is considered highly valuable for reasons discussed in section 1.3. Section 1.4 lists the specifications and requirements for such a system.

The rest of this thesis is outlined as follows. Chapter two discusses the possible approaches for designing such an integrated 3D measurement system by analysing selected types of structured light solutions and time-of-flight technology.
Chapter three starts by discussing three variants of phase shifting profilometry that may be used to reconstruct the 3D topology. Algorithms for triangulation of points in 3D space are discussed and a computationally effective algorithm is derived. The chapter ends by estimating and analysing the accuracy of the 3D reconstructions.
Chapter four discusses and explains how to perform accurate and robust stereo calibration of the camera and the projector in order to estimate their intrinsic and extrinsic parameters. The accuracy of the stereo calibration is analysed and evaluated.
Chapter five showcases two specific applications that benefit from combining 3D and multispectral imaging. First a novel algorithm is presented for classification of grains orientation into the categories of either dorsal or ventral. It is then studied how segmentation algorithms for granular products, such as rice, grains or seeds, may benefit from combining the multispectral image with 3D data.
Finally chapter six concludes on the entire thesis, summarizes the main results and gives a review of future work.

1.1 Videometer A/S

Videometer[1] is a Danish company that specializes in spectral imaging, automated visual measurements, quality control and accurate vision-based measurements of texture, colour, topography, gloss, shape, and surface chemistry. Furthermore Videometer also specializes in in-line visual quality control systems where samples are inhomogeneous or where human vision is the current reference method, robot vision as well as R&D intensive vision technology projects. All using fast, non-contact and objective assessments ensuring reproducible and robust measurements.

The high performance vision systems developed by Videometer are used in a broad range of industries. Both as laboratory analysis or as in-line measurements in a running production. Videometer solutions have among others been successful in the following fields:

Material surfaces
Quality control in the production is boosted using objective colour and surface quality assessments of e.g. fabrics, paper, fur, wood, ceramics, liquids, metal and plastic.

Biotechnology
Counting and identification of micro-organisms. Visual assessment of enzymatic treatment. Automatic identification of fungi strains/clones with desired properties.

Food industry
Food quality is often assessed visually. By using wavelengths in the ultraviolet, visual and near-infrared range you obtain information regarding chemical composition. By using near-infrared illumination it becomes possible to visually inspect bruising, rotten areas and defects in meat, fruit and vegetables before these become visible to the naked human eye. Also freshness and moisture content can be measured well using near-infrared imaging.

Pharmaceutical industry
Colour, texture and surface features of pharmaceutical products are monitored, and visual quality is controlled in liquids, powders, granulates and solid products. This offers a rapid solution to screening for out of specification tablets and contaminated pharmaceutical powders.

Vision controlled robot technology
Vision controlled robots are used for flexible handling of parts in a production line, high precision monitoring or to control e.g. filling processes.

[1] Videometer A/S, Lyngsø Allé 3, DK-2970 Hørsholm. CVR 24230430.

1.2 The VideometerLab

The VideometerLab is a camera system developed to take multispectral images under calibrated and controlled lighting conditions. The VideometerLab consists of a hollow integrating sphere, known as an Ulbricht sphere, with a white titanium coating on the inside. This ensures uniform scattering and a diffusing effect. A light ray emitted by one of the LEDs are by multiple scattering reflections distributed equally throughout the entire sphere. By lowering the sphere onto the sample all external light sources are excluded. Along the equator of the sphere are mounted a series of monochromatic LEDs. The monochromatic LEDs are flashed one by one and a monochromatic image of the sample's reflections of the LEDs specific wavelength is obtained by the camera. Thus every pixel in the captured image is a reflectance spectrum and the instrument may include up to 19 wavelengths in the ultraviolet, visual, and near-infrared spectrum. Using a robotic arm a physical high-pass filter can be placed in front of the camera. This allows for the capture of fluorescence images[2]. Usually the emitted light has a longer wavelength then the absorbed light which justifies the use of a high-pass filter. For the VideometerLab4 the with of a pixel is ≈ 36µm and the diameter of the inspection opening is 110mm. A full list of technical specifications can be found in appendix A. Figure 1.1 and 1.2 on the following page shows a picture of the VideometerLab and a schematic drawing showing the placements of the LED's, camera and sample area.

1.3 Additional 3D information

Many of Videometers customers are using the VideometerLab for analysis on seeds and grains. An important first step in almost any analysis on seeds and grains is to segment into individual seeds or grains. Segmentation in 2D images of touching objects is a well studied problem however high quality segmentation of seeds, grains or similar small touching and possibly overlapping objects still pose a challenging task. Having access to the surface geometry of a 3D (technically 2.5D) hight model of the samples together with the 2D multispectral images are expected to dramatically improve segmentation results of seeds and grains and similar objects.

Videometer has several other desirable applications of 3D measurement capability within the VideometerLab. In the current version of the VideometerLab a body of revolution is used to estimate several properties of objects. However

[2] Fluorescence is the emission of light by a substance that has absorbed light or other electromagnetic radiation.

Figure 1.1: *The VideometerLab.*

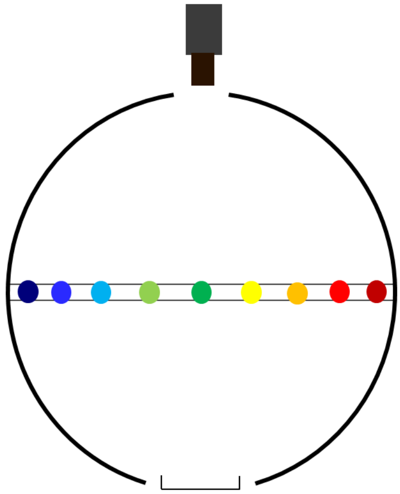


Figure 1.2: *A schematic drawing showing the placements of the LED's, camera and sample area.*

a body of revolution is an incomplete approximation of the actual 3D shape. This introduces considerable errors when estimating the flatness of an object, volume or density of objects or three sided seeds[3]. If the resolution and accuracy permits a description of the physical surface texture this is also highly desirable. Spectral imaging is an efficient and fast way to characterize and quantify many types of skin lesions and other dermatological conditions [1]. With the addition of 3D information it will be possible to quantify if and how much skin has swollen. Furthermore several confidential applications of 3D information have been requested by multiple of Videometers customers.
It is therefore considered highly valuable to have integrated 3D measurements in the VideometerLab.

[3] Such as seeds from black bindweed (fallopia convolvulus) characterized by being 3 to 4.4mm long, 3 sided with faces more or less equal, minutely roughened, dull black and polished on its angles.

1.4 Problem statement and specifications

The overall goal of this thesis is to design:

A system for measuring 3D simultaneously with spectral image recording in the VideometerLab4 instrument, analysis algorithms to exploit the combined 3D and spectral information, and to demonstrate that this can be utilized efficiently in 1-3 applications.

The requirements and specifications of the final solution is as follows below.

1. The system must deal with at least one of the following scenarios and preferably two.
   (a) 3D measurement while the sphere is moving down (and/or up).
   (b) 3D measurement while the sphere is down.
   (c) 3D measurement while the sphere is down and the conveyor is moving[4].
2. The 3D measurement should at most extend the acquisition time by 30%
3. The system must deal with at least one of the following dimensions of measuring volume and preferably two.
   (a) Lateral ⌀110mm and height 30mm.
   (b) Lateral ⌀95mm and height 30mm.
   (c) Lateral ⌀110mm and height 20mm.
4. The lateral accuracy must be determined by camera resolution. Height accuracy must be as close to lateral accuracy as possible and no lower then 0.1mm.
5. An easy and fast calibration procedure must be designed to calibrate and verify the 3D measurements.
6. The calibration must be verified on at least 2 different instruments to prove transferability of the technology.
7. If a laser is used it must be classified as a class 2 laser or less in accordance with current laser safety regulations. A class 2 laser is safe as the human blink reflex will limit unintended exposure to at most 0.25 seconds. A class 2 laser is limited to 1 mW for continuous beams or more if the emission time is less than 0.25 seconds or if the emitted light is not spatially coherent [2].

[4] In some applications the VideometerLab is used mounted over a conveyor belt.

8. The 3D measurement and calibration must be able to handle varying intensity of objects. In the same manner as the light setup[5] handles the dynamics of the spectral image.

9. The output of the 3D measurement must be a topographical map with the same sampling as the spectral image. All pixels in the 3D image must be handled properly including occluded pixels/unobserved pixels.

10. The 3D geometric calibration and measurement must be consistent with the 2D geometric calibration on a flat surface.

11. The 3D system should minimize the added production cost of the VideometerLab preferably such that it becomes a regular feature rather than an optional feature.

12. An algorithm for segmentation of granular product e.g. rice utilizing the combined spectral and 3D image must be made.

13. The system must be demonstrated in at least one application.

14. When established the 3D measurement must be able to be integrated, and preferably integrated, into the VideometerLab4 instrument and software in a way that works smoothly with the VideometerLab4 hardware and software, and that provides the topographical map as an additional band in the spectral image.

15. Optionally a 3D viewer combining spectral and 3D information can be made.

[5] An integrated software that controls the strobe time for the individual LEDs based upon the reflective properties of the sample. In this way saturated pixels are avoided independently of the sample.

Chapter 2

Possible approaches

Many different methods exist for 3D scanning however for use as an integrated solution in a VideometerLab only two main types of techniques are considered feasible. Structured light solutions and time-of-flight technology. Both techniques perform noncontact 3D surface measurement and are of a physical size allowing them be build into the VideometerLab. However as the VideometerLab already contains a high quality camera with an already established high accuracy calibration procedure it is an obvious choice to make use of this camera. Either combined with a projector or a laser in a structured light approach or in a stereo setup combined with another similar camera that reuses the same calibration procedure. A third option that is also obvious to consider is using a time-of-flight camera looking strait down on the sample area as this will eliminate the problems of occlusion faced by the structured light approaches. The specification of requirements demands the lateral accuracy to be as close to camera resolution as possible and the accuracy in height to be as close to lateral accuracy as possible and no larger then 0.1mm. Ideally giving squared voxels in the 3D model. The camera resolution is 0.0366mm/pixel. According to Odos Imaging[1] such high-resolution 3D images can not be reconstructed using time-of-flight technology [3]. The scale of measurement offered by different techniques are seen in figure 2.1 provided by Odos Imaging [3]. The desired

[1] Odos imaging is a technology focused company specializing in the development and manufacture of vision systems for the capture of high-resolution 3D images, using time-of-flight technology.

solution has to be able to operate in the bottom left corner of the graph which render it impossible to use a time-of-flight solution. Due to the fact that the object size is below 11cm and the desired accuracy and precision is 0.1mm as described in section 1.4 on page 5.

The two different structured light setups considered are a camera-laser setup and a camera-projector setup. Both have the advantage that they utilize the already build in camera in the VideometerLab. An overview of the advantages and disadvantages of both systems are given in table 2.2 and 2.3 respectively. A similar table is also provided for a time-of-flight solution in table 2.1. It is beyond the scope of this project to give a complete and detailed review of state of the art structured light systems and the interested reader is referred to [4] for a detailed description.

The following section 2.1 elaborates slightly on the advantages and disadvantages of using a time-of-flight camera. Section 2.2 elaborates and explains the basis behind laser triangulation for use in a camera-laser setup. Section 2.3 elaborates on and discusses the possibilities of using a camera-projector setup. Finally an overall assessment is conducted in section 2.4 on the basis of the previous sections.

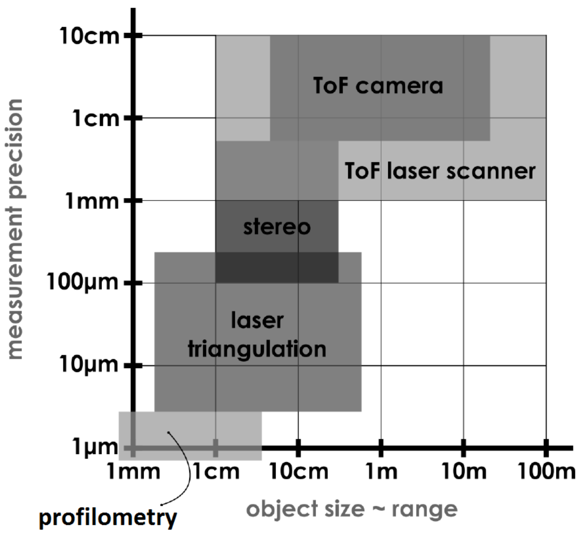


Figure 2.1: *The scale of measurement offered by different modalities [3]. The desired solution has to be able to operate in the bottom left corner of the figure which render it impossible to use a time-of-flight solution as the object size is below 11cm and the desired accuracy is at least is 0.1mm.*

2.1 Time-of-flight camera

The basic principle and simplest version of a time-of-flight camera is to use a single short light pulse. The illumination is switched on very briefly and a light pulse is sent towards the scene. When the light pulse hits the scene part of the pulse is reflected back to the camera. The further away the scene the longer it takes for the light pulse to reach back to the camera and by measuring the flight time of the pulse one can compute the distance to the object. Hence the name a "time-of-flight" camera.

Using a time-of-flight camera looking strait down on the sample area has the great advantage that it eliminates the problems of occlusion faced by the structured light approaches. However price and expected measurement precision make the time-of-flight an undesired solution. A summary of the advantages and disadvantages of a time-of-flight camera are given in table 2.1.

Advantages	Disadvantages
The time-of-flight camera can be mounted to look straight down and parallel to the VideometerLab cameras optical axis. This eliminates the problem of occlusion.	Poor measurement precision [3].
Fast acquisition time.	Most high-end TOF-cameras are physically very big [3]. A small solution is preferred in order to avoid having to change the physical shape of the VideometerLabs exterior.
No post processing necessary to obtain the 3D information.	High end TOF-cameras are very expensive. Videometer wishes the 3D system to minimize the added production cost of the VideometerLab preferably such that it becomes a regular feature rather than an optional feature. This argues against a TOF solution.
	Aligning have to be performed to make the 3D measurement to align with the spectral image.

Table 2.1: *Advantages and disadvantages of a* ***time-of-flight setup****.*

2.2 Camera-laser setup

The basic idea behind the camera-laser setup is to use a classical laser scanning technique. As illustrated in figure 2.2 a single laser line is projected onto the surface of an object and observed with a camera. Based upon the displacement of the laser line an exact retrieval of the 3D coordinates of the objects surface can the computed. By moving the laser in small steps indicated by the arrow z the entire surface can be reconstructed one line at a time.

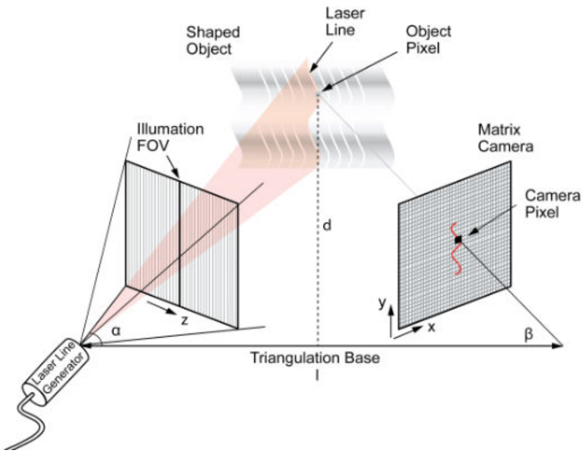
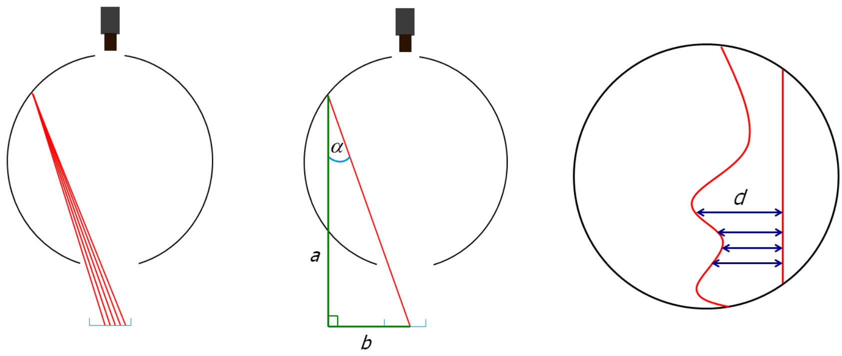


Figure 2.2: *A schematic illustration of the principle behind a laser scanner. From coherent.com.*

The main idea behind using a laser is to take advantage of the fact that the sphere is lowered onto the sample to exclude all external light sources. This movement of the sphere can be used to move a laser line (or multiple laser lines at the same time) across the scanned sample. By taking regular images while the sphere is moving a sequence of images is obtained that can be used to reconstruct the 3D surface geometry. Figure 2.3a on page 12 shows a schematic illustration of a setup with multiple laser lines. The 3D reconstruction performed on each laser line is based upon the right triangle shown in figure 2.3b. If both the angle α and the spheres hight over the sample baseplate is known then the expected distance b can be computed as $b = a \cdot tan(\alpha)$. This length b is the *expected* distance if no objects are present in the scene. If an object is present then the laser will hit the top of the object and the camera will see the laser line further to the left. This means that the *measured* distance will be less then the *expected* distance. The difference is noted d and is proportional to the hight of the object by the relation $h = d \cdot tan(\alpha)^{-1}$. See figure 2.3c. By computing the hight h for each pixel along the laser line for all the laser lines a complete height map of the object can be computed.

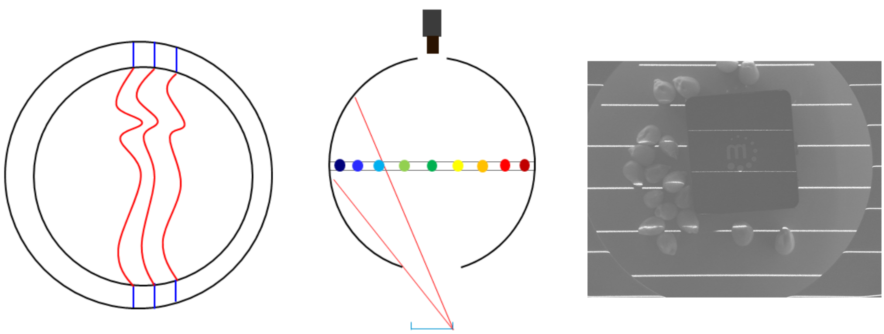
A laser approach has the key advantage of no added acquisition time as the entire acquisition can be done during the spheres moment up and/or down. In a configuration with a conveyor belt the movement of the conveyor belt can be utilized instead giving the system no added acquisition time in this configuration as well. Additionally very high measurement precision and accuracy can be achieved with laser scanners [5]. However this approach assumes that the spheres hight a over the sample baseplate is known. The motor controlling the spheres movement can not accurately enough provide the spheres hight at any given time. Therefore this height has to be estimated for each image. The only way to estimate the height an image is taken in is by observing the position of the laser lines on either the sample (of unknown height) or the baseplate on which the sample is located. As the sample is of unknown height this approach can not be used leaving only to observe the baseplate. Consequently this restricts the sample to only be located in a known part of the image field for instance inside a petri dish. The standard diameter for a petri dish is 95mm and the image field in the VideometerLab is 110mm. The parts of the laser line hitting the baseplate on either side of the petri dish will be flat and can be used to estimate the spheres hight a over the sample. See figure 2.4a. The height can be estimated for the right triangle from figure 2.3b by $a = b \cdot tan(\alpha)^{-1}$. This calculation is based on a relatively short part of the laser line and is therefore expected to introduce uncertainty into the 3D reconstruction.
With VideometerLab4 a frame rate of 25 frames per second is expected and the sphere takes ≈ 1.8 seconds to move from its up position to the down position. Using both the downward and upward movement this gives a total of ≈ 90 frames. Assuming that the geometry allows for one laser to scan the entire 110mm of the image field the laser line would be moving ≈ 1.22mm between each frame. Using multiple laser e.g. 10 laser lines would reduce this distance to only ≈ 0.122mm. Giving a much more usable resolution in the 3D scan. The resolution along the laser line would be determined by the camera resolution. However as seen in figure 2.4b the geometry does not allow for one laser to scan the entire 110mm of the image field without attaching the laser in the bottom half of the sphere which is undesirable as no physical space is available at this position and to place the laser in the bottom half of the sphere the VideometerLabs exterior would need to be changed.
Multiple lasers therefore have to be used as illustrated in figure 2.3a where each laser only lights on the sample in some part of the movement. Figure 2.4c shows an example of an image taken using multiple laser lines of some corn and a small box.

A full list of advantages and disadvantages of a camera-laser setup is given in table 2.2 on page 13.



(a) (b) (c)

Figure 2.3: ***(a)*** *A schematic drawing of the laser-camera setup using multiple laser lines from the same laser. The individual laser lines are created by optics.* ***(b)*** *Knowing both the angle α and the spheres hight a over the sample baseplate the expected distance b can be computed as $b = a \cdot tan(\alpha)$. This length b is the expected distance if no objects are present in the scene. If an object is present then the laser will hit the top of the object and the camera will see the laser line further to the left. This means that the measured distance will be less then the expected distance.* ***(c)*** *The difference in distances is noted d and is proportional to the hight of the object by the relation $h = d \cdot tan(\alpha)^{-1}$.*



(a) (b) (c)

Figure 2.4: ***(a)*** *The blue lines correspond to the part of each laser line used to estimate the height of the sphere at that given time.* ***(b)*** *A schematic drawing of the limitations of only using one laser line. Multiple laser lines are necessary to scan the entire image field by utilizing the spheres up and/or down movement.* ***(c)*** *An example of an image taken using multiple laser lines of some corn and a small box.*

Advantages	Disadvantages
Very high measurement precision and accuracy can be achieved. Small lasers are cheap and fulfill the wish to let the 3D capabilities become a regular feature rather than an optional feature. The physical size of the laser is small and can be fitted into the existing hardware without changing the VideometerLabs exterior. No aligning necessary as only one camera is used. No added acquisition time as the entire acquisition can be done during the spheres moment up and/or down. In the configuration with a conveyor belt is used the movement of the conveyor belt can be utilized instead giving the system no added acquisition time in this configuration either.	The entire image field can not be used as some of the flat background needs to be visible in the image. This is needed in order to compute at which hight the individual image is taken as the sphere moves up and down. The motor controller is not able to provide this information. Occluded pixels/unobserved pixels have to be handled. It might prove difficult to estimate the spheres hight over the sample.

Table 2.2: *Advantages and disadvantages of a* ***camera-laser setup****.*

2.3 Camera-projector setup

Numerous different structured light techniques exist using a camera-projector setup. All have in common that they project a known pattern onto an object and view it from a camera. By analysing how the shape of the object have deformed the projected pattern the 3D shape of the object is recovered.

The different projector guided structured light techniques can be classified into being either a sequential (multiple-shot) or a single-shot technique. If the object desired scanned is static and the requirements to the acquisition time allows a sequential multiple-shot technique this will often give the most reliable and accurate results. However if the object is moving a single-shot technique has to be used to freeze the object in time. The different single-shot approaches may be further divided into three main categories. Techniques using continuously varying structured-light patterns, techniques using 1D encoding schemes (strip indexing), and techniques using 2D encoding schemes (grid indexing). This is illustrated schematically in figure 2.6 on page 16 and an introduction to each technique can be found in [6].

In this project a phase shifting method is chosen as the most suitable method given the specifications to the system (described in section 1.4 on page 5). A review of the specific phase shifting method chosen is given in chapter 3.

The geometry of the VideometerLab may prove a limitation as most projectors have throw ratios around 1.4. The throw ratio of a projector is given by the distance (D) from the lens to surface projected upon, divided by the width (W) of the image that it will project. See figure 2.5. In the VideometerLab the desirable width is $W = 11$cm and the distance $D \approx 38$cm given a throw ratio of ≈ 3.46. If D is maintained a large part of the projected image will therefore be projected outside the cameras field of view and thus a large part of the projectors pixels will be "lost" as they are not seen by the camera.

The 3D measurement is at most allowed to extend the total acquisition time by 30%. This time constraint allows for ≈ 50 images to be taken given the planned hardware for the VideometerLab4 if the time is split $^1/_4$ to image acquisition and $^3/_4$ to post processing and 3D reconstruction. A set of 50 images is more then enough to achieve high precision and resolution using the chosen phase shifting approach.

A full list of advantages and disadvantages of a camera-projector setup is given in table 2.3 on the facing page.

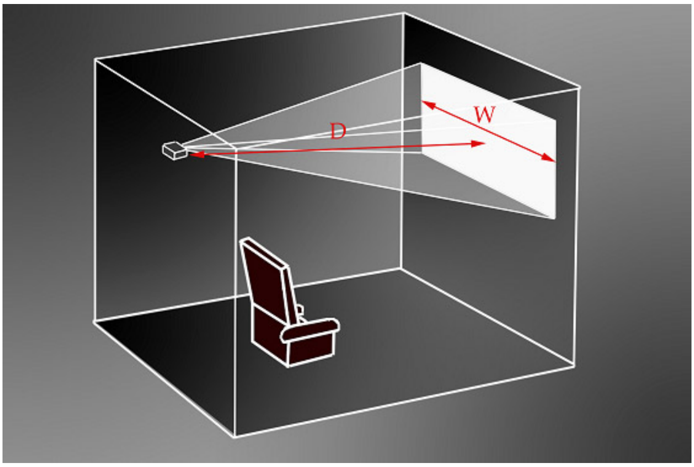


Figure 2.5: *An illustration of a projectors throw ratio. From theprojectorpros.com.*

Advantages	Disadvantages
Very high measurement precision and accuracy can be achieved.	Occluded pixels/unobserved pixels have to be handled.
Small fringe projectors are relatively cheap.	A projector will generate undesirable heat which might introduce noise in the camera images when heating the camera sensor. No ventilation is possible under the shell of the VideometerLab as a ventilation hole would allow for ambient light and dust to enter the instrument.
The physical size of the projector is small and can be fitted into the existing hardware without changing the VideometerLabs exterior.	The post processing may prove computationally expensive.
No aligning necessary as only one camera is used.	Might be time consuming to calibrate.

Table 2.3: *Advantages and disadvantages of a* ***camera-projector setup****.*

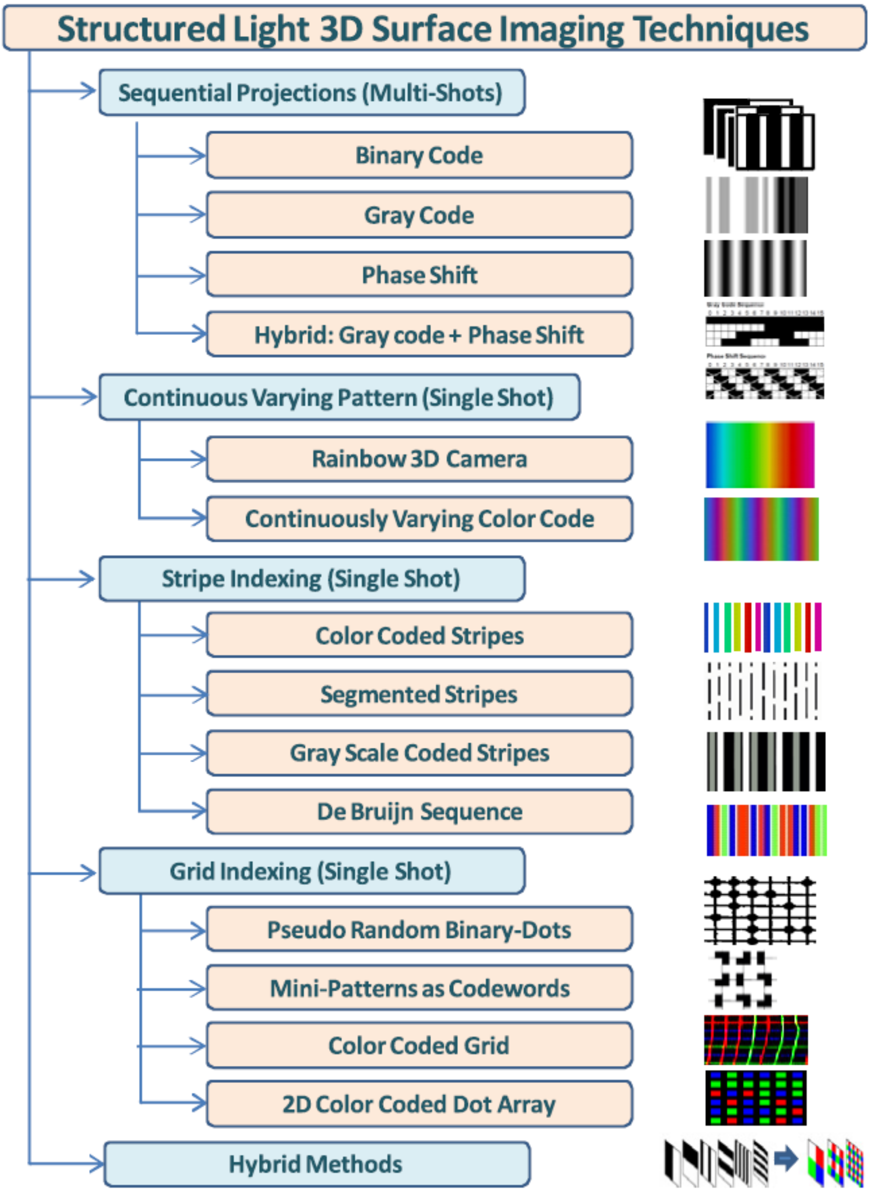
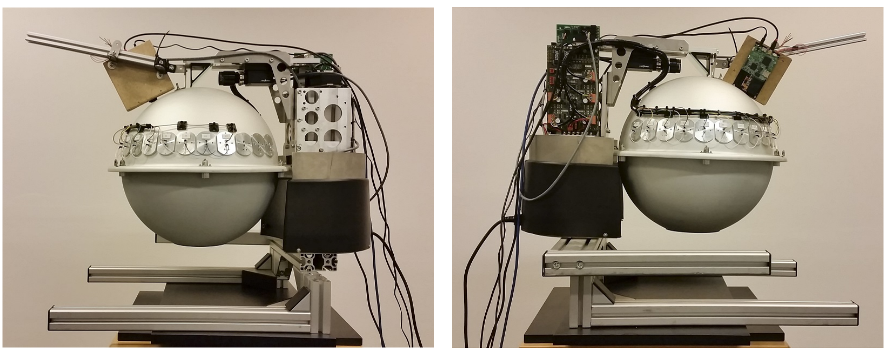


Figure 2.6: *Schematic overview of structured light techniques. Figure from [6].*

2.4 Overall assessment

Based upon the above stated considerations a comprehensive assessment has been made to use a camera-projector setup and perform a variation of phase shifting profilometry (PSP). This method is chosen based on an overall assessment that it will provide the best integrated solution in the geometry of a VideometerLab and best meet all Videometers requirements and specifications listed in section 1.4 on page 5. A picture of the experimental setup is seen in Figure 2.7 below.

Other structured light methods such as binary patterns allow for very accurately identification of outliers and are typically less dependent on surface characteristics like reflections or subsurface scattering and have high signal-to-noise ratios. However PSP has complete scene-coding in just a few projected patterns allowing for fulfillment of the wish for fast acquisition speed while still leaving plenty of time to post processing. PSP is also one of the most common structured light techniques and is well known for its accuracy and simplicity and are implemented in most commercial scanners including products from GOM or Hexagon Metrology, the Breuckmann scanner from Aicon 3D Systems and the Comet from Steinbichler.



(a) *Seen from the back.* **(b)** *Seen from the front.*

Figure 2.7: *The experimental setup with the projector mounted onto the VideometerLab. The projector is seen mounted outside the sphere in the upper right corner in the front view.*

Chapter 3

Phase shifting profilometry

In physics when two 2D wavefronts interfere with each other the resultant intensity pattern formed can be described as

$$I(x,y) = I'(x,y) + I''(x,y)cos[\phi_1(x,y) - \phi_2(x,y) + \delta_k] \tag{3.1}$$

where $I'(x,y)$ is the average intensity which can also be thought of as the intensity bias or ambient light, $I''(x,y)$ is the fringe or intensity modulation and δ_k is the time varying phase shift and lastly $\phi_1(x,y)$ and $\phi_2(x,y)$ are the intensity of the two interfering wavefronts [7]. If the difference in the phase between the two interfering wavefronts is expressed as $\phi(x,y) = \phi_1(x,y) - \phi_2(x,y)$ then the fundamental equation of phase shifting is obtained as

$$I(x,y) = I'(x,y) + I''(x,y)cos[\phi(x,y) + \delta_k] \tag{3.2}$$

where $\phi(x,y)$ is the unknown phase caused by the temporal phase shift of the sinusoidal variation.
In analog times this pattern was made using the interference of two wavefronts. A technique mostly referred to as phase shifting interferometry. Today with the development of digital light processing technology this is done using a projector projecting already sinusoidal patterns and a more modern term is phase shifting profilometry or in the computer vision community often just “phase shifting”.

A number of phase shifting algorithms have been developed e.g. many variations of the three step algorithm and least square algorithms. See e.g [8], [9] or [10]. Another approach is to use Fourier analysis to recover the unknown phase as in [11]. All these approaches have in common that they rely on a set of fringe images being projected at the scene and captured by a camera while the reference phase is varied. They differ in the number of recorded fringe images and the susceptibility of the algorithm to errors in the phase shift or environmental noise such as vibrations or turbulence.
In this project a set of phase varying sinusoidal fringe patterns are used to encode the scene and the 3D topology are reconstructed using Fourier analysis. The overall principle is illustrated in figure 3.1.

The rest of this chapter is organized as follows. Section 3.1 describes a direct formula for computing the phase value ϕ given exactly three phase shifted images. Section 3.2 generalize the formula to N images and section 3.3 shows how to achieve the same result using Fourier analysis. After using either of the three methods the recovered phase will be ambiguous of $2k\pi$, $k \in \mathbb{Z}$ and so needs to be unwrapped as described in section 3.4. Section 3.5 explain how to convert the unwrapped phase to a 3D point cloud using triangulation. Section 3.6 brings it all together and go through an example of the entire pipeline. Finally section 3.7 estimates the accuracy of the 3D data.

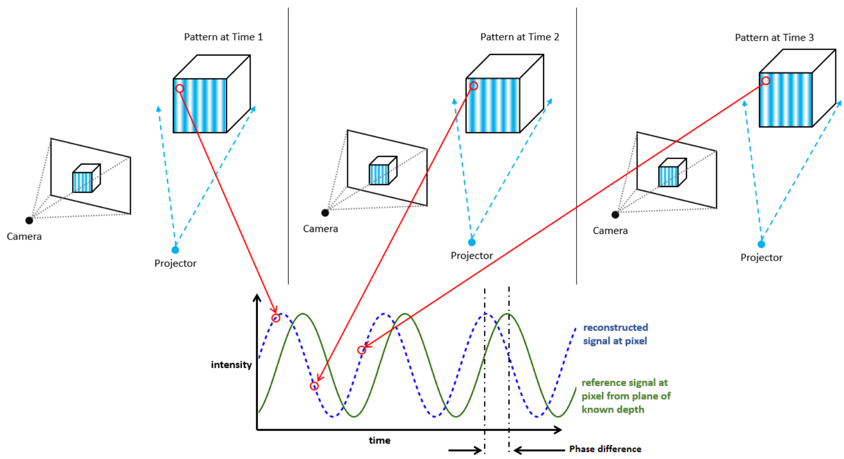


Figure 3.1: *A set of three phase varying sinusoidal fringe patterns are projected onto the scene. A pixel is sampled from each image (red circles and arrows) and the three samples are used to reconstruct the sinusoidal pattern at that pixel (dotted blue sine wave). The phase of the reconstructed signal is compared to the phase of a reference signal for that pixel at a known distance (green sine wave) and the difference in phase computed. The distance from the camera sensor to the object is roughly linearly related to the phase difference. From astinc.us.*

3.1 The three step phase shifting algorithm

In most literature a direct formula is used, but not derived, for computing the phase value ϕ given exactly three phase shifted images. Three images are enough since there is only three unknowns in equation 3.2. Equal phase steps of fixed size α is mostly used making $\delta_k = \{-\alpha, 0, \alpha\}$ for $k = \{1, 2, 3\}$. If $\alpha = 2\pi/3$ and is inserted into equation 3.2 a general solution to the three equations are found

$$\phi(x, y) = tan^{-1}\left(\sqrt{3}\frac{I_1 - I_3}{2I_2 - I_1 - I_3}\right) \tag{3.3}$$

$$I'(x, y) = (I_1 + I_2 + I_3)/3 \tag{3.4}$$

$$I''(x, y) = \frac{2(I_1 + I_2 + I_3)}{3\sqrt{3(I_1 - I_3)^2 + (2I_2 - I_1 - I_3)^2}} \tag{3.5}$$

where for clarity $I_k(x, y)$ is written simply as I_k. The term $\phi(x, y)$ is the phase value of a given pixel and is used to compute from which projector row the light was emitted by the simple relation $y^p = \frac{\phi R}{2\pi}$ where y^p is the row number on the projector DMD which is the projectors equivalent to the cameras sensor and R is the total number of rows on the DMD. Once the phase ϕ (and thereby the projector row) is known for a specific pixel on the camera sensor the 3D world coordinate on the scanned object can be derived through triangulation [5].

3.2 An N-step phase shifting algorithm

Precision and robustness against noise can be gained by making the system overdetermined by using more then three phase shifts and the corresponding number of images. Another way of increasing the precision is to also use multiple phases in a single image, as illustrated in figure 3.1, and then performing phase unwrapping. See section 3.4 on page 24. A more general formula for using N phase shifted images can also be derived as follows [12].

Each projected pattern can be expressed as

$$I_n^p(x^p, y^p) = A^p + B^p cos\left(2\pi f y^p - \frac{2\pi n}{N}\right) \tag{3.6}$$

Where (x^p, y^p) is the row and column coordinate of a pixel in the projector, I_n^p is the light intensity of that pixel in a projector dynamic range from 0 to 1, A^p and B^p are user defined constants (typically set to 0.5), f is the frequency of the sine wave, n represents the phase shift index and N is the total number of phase shifted patterns. If $f \neq 1$ then phase unwrapping is needed as described in section 3.4 on page 24. Without loss of generality let's for now assume that $f = 1$. Generally phase shifting profilometry assumes a linear camera-projector response in practice meaning among other things that the gamma of the projector and camera is set to one. Without considering gamma the intensity of a pixel in the captured image is given by

$$I_n^c(x^c, y^c) = A^c + B^c cos\left(\phi - \frac{2\pi n}{N}\right) \tag{3.7}$$

For clarity in the following the intensity of a given camera pixel $I_n^c(x^c, y^c)$ is written simply as I_n^c. The term A^c is the averaged pixel intensity across the patterns including the ambient light component and is expressed simply as

$$A^c = \frac{1}{N}\sum_{n=0}^{N-1} I_n^c \tag{3.8}$$

Correspondingly the term B^c is the amplitude of the observed sinusoid and can be derived from I_n^c as the following [12]

$$B^c = ||B_{\mathcal{R}}^c + iB_{\mathcal{I}}^c|| = \{{B_{\mathcal{R}}^c}^2 + {B_{\mathcal{I}}^c}^2\}^{0.5} \tag{3.9}$$

where

$$B_{\mathcal{R}}^c = \sum_{n=0}^{N-1} I_n^c cos\left(\frac{2\pi n}{N}\right) \tag{3.10}$$

$$B_{\mathcal{I}}^c = \sum_{n=0}^{N-1} I_n^c sin\left(\frac{2\pi n}{N}\right) \tag{3.11}$$

It is to be noted that if I_n^c is constant or less affected by the projected sinusoid patterns then B^c will be close to zero. As such B^c can be thought of as a signal-to-noise ratio. This is utilized to make a shadow noise detector to remove regions with high shadow noise levels and thereby with small B^c and discard these regions from further processing [11]. Of the remaining pixels with sufficiently large B^c the phase value ϕ of the captured sinusoid pattern can be written as the angle of the complex number $B_{\mathcal{R}}^c + iB_{\mathcal{I}}^c$ and expressed as

$$\phi = tan^{-1}\left(\frac{B_{\mathcal{I}}^c}{B_{\mathcal{R}}^c}\right) = tan^{-1}\left(\frac{\sum_{n=0}^{N-1} I_n^c sin\left(\frac{2\pi n}{N}\right)}{\sum_{n=0}^{N-1} I_n^c cos\left(\frac{2\pi n}{N}\right)}\right) \tag{3.12}$$

Exactly as in section 3.1 once the phase ϕ (and thereby the projector row) is known for a specific pixel on the camera sensor the 3D world coordinate on the scanned object can be derived through triangulation [5]. The only difference between the three step method introduced in section 3.1 and the N-step method introduced in this section is simply the number of phase shifted images used in the reconstruction. Using more images makes the 3D reconstruction more accurate and makes it less sensitive to noise in captured images. It may not be immediately clear that equation 3.3 is just a special case of equation 3.12 with $N = 3$. A proof is conducted in appendix B on page 87.

3.3 Fourier analysis

Another approach is to use Fourier analysis to recover the unknown phase [11]. Let N samples of a single camera pixel be regarded as a single period of a discrete time signal, $x[n]$ for $n = 0, 1, 2, ..., N-1$, then in Fourier terms one can define $X[k]$ for $k = 0, 1, 2, ..., N-1$, using a discrete time Fourier transform. Note that then A^c can be expressed as

$$A^c = \frac{1}{N} X[0] \tag{3.13}$$

and that B^c and ϕ are related to $X[1] = X[N-1]$ according to

$$B^c = \frac{2}{N} ||X[1]|| = \frac{2}{N} ||X[N-1]|| \tag{3.14}$$

$$\phi = \angle X[1] = -\angle X[N-1] \tag{3.15}$$

The frequency terms $X[0]$, $X[1]$ and $X[N-1]$ are referred to as the principal frequency components while the remaining terms are referred to as the non-principal terms and are only the harmonics of $X[1]$. Under ideal conditions the non-principal frequency components $X[k]$ for $k = 2, 3, ..., N-2$ will always be equal to zero. However in the presence of sensor noise these terms can be considered as additive white noise.

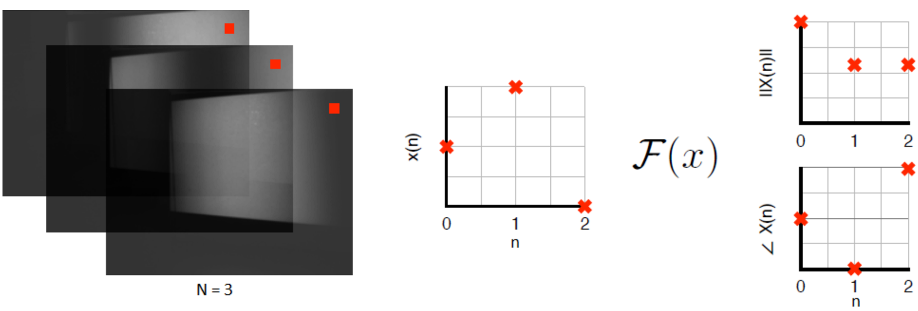


Figure 3.2: *An illustration of the frequency domain interpretation of the phase recovery process for $N = 3$ phase shifted patterns. Illustration from [11].*

3.4 Phase unwrapping

Using either of the three equations derived above (3.3, 3.12 or 3.15) one can recover the phase however only with an ambiguity of $2k\pi$ where $k \in \mathbb{Z}$. This ambiguity manifests itself by discontinuities in the reconstructed phase map every time ϕ changes more then 2π. See figure 3.3a. This ambiguity is only visible if more then one period are used in the projected sinusoidal pattern. I.e. when the frequency (f in equation 3.6) of the sine wave is greater then one. However the higher frequency used in the projected pattern the greater the accuracy can be achieved in the depth reconstruction. When the phase shift becomes greater than 2π, which may also be thought of as $360°$, then the sinusoidal signal overlaps with itself shifted by one or more whole periods and one loses the absolute phase of the signal. In this situation a phase of $10°$ looks identical to a phase of e.g. $370°$ or $730°$. The consequence of this is that one can determine the position of a pixel very accurately within a small window but the absolute position of this window is ambiguous by $2k\pi$. For example if the maximum phase shift corresponds to 10mm and one measures a pixel to have a depth of 0.63mm it is completely unknown whether this pixels actual depth is 0.63mm, 10.63mm, 20.63mm or 30.63mm, etc.

The workaround employed to take the ambiguity into account and correcting it is called phase unwrapping. However this becomes challenging in situations like the one illustrated in figure 3.4 where the scanned surface jumps in depth relative to either the camera or projector. A number of methods have been investigated to enhance the reliability of phase unwrapping by including different forms of a priori spatial constraints through branch cut [13], discontinuity minimization [14], agglomerative-clustering [15], or least-squares [16]. In figure 3.4 is seen a 3D scene consisting of a flexed wall in three sections. The projector lights all three sections however the transition from section A to B and B to C are identical as section B is 4π in length. In the camera only section A and C are visible thus resulting in a missing vertical jump in depth in the 3D reconstruction as the camera can not detect the presence of section B and thinks section A and C are touching. According to Saldner and Huntley [17] the above proposed methods are not able to solve the difficulty of geometric surfaces parallel to the light rays of neither projector nor camera. In order to overcome this challenge a technique called temporal unwrapping is needed. Temporal methods overcome the difficulty illustrated in figure 3.4 and remove the depth ambiguities by the cost of projecting more patterns. This is undesirable if real-time acquisition is needed by the application however if the slightly longer acquisition time can be allowed temporal unwrapping methods are known to be very accurate. For this reason this project applies a temporal unwrapping method called "two wavelength phase unwrapping" by using a single phase cue sequence of three images with wavelength large enough to cover the whole scene [18].

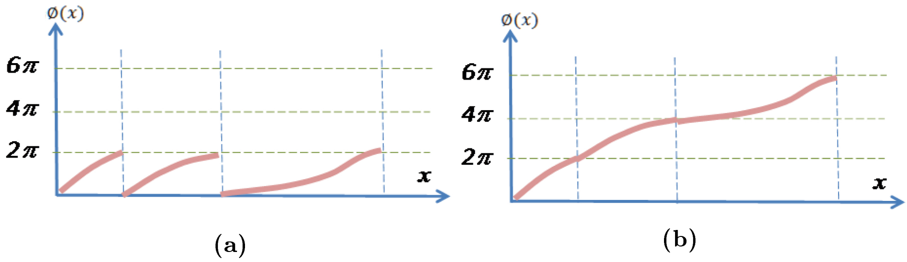


Figure 3.3: ***(a)*** *A profile of a phase map where the phase changes from 0 to* 6π *and therefore have two discontinuous jumps. Each of the three sections are ambiguous of* $2k\pi$ *where* $k \in \mathbb{Z}$. ***(b)*** *The phase unwrapped version of (a) with no ambiguities. Illustration from [6]*

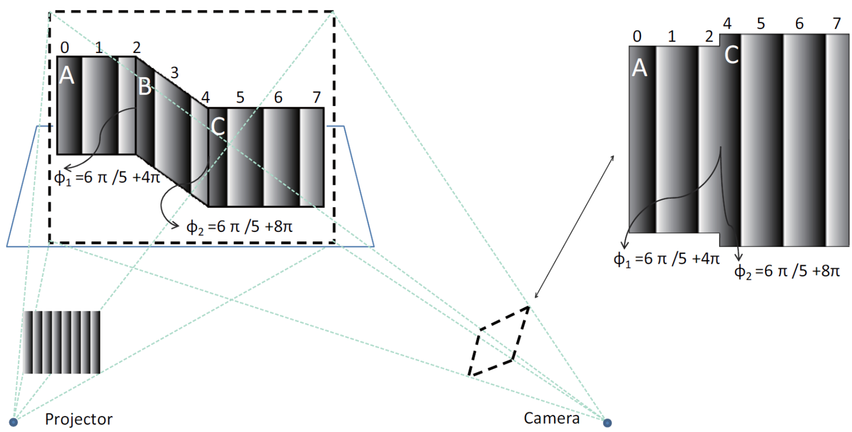


Figure 3.4: *A 3D scene consisting of a flexed wall in three sections. The projector lights all three sections however the transition from section A to B and B to C are identical as section B is* 4π *in length (*$2k\pi$ *with* $k = 2$*). In the camera only section A and C are visible and appear to be touching in 3D as the transition between then is smooth. In this case temporal unwrapping is needed to reconstruct the correct position of section A and C relative to each other. Otherwise the reconstruction would be a missing a vertical jump in depth between section A and C due to the surface of section B being parallel to the light rays of the camera while having a length of* 4π*. Figure from [19].*

3.4.1 Two wavelength phase unwrapping

The idea behind two wavelength phase unwrapping (TWPU) is to use two different wavelengths in two different sets of projected images [8]. One set consisting of a single phase cue sequence of three images with a wavelength large enough to make sure that the geometry never changes more then 2π. As consequence no unwrapping is needed for this sequence. The other set is the main sequence and consists of N images with smaller wavelength in order to achieved greater accuracy. In this case the scanned geometry are likely to change more then 2π somewhere on the surface and thus unwrapping is needed for this sequence.
The use of the single phase cue sequence is to extract the phase with poor accuracy but with zero ambiguity. Then using the main sequence a very accurate phase can be computed however with ambiguities. The ambiguities are then removed by unwrapping the accurate phase while keeping the geometric consistency with the phase of the cue sequence.
Figure 3.5 shows an example of the TWPU algorithm.

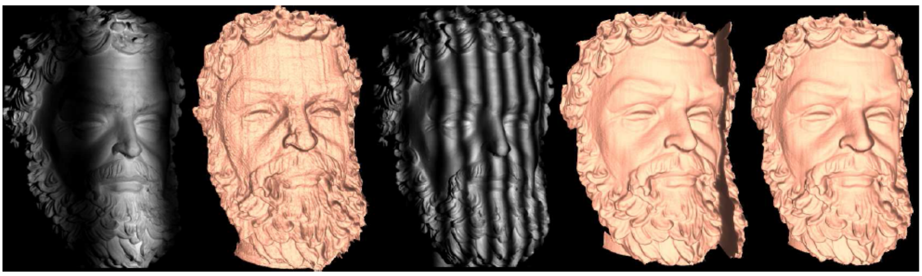


Figure 3.5: *A 3D reconstruction of a statue of a man's face using the two wavelength phase unwrapping algorithm. The first image is one of the fringe images with the longer wavelength. The geometry changes are less than 2π and so 3D information can be retrieved correctly although the quality is poor. The poor quality is seen as the vertical lines in the second image. The third image shows one of the fringe images with a shorter wavelength and here the geometric changes are beyond 2π somewhere on the surface. Thus phase unwrapping cannot correctly reconstruct the geometry as illustrated in the fourth image. But with the reference of geometric information reconstructed with the longer wavelength, the 3D shape can be correctly reconstructed as shown in the last image. The figure and partly this caption is reproduced from [8]. Note that this figure appears more clearly when viewed on a screen. A link to an electronic version of this report can be found in the preface.*

In practice the unwrapping is performed as follows. The phase, also sometimes known as a "phase map", is computed individually for both the phase cue sequence and the main sequence of N images. Then the following term is computed

$$\text{index} = \text{floor}\left(\frac{\phi_{Cue} \cdot \text{nPhase} - \phi_{Main}}{2\pi}\right) \tag{3.16}$$

where nPhase is the number of phases or periods used $(\lambda_1 \lambda_2^{-1})$. The computed index is an integer for each pixel in the camera images which corresponds to the k in the ambiguity term $2k\pi$, $k \in \mathbb{Z}$.

In the literature a formula like equation 3.16 is very rarely written out. Often it is only stated that the goal is to "keep geometric consistency". The only explicit formula found during this project was in non published software made and supplied by Ph.D. J. Wilm the main author of among others [18] and [20]. In the software normal rounding was used in the equivalent to equation 3.16. This project therefore started out using a normal rounding. However by analysing 3D points that was reconstructed with incorrect heights it was found to dramatically reduce the spread of these hight errors by simply using a "floor operation" instead. By "floor operation" is meant to always round to the nearest integer towards minus infinity.
The dominating error source that makes any kind of rounding necessary is ϕ_{Cue} the phase map of the cue sequence. Using a normal rounding would implicate that the errors in ϕ_{Cue} are symmetrically distributed around zero. However in this project the errors are found to almost always be greater then zero with the direct consequence that the computed index values almost always have the correct integer part but due to the noise have a large non zero fractional part. When rounding down one are effectively truncating the noisy fractional part away leaving only the almost always correct integer part.
The reason this behaviour is observed is believed to be due to the fact that in this project either the background or object will always be visible in the entire image plane while also always being inside the cameras depth of field (DOF). The index for pixels where the projectors light is "lost" outside the cameras DOF the errors in the index is believed to be symmetrically distributed around zero (and probably normally distributed) as the signal to noise radio is zero for these pixels. In this project it is known that no light is "lost" and the light is known to always hit something in the camera DOF and it is therefore believed that this alters the distribution of the errors to the one observed in this project.

Knowing the index the final unwrapped phase can then be computed simply as

$$\text{unwrapped} = \frac{\phi_{Main} + 2\pi \cdot \text{index}}{2\pi \cdot \text{nPhase}} \tag{3.17}$$

Where "unwrapped" is the unwrapped phase rescaled between zero and one. This rescale has the advantage that by simply multiplying by the number of projector rows (or columns) the projector row (or column) for each pixel in the camera image can be extracted for triangulation.

However the TWPU method still have some limitations [8]. The phase ambiguity of $2k\pi$ needs to be smaller then the phase error caused by the discretization error of the phase cue sequence. If referring to the wavelengths as λ_1 and λ_2 and the number of bits used for each pixel as n then

$$\frac{\lambda_1}{\lambda_2} < 2^n \tag{3.18}$$

In practice an eight bit camera and eight bit projected images are used. The wavelength λ_1 of the cue sequence covers the whole length of the scene and the shorter wavelength λ_2 can thus be expressed as $\lambda_2 = \frac{\lambda_1}{x}$ for some constant x. Thus 3.18 becomes

$$\frac{\lambda_1}{\frac{\lambda_1}{x}} < 2^8 \Leftrightarrow x < 256. \tag{3.19}$$

Thus equation 3.19 means that the wavelength of the shorter wavelength λ_2 must be larger then $\frac{\lambda_1}{256}$ corresponding to using up to a total of 256 periods. In practice a far less number of periods are typically used. The work done in [18], [8], [19], [12], and [20] uses 8, 8, 14, 16 and 32 periods respectively. In this project 16 and 32 periods is used.

3.5 Triangulation

Once the unwrapped phase map is computed the 3D topology can be reconstructed using point triangulation. The basis of point triangulation is to find the coordinates of a 3D point Q by computing the intersection of the back projected lines of 2D observed points $\mathbf{q_i}$. In noise-free conditions one would only have to find the intersection of these 3D lines. However in the presence of noise one finds the 3D point that is closest to the lines. Figure 3.6 shows a schematic illustration of the point triangulation problem.

Assume that the internal and external parameters of both the camera and the projector is known. How to obtain these parameters are discussed in chapter four. Let Q be a 3D point with known projections $\mathbf{q_c}$ and $\mathbf{q_p}$ in the camera and projector respectively and let $\mathbf{P_c}$ and $\mathbf{P_p}$ be the pinhole camera model for both the projector and the camera respectively. Recall that the pinhole camera model is

$$\mathbf{q_i} = \mathbf{A}[\mathbf{R}\ \ \mathbf{t}]Q_i = \mathbf{P}Q_i\ , \quad \mathbf{P} = \mathbf{A}[\mathbf{R}\ \ \mathbf{t}] \tag{3.20}$$

where Q_i is a 3D point with the projection $\mathbf{q_i}$ in the camera defined by $\mathbf{P}$. The pinhole camera model will be the basis of the rest of this section. Additional theory and a deduction of the pinhole camera model can among others be found in H. Aanæs' lecture notes [5].

In the following section 3.5.1 derives a simple algorithm for point triangulation and on the basis of this section 3.5.2 derives a computationally effective algorithm for simultaneous triangulation of a large number of 3D points.

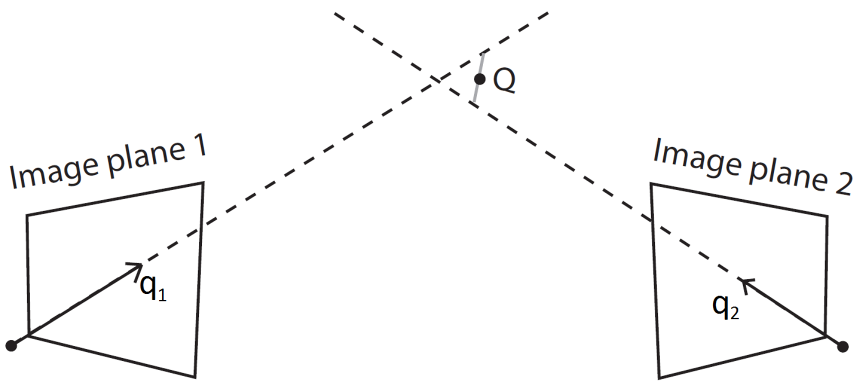


Figure 3.6: *The result of point triangulation is the 3D point Q closest to the back projections of the observed 2D points q_1 and q_2 respectively. This figure and caption is adapted from [5].*

3.5.1 The conventional algorithm

Almost any textbook on computer vision will present the following linear algorithm for triangulation as it is the basis for most more sophisticated algorithms. This linear algorithm has the advantage of simplicity at the cost of computational efficiency. However this algorithm provides a necessary starting point for the computational efficient algorithm applied in this project.

Let $\mathbf{P_c}$ donate the camera matrix and $\mathbf{P_p}$ donate the camera matrix computed for the projector. How to compute these are discussed in chapter four. To ease notation the rows are donated with a superscript

$$\mathbf{P_c} = \begin{bmatrix} P_c^1 \\ P_c^2 \\ P_c^3 \end{bmatrix} \quad , \quad \mathbf{P_p} = \begin{bmatrix} P_p^1 \\ P_p^2 \\ P_p^3 \end{bmatrix} \tag{3.21}$$

allowing for the pinhole camera model (3.20) to be written as[1]

$$\mathbf{q_i} = \begin{bmatrix} s_i x_i \\ s_i y_i \\ s_i \end{bmatrix} = \begin{bmatrix} P_c^1 \\ P_c^2 \\ P_c^3 \end{bmatrix} Q \Rightarrow \tag{3.22}$$

$$s_i x_i = P_c^1 Q \quad , \quad s_i y_i = P_c^2 Q \quad , \quad s_i = P_c^3 Q \Rightarrow \tag{3.23}$$

$$x_i = \frac{s_i x_i}{s_i} = \frac{P_c^1 Q}{P_c^3 Q} \quad , \quad y_i = \frac{s_i y_i}{s_i} = \frac{P_c^2 Q}{P_c^3 Q} \tag{3.24}$$

With the use of some arithmetic equation 3.24 becomes

$$x_i = \frac{P_c^1 Q}{P_c^3 Q} \quad , \quad y_i = \frac{P_c^2 Q}{P_c^3 Q} \Rightarrow \tag{3.25}$$

$$P_c^3 Q x_i = P_c^1 Q \quad , \quad P_c^3 Q y_i = P_c^2 Q \Rightarrow \tag{3.26}$$

$$P_c^3 Q x_i - P_c^1 Q = 0 \quad , \quad P_c^3 Q y_i - P_c^2 Q = 0 \Rightarrow \tag{3.27}$$

$$(P_c^3 x_i - P_c^1) Q = 0 \quad , \quad (P_c^3 y_i - P_c^2) Q = 0 \tag{3.28}$$

Equation 3.28 contains linear constraints in Q and since Q has three degrees of freedom at least three of such constraints are needed to determine Q. This may also be seen as a set of linear equations in Q. The 3D point Q only has three unknowns namely its x, y and z coordinate and as such only three equations are enough to compute Q.
This corresponds to e.g. knowing both coordinates of the projection of Q into the camera $\mathbf{P_c}$ and only knowing one of the coordinates of the projection of Q into the projector $\mathbf{P_p}$ and that is precisely the situation in this project.

[1] The argumentation in equation 3.22 to equation 3.28 is made using the camera $\mathbf{P_c}$ as example. Exactly the same result can be made with the projector $\mathbf{P_p}$ if substituted for $\mathbf{P_c}$.

To compute Q based only on these three linear constraints they are stacked to form a matrix

$$\mathbf{B} = \begin{bmatrix} P_c^3 x_i - P_c^1 \\ P_c^3 y_i - P_c^2 \\ P_p^3 x_i - P_p^1 \end{bmatrix} \tag{3.29}$$

making equation 3.28 equivalent to

$$\mathbf{B}Q = \mathbf{0} \tag{3.30}$$

In all practical situations noise will be present to some extent and as such equation 3.30 will not hold perfectly and instead one solves

$$\min_{Q} = ||\mathbf{B}Q||_2^2 \tag{3.31}$$

The minimisation problem in equation 3.31 is seen to be a least squares problem and is solved straight forward e.g. using singular value decomposition. In Matlab notation it would simply be `[u,s,v]=svd(B); Q=v(:,end);`. However despite it is easy to compute *one* Q using equation 3.31 recall that with this approach one will have to solve equation 3.31 once for *all* of the $\approx$ 4,500,000 points in a standard VideometerLab4 scan. It can obviously be parallelized however it is still not a feasible solution.

3.5.2 A computationally effective alternative

Recall that in the "problem statement and specifications" section on page 5 in chapter one it is specified that "the 3D measurement should at most extend the acquisition time by 30%". A standard VideometerLab4 scan contains $\approx$ 4,500,000 points to be triangulated and as such a computationally effective algorithm is needed. In this section an algorithm is derived that allows for computing the 3D coordinates of the *entire scene all at once* in a fast and vectorized manner. The execution time of the triangulation of the 4,800,000 points that make up the LEGO[2] bricks seen in Figure 3.13 on page 41 was measured to 0.59 seconds on one PC[3] and 0.99 seconds on another PC[4]. The method used to efficiently solve equation 3.30 for such a large number of points is derived in the following.

[2] LEGO® is a trademark of the LEGO Group of companies which does not sponsor, authorize or endorse this thesis.

[3] AMD FX-8350 eight-core processor 4.00 GHz, 8 GB ram, Windows 7 professional (64 bit), Matlab 2015b (64 bit).

[4] Intel i7-4702MQ 2.20 GHz quad-core processor, 8 GB ram, Windows 8.1 professional (64 bit), Matlab 2015b (64 bit).

It is clear that any vector Q orthogonal to all rows in $\mathbf{B}$ would satisfy $\mathbf{B}Q = \mathbf{0}$. The vector cross product can normally be used to find orthogonal vectors however the vector cross product is only defined for 2 vectors in 3 dimensions. However a generalisation for $n-1$ vectors in an n-dimension linear space w can be made as follows [21]

$$\langle w, v_1 \times \ldots \times v_{n-1} \rangle = \det(v_1, \ldots, v_{n-1}, w) \tag{3.32}$$

The following equality holds and per definition v_k is always orthogonal to it [22]

$$v_1 \times \ldots \times v_{n-1} = \sum_k \det(v_1, \ldots, v_{n-1}, e_k) e_k \tag{3.33}$$

The term e_k is the i^{th} column of the identity matrix of size $n \times n$. As a direct consequence of equation 3.32 and 3.33 a solution to equation 3.30 is given by the generalised vector cross product of the rows of 3×4 matrix $\mathbf{B}$

$$Q = \sum_{k=1}^{4} \det\left(P_c^3 x_i - P_c^1 \ , \ P_c^3 y_i - P_c^2 \ , \ P_p^3 x_i - P_p^1 \ , \ e_k\right) e_k \tag{3.34}$$

Note that the resulting Q is in homogeneous coordinates. By utilizing the linearity and antisymmetry of the determinant tensor the k^{th} component of Q can de written as [22]

$$\boldsymbol{Q}^k = \boldsymbol{D}^k_{1,2,1} - \boldsymbol{u_c} \boldsymbol{D}^k_{3,2,1} - \boldsymbol{v_c} \boldsymbol{D}^k_{1,3,1} - \boldsymbol{u_p} \boldsymbol{D}^k_{1,2,3} + \boldsymbol{u_p} \boldsymbol{u_c} \boldsymbol{D}^k_{3,2,3} + \boldsymbol{u_p} \boldsymbol{v_c} \boldsymbol{D}^k_{1,3,3} \tag{3.35}$$

where $\boldsymbol{D}^k_{i,j,l} = \det\left(P_c^i, P_c^j, P_p^l, e_k\right)$ are constants that can be precomputed as they only depend on the camera and projector matrices, $\mathbf{P_c}$ and $\mathbf{P_p}$ respectively. The terms $\boldsymbol{u_c}$ and $\boldsymbol{v_c}$ are both $h \times w$ matrices containing the cameras image coordinates in the horizontal respectively vertical direction[5]. The term $\boldsymbol{u_p}$ is also a $h \times w$ matrix containing the projectors horizontal image coordinates[6]. The term $\boldsymbol{Q}^k$ is also a $h \times w$ matrix and as defined contains the k^{th} component of Q meaning the k^{th} part of the homogeneous coordinate for *all* the 3D points. As both $\boldsymbol{u_c}$ and $\boldsymbol{v_c}$ are constant matrices defined exclusively by the resolution of used camera[7] a large part of $\boldsymbol{Q}^k$ can be precomputed as well. This leads to the following reordering of equation 3.35

$$\begin{aligned}
\boldsymbol{Q}^k &= \left(\boldsymbol{D}^k_{1,2,1} - \boldsymbol{u_c} \boldsymbol{D}^k_{3,2,1} - \boldsymbol{v_c} \boldsymbol{D}^k_{1,3,1}\right) - \boldsymbol{u_p} \left(\boldsymbol{D}^k_{1,2,3} - \boldsymbol{u_c} \boldsymbol{D}^k_{3,2,3} - \boldsymbol{v_c} \boldsymbol{D}^k_{1,3,3}\right) \\
&\Downarrow \\
\boldsymbol{Q}^k &= \boldsymbol{S}^k_1 - \boldsymbol{u_p} \boldsymbol{S}^k_2
\end{aligned} \tag{3.36}$$

5 With respect to the camera sensor.
6 With respect to the projector DMD or “sensor” when thought of as a camera.
7 In Matlab notation it would simply be
```
[uc, vc] = meshgrid(0:size(im, 2)-1, 0:size(im, 1)-1);
```
where `im` contains an image taken with the camera.

where both $\boldsymbol{S}_1^k$ and $\boldsymbol{S}_2^k$ can be precomputed and are given by

$$\boldsymbol{S}_1^k = \boldsymbol{D}_{1,2,1}^k - \boldsymbol{u_c} \boldsymbol{D}_{3,2,1}^k - \boldsymbol{v_c} \boldsymbol{D}_{1,3,1}^k \tag{3.37}$$

$$\boldsymbol{S}_2^k = \boldsymbol{D}_{1,2,3}^k - \boldsymbol{u_c} \boldsymbol{D}_{3,2,3}^k - \boldsymbol{v_c} \boldsymbol{D}_{1,3,3}^k \tag{3.38}$$

This dramatic precomputation of $\boldsymbol{S}_1^k$ and $\boldsymbol{S}_2^k$ is the reason why over 4,500,000 points can be triangulated in less then a second as the triangulation is condensed into just four element wise matrix multiplications and four element wise matrix subtractions. The triangulated coordinates found by equation 3.36 are in homogeneous coordinates and so to convert to cartesian coordinates an extra *three* element wise matrix divisions is needed.

3.6 Example and implementation tricks

This section shows an example of how to use the theory described in this chapter to do a 3D scanning. The object scanned is five LEGO[8] bricks built together standing on a white piece of paper. The figures mentioned in this section can be found on page 38 to 41.

3.6.1 Capture a sequence of images

The first step is to capture one image of each pattern in a sequence of phase varying sinusoidal fringe patterns projected onto the bricks. Figure 3.7 shows the 12 images captured for the example in this section. The first nine images are used as the main sequence (Figure 3.7a -3.7i) and the last three images are used as the cue sequence (Figure 3.7j-3.7l). Figure 3.8 shows the corresponding patterns as they are projected by the projector. The patterns are made as described by equation 3.6 using 16 phases i.e. $f = 16$. If one counts one will see that exactly 16 horizontal stripes are present in image a-i in Figure 3.8. The cue sequence is per definition made using only a single phase i.e. $f = 1$.

3.6.2 Mask out unwanted pixels

The opening of the VideometerLabs sphere is circular and can be seen in the corners of the captured images in Figure 3.7. These areas are masked out by

[8] LEGO® is a trademark of the LEGO Group of companies which does not sponsor, authorize or endorse this thesis.

applying the mask seen in Figure 3.9a where the light gray area is kept and the black corners are removed from further processing. If any pixels are saturated or have not resized any light at all then these pixels are discarded as well. As eight bit images are used this translates to simply discarding all pixels having a value of either 0 or $2^8 - 1 = 255$.

3.6.3 Compute ϕ and A^c

Using the Fourier approach from section 3.3 and in particular equation 3.15 the phase is computed for both the main and cue sequence. It might be seen as inefficient to compute a full Fourier transform when only a few components are needed. But in practice it is often so well implemented that it actually provides better performance then the N-step approach from section 3.2.
Figure 3.10a shows the phase of the main sequence and Figure 3.10b shows the phase of the cue sequence. A shading image (A^c) is also computed using equation 3.13 and is seen in Figure 3.9b. Note that as the camera is looking vertically down on the bricks one loses the sense of depth and the bricks appear flat without any height.

3.6.4 Compute B^c and the noisemask

As described in section 3.2 if part of the bricks can be seen by the camera but are hidden from the projectors point of view (for example due to self-occlusion) then I_n^c (in equation 3.7) will be constant or less affected by the projected sinusoid patterns and therefore will B^c be close to zero. As such B^c can be thought of as a signal-to-noise ratio. By thresholding the B^c image a noise mask is made containing regions with high shadow noise levels. The B^c image is seen in Figure 3.11 together with its histogram. Note that the background has very high B^c values as the white paper is reflecting almost all of the projected light into the camera. Notice also the dark blue region to the right of the bricks. This region is in constant shadow behind the bricks as the projectors light is coming down at an angle from the left. Similar dark blue regions are seen to the right of each stud[9]. A threshold is determined based on the histogram of B^c image. Three peaks are seen in the histogram in Figure 3.11. The peak around zero is noisy pixels with very low signal-to-noise ratio The peak around 150 is mostly pixels of the bricks and the peak around 200 is mostly the white paper which is highly reflective. A threshold value of 50 is therefore used in this case. The resulting noise mask can be seen in Figure 3.9c.

[9] The studs are the small cylindrical bumps situated on top of the bricks.

The value of this threshold is dependent on the different scanning parameters and the colour of the scanned objects. In this project the value has been chosen manually for each different type of scan. The same value has been used on all scans of barley and similar seeds, and value another have been used for all scans of LEGO[10] bricks, and another value when scanning a printed checkerboard during calibration. The determination of a suitable threshold may be made automatic by analysing the histogram in Figure 3.11. The details are presented in the future work section on page 83.

The light yellow, green and bluish colour next to the left edge of the image is probably due to unintended internal light reflections inside the sphere. This can easily be avoided by adjusting the patterns in Figure 3.8 to be dark outside the cameras field of view. More details are found in the future work section.

On the immediate left of the bricks vertical stripes appear on the background. This is due to unwanted reflections generated by the smooth blank plastic sides of the bricks. These reflections locally alter the projected pattern appearing on the background and thereby result in faults in the 3D reconstruction of this region. It is less visible in Figure 3.11 but a similar effect is seen on the two studs to the left of the top brick. The reflections effect on the 3D reconstruction can be seen in Figure 3.13.

3.6.5 Perform unwrapping

Unwrapping is performed by simply computing the index given by equation 3.16 after which equation 3.17 is used to compute the unwrapped phase map. The vast majority of the area in the noise mask have so low signal to noise ratio that the main phase map ϕ_{Main} appear almost random in these areas. This is clearly seen in the area to the right of the bricks in Figure 3.10a and can be compared to the noise mask in Figure 3.9c. For this reason all index values inside the noise mask are discarded. The result is seen in Figure 3.12a.

Even though the index map is almost noise free some noise is still present and seen as scattered dots in the index map. A noise reduction method has been developed to accurately remove the noisy scattered dots from the index map:

1. An index value less then zero makes no sense so discard all negative values.

[10] LEGO® is a trademark of the LEGO Group of companies which does not sponsor, authorize or endorse this thesis.

2. Loop through all possible index values i.e. stripes in Figure 3.12a. Note that this will never take more iterations than the number of phases used in the encoding[11] i.e. 16 in this example. In this project 32 phases is the maximum number of phases used.

3. Fill in any hole in the stripe that differs in index value *if* the hole is more then 0.55 mm away from the edge of the stripe. In the VideometerLab4 this corresponds to 15 pixels[12].
   The reason why it is necessary to keep a little distance to the edge of the stripe is that the transition from one stripe to the next is not completely sharp but a smooth transition over a distance of ≈ 1 mm.

Figure 3.12 shows the index map before and after performing the above described noise reduction. When comparing the two figures note how the noisy speckle pattern in Figure 3.12a is removed in Figure 3.12b.

After computing the unwrapped phase map a small 3×3 median filtering is performed for further noise reduction. The physical size of the filter is tiny $\approx$ 0.11mm$\times$0.11mm and so no important details are considered lost by the filtering. The resulting unwrapped phase map is of very high quality and considered immensely close to noise-free.

The unwrapped phase map is seen in Figure 3.10c and is computed based on the noise reduced index map seen in Figure 3.12b.

If the index value is changed by one the resulting difference in reconstructed height is between 15.80mm and 16.74mm if 32 phases are used and between 30.40mm and 32.20mm if 16 phases are used. As a natural consequence if scanning with 16 phases and if the 3D scanned object is not taller then 15.80mm then it is not needed to capture a cue sequence as one a priori knows that the surface topology never changes more then 2π. The same is true for objects under 30.40mm in height scanned with 16 phases. Unwrapping can then be performed using either of the methods hereinafter described in 3.6.5.1 and 3.6.5.2.

3.6.5.1 Spectral phase unwrapping

Theoretically the cue sequence is not needed as the main sequence can be unwrapped using *spectral* phase unwrapping. This is done by taking the neighbouring pixels into account and unwrapping the current pixel with respect to its

[11] Depending on the terminology this is $f = 16$ in equation 3.6 on page 21 or nPhases = 16 in equation 3.16 on page 27.

[12] As one pixel in the VideometerLab4 has a side length of 0.0366mm.

neighbour. In this kind of unwrapping it is assumed that every time the phase changes more then 2π it must have been "wrapped". So the phase is unwrapped by simply adding or subtracting multiples of 2π at each pixel until it is within 2π of its neighbour. If done rowwise the first pixel in the row is arbitrarily chosen as starting point and assumed to have the correct phase value. However if the first pixel in a row is noisy then this noise will be used as reference for unwrapping the rest of the row and the noise is thereby propagated to the rest of the row. This will result in the entire row being unwrapped incorrectly by some multiple of 2π. To correct for this one could perform the same unwrapping operation on the transposed result and thereby correct any incorrectly unwrapped rows. However in the presence of NaN's a more robust solution is needed. This could be achieved by e.g. mean filtering the unwrapped phase by a large vertical filter, then rounding to the nearest 2π and finally subtracting the difference. This spectral method does not work well enough in practice and is therefore not used in this project.

3.6.5.2 Hardcoded phase cue unwrapping

Another approach is to simply scan a cue sequence *once* during scanner calibration and simply reuse this scan as the cue sequence in *all* subsequent scans. This might sound rather simple however it outperforms both the traditional unwrapping and spectral unwrapping. In practice the only difference is to reuse the same cue phase map over and over again instead of computing a new one each time[13].

3.6.6 Triangulate the 3D topology

Once the unwrapped phase map is computed the 3D topology is reconstructed using point triangulation. This is done using formula 3.36 derived in section 3.5. Figure 3.13 shows the 4,800,000 points that constitute the 3D point cloud of the bricks used as example throughout this section. Figure 3.14 shows a zoomed in view of two of the studs. As the camera is looking vertically onto the bricks it can not see the vertical sides of the bricks. This results in "holes" in the point cloud. However no 3D information is lost as every (x, y) point *is* having a z-coordinate as well[14]. The point cloud can therefore be thought of as representing the discrete function $f(x, y) = z$.

[13] Using equation 3.16 as it was described in section 3.4.1.
[14] Excluding the points removed from processing by the noise mask.

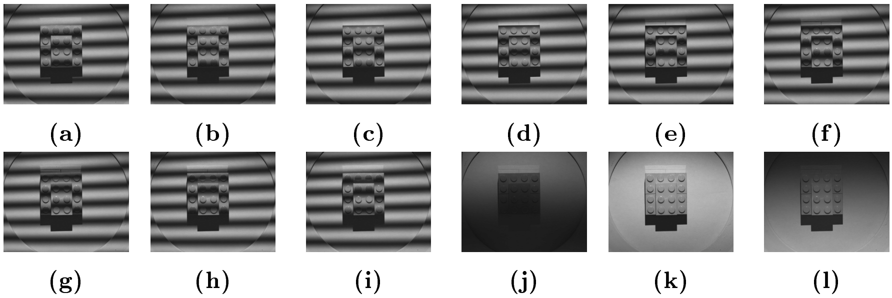


Figure 3.7: *A sequence of 12 images of five stacked LEGO bricks while a sequence of phase varying sinusoidal fringe patterns are projected upon them. The first nine images are used as the main sequence and the last three images are used as the cue sequence.*

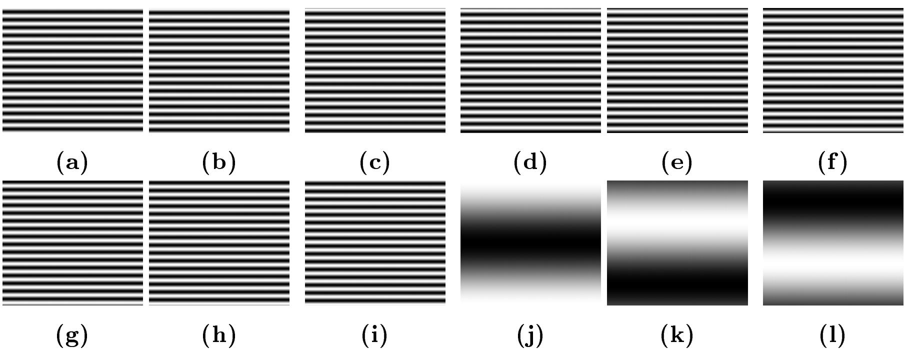
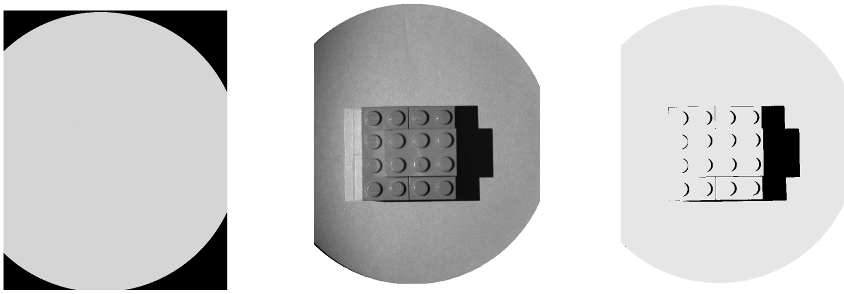
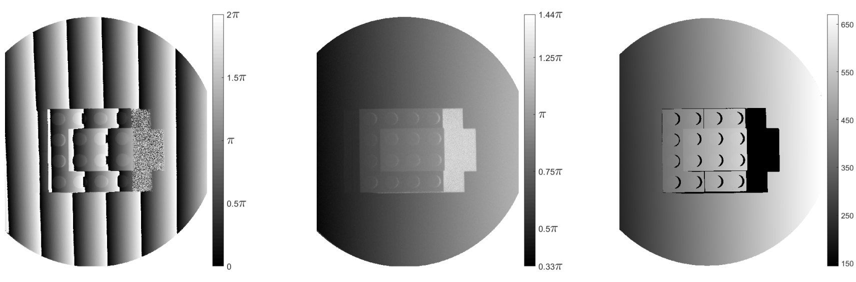


Figure 3.8: *The projected patterns corresponding to images of Figure 3.7. The images look similar, but the stripes move in the vertical direction as described by equation 3.6.*



(a) (b) (c)

Figure 3.9: *(a) The mask used to remove the visible corners of the opening in the VideometerLab's sphere. The light gray area is kept and the black corners are removed from further processing. Any saturated pixels are removed with this mask as well. (b) The shading image (A^c) computed using equation 3.13. Note that the camera is looking vertically down on the bricks and thus one loses the sense of depth and the bricks appear flat without any height. (c) The noise mask computed by thresholding in the B^c the image seen in Figure 3.11. The light gray area is kept and the black parts are discarded. Notice that the black regions are to the right of the bricks themselves and to the right of all the studs as these regions are in constant shadow as the projectors light is coming down at an angle from the left.*



(a) *Main sequence* (b) *Cue sequence* (c) *Unwrapped*

Figure 3.10: *Phase maps of the main sequence and the cue sequence respectively and the unwrapped phase map.*

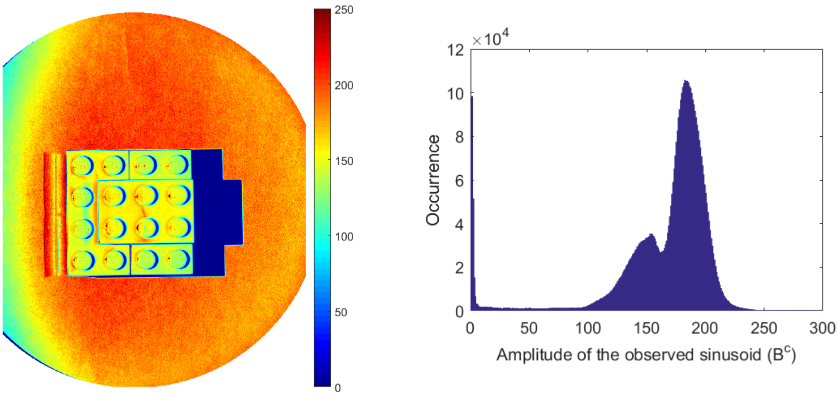
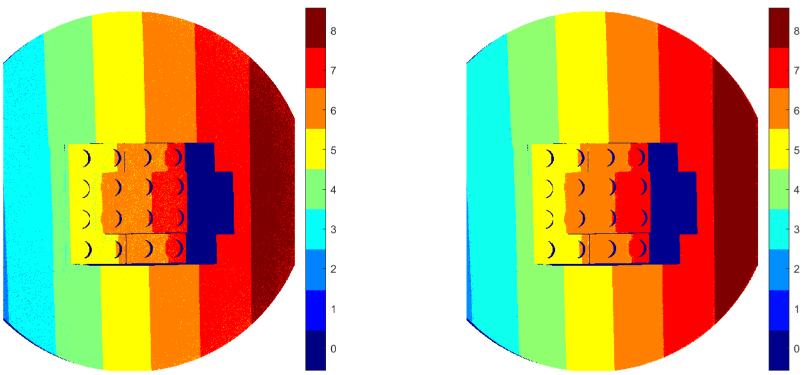


Figure 3.11: *The B^c image together with its histogram.*



(a) *Before noise reduction* **(b)** *After noise reduction*

Figure 3.12: *The index map used for unwrapping. Notice how the noisy speckle pattern in* ***(a)*** *is removed in* ***(b)****.*

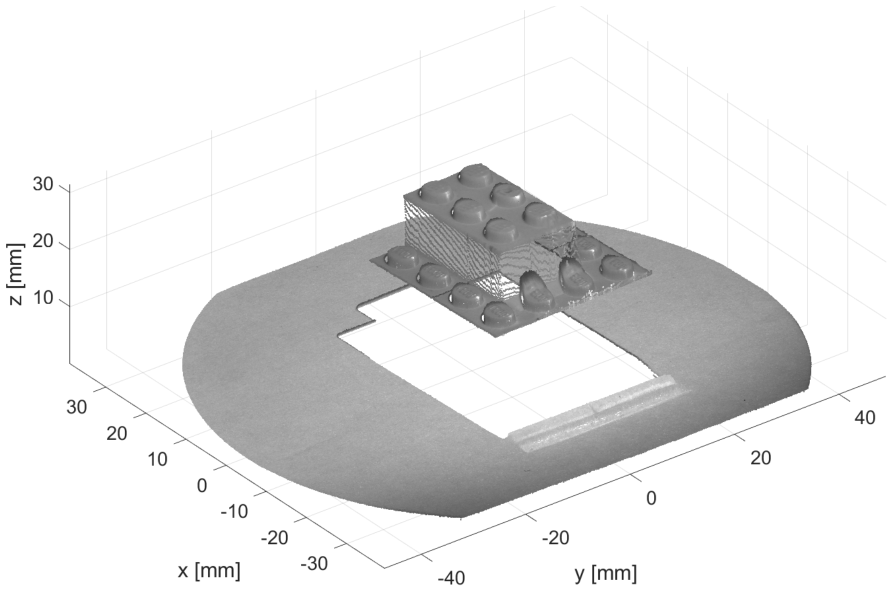


Figure 3.13: *The 3D point cloud of the scanned LEGO bricks made up by 4,800,000 points. The points are coloured using the shading image. The apparent holes in the figure are vertical parts of the bricks.*

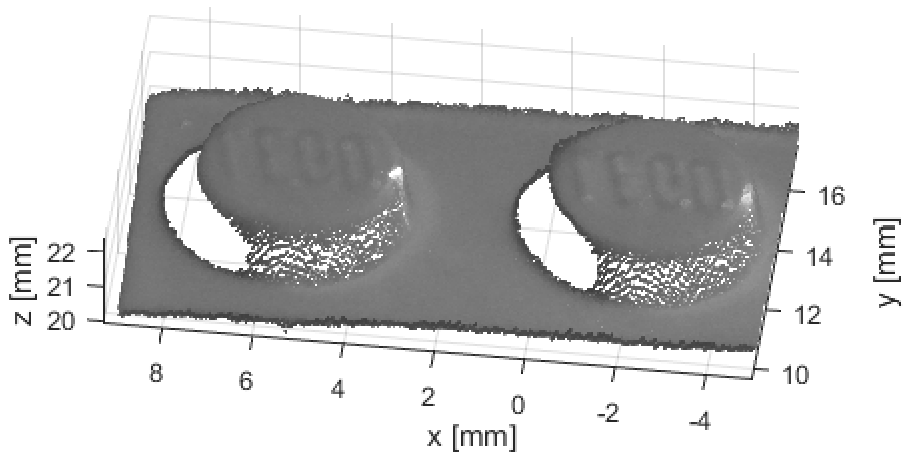


Figure 3.14: *A zoomed in view showing two of the studs. Note that this figure appears more clearly when viewed on a screen. A link to an electronic version of this report can be found in the preface.*

3.7 3D accuracy

The lateral accuracy of the 3D measurement is determined by half of the diagonal size of a camera pixel in the measurement area, namely 0.0259mm. In order to estimate the height accuracy an approach was taken that is similar to the relevant ISO standard. A flat plane was 3D scanned and the deviation from perfectly flat measured. Based upon this an estimate of the 3D height accuracy is computed.

The scanned plane was the back of a flat 1mm thick metal plate that had been used as calibration target as described in chapter four. A plane was fitted to the resulting point cloud and for each point the perpendicular deviation from the fitted plane was computed and regarded as height error. This implicitly assumes that the metal plate is perfectly flat. The error is visualized in Figure 3.15 on page 44. It is to be noted how the circular error pattern is roughly the same shape and size as the region between the wrist and the palm of the human hand. When the calibration target was made a piece of paper was spray glued on the front side of the plate and pressed together by hand to make sure it was glued together properly. For this reason it was hypothesized that maybe the metal plate was deformed very subtle (≈ 0.3mm) when it was manually pressed together. In an attempt to reject this hypothesise and to get a more accurate error estimate a 10mm thick plate of non tempered float glass[15] was purchased and 3D scanned. As the glass is transparent it can not be 3D scanned without making it opaque. Several approaches was tried out and are summarised below.

First a mat spray was applied on top of the glass to give the glass a dull and non reflective surface. As this was not enough to prevent light from passing through the glass a white piece of paper was placed under the glass and manually held in place. However as seen in Figure C.1 in appendix C on page 89 this did not provide a good 3D scan and a clear pattern was seen in the error. The error is also far greater then the one seen for the metal plate (see Figure 3.15). The high positive errors (red parts in Figure C.1) was seen to move when a different grip was used to hold the paper in place. Therefore instead of manually holding the paper in place a white spraypaint was applied to the back of the glass. Resulting in a much more uniform error pattern and roughly halving the error as seen in Figure C.2. Ph.D. J. Wilm the main author of among others [18] and [20] has good experiences with the use of the two above mentioned sprays which inspired their use. The sprays was however found not to be suited for this application.

[15] Float glass is made by continuously pouring liquid glass ($\approx 1500°C$) out of a melting furnace and on to a large bath of molten tin ($\approx 1100°C$) The glass floats on the tin and is by gravity spread of in a planar layer. During tempering there is a risk of the glass deforming slightly giving the glass a subtle uneven surface. For this reason the used glass is not tempered. A 10mm thick plate was used to ensure an as rigid plate as possible.

A new approach was therefore taken and a white piece of paper was spray glued onto the front of the glass plate[16]. As the glass is 10mm thick and very rigid there was considered to be no risk of deforming the glass surface when applying slight pressure to assure the paper has been glued firmly onto the glass. The only expected imperfections from a perfectly flat surface are considered to be the layer of glue that may not be completely homogeneous and the surface of the paper where the fibres are slightly visible. The found deviation from a perfect plane is seen in Figure 3.16 on the following page. The majority of the absolute error is seen to be less then 0.2mm. An interesting pattern is also seen in the error and its histogram that has a long tail to the right. This tail is attributed to the lower right corner of the error map that is seen to be dark blue in Figure 3.16. The reason is that for this small part of the scan the height error is much larger than in the remainder of the scan. It is believed that this is due to the projectors orientation in the used hardware setup. When the hardware setup was built the projector was placed in a way that allowed it to project images onto the entire area visible in the camera. However mistakenly the projector was not placed so the center of the projectors image was near the center of the cameras field of view. Thereby only a part of the outermost part of the projectors image field is visible in the camera making it nearly impossible to estimate and correct for non linear lens distortion in the projectors optics. As lens distortion is modelled polynomially a poorly estimated correction is known to go towards infinitely near the edge of the image. It is believed that this is what causes the tail in the histogram and the dark blue region in Figure 3.16.

If the center of the error map in Figure 3.16 is cropped out and analysed by itself much better results are obtained that are expected for the entire scan if the projector is moved as just described and then recalibrated. Figure 3.17 marks the cropped out region and Figure 3.18 and Table 3.1 show the corresponding results. Note that the error is normally distributed around zero with a standard deviation of just 34.7 micrometers and so 95.45% of the height errors are expected to be below 69.4 micrometers and thus well under the 100 micrometer specification stated in the problem specifications in chapter one.

# of standard deviations	Percentage	Less then [mm]
1	68.2689%	0.0347
2	95.4500%	0.0694
3	99.7300%	0.1041
4	99.9937%	0.1388
5	99.9999%	0.1735

Table 3.1: *Upper bounds on the expected height error i a 3D scan.*

[16] Both the mat and white spray is soluble in water and glass was washed clean first.

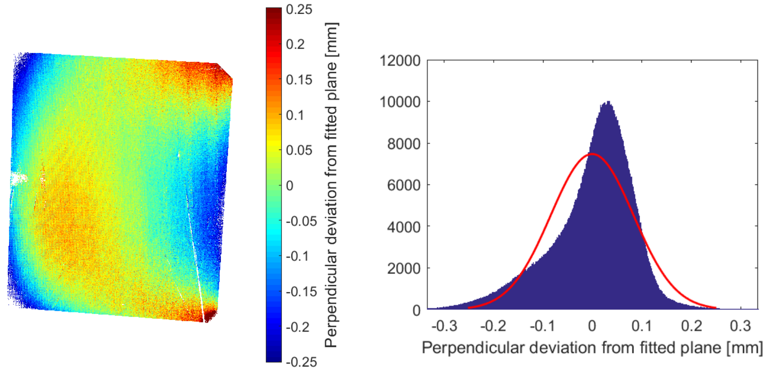


Figure 3.15: *The image to the left shows the perpendicular deviation for each pixel across the surface of the metal plate. A histogram of the deviations is seen to the right. A normal distribution has been fitted to the histogram and shown in red for comparison. Note the similarity between the circular error pattern and the shape of the region between the wrist and the palm of the human hand.*

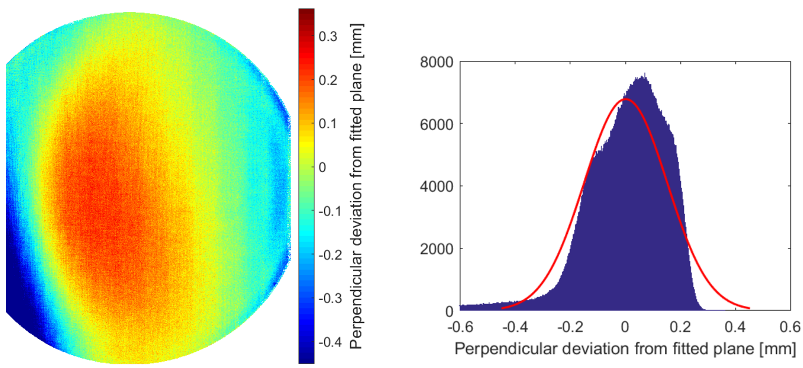


Figure 3.16: *The image to the left shows the perpendicular deviation for each pixel across the surface of the glass plate with a white piece of paper spray glued on top. A histogram of the deviations is seen to the right. A normal distribution has been fitted to the histogram and shown in red for comparison.*

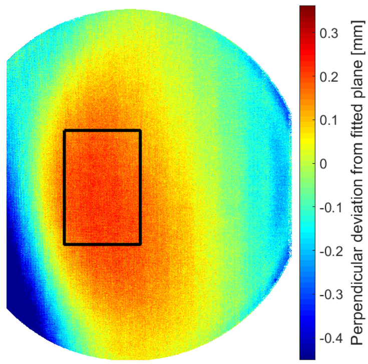


Figure 3.17: *This figure is the same as the image to the left in Figure 3.16 with the addition of a black square marking the area cropped out and used for further analysis in Figure 3.18.*

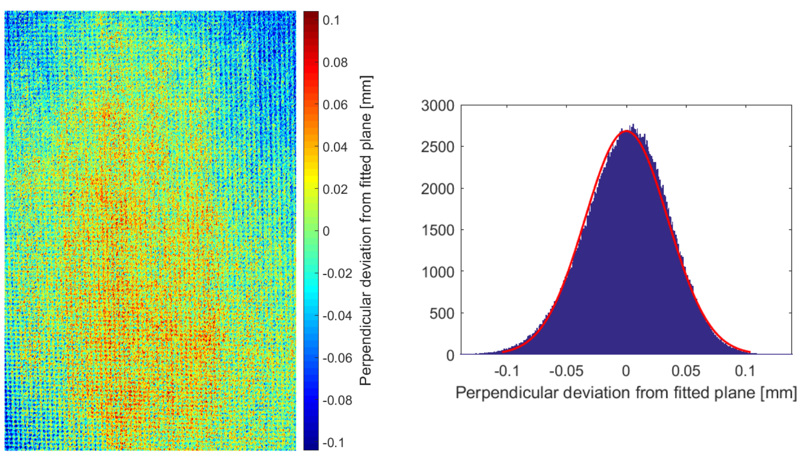


Figure 3.18: *The cropped out part of Figure 3.17. Note how the error is normally distributed around zero with a standard deviation of just 35 micrometers and so 95% of the height errors are below 70 micrometers and thus well under the 100 micrometer limit stated in the problem specifications in chapter one.*

Chapter 4

Stereo calibration

In order to perform accurate point triangulation both the camera and projector intrinsic and extrinsic parameters need to be precisely known. These parameters are estimated using a stereo calibration procedure similar to the one presented by Wilm et. al. in [18]. Camera calibration is a well studied problem however calibration of DLP systems have not been studied as extensively. Most developed methods use a camera and extract known features from projected patterns. The calibration proposed by Wilm et. al. is developed specifically for phase shifting profilometry which for this project is seen as an advantage over other methods. The presented method does not rely on an initial calibrated camera and therefore does not propagate camera calibration error into the projector calibration. By using radical interpolation to translate projected features into the projectors internal coordinate system sub-pixel accuracy can be achieved by ordinary stereo calibration.

This chapter is structured as follows. Section 4.1 describes related work on calibration of a camera-projector setup. Section 4.2 gives a concise overview of the chosen calibration method. Section 4.3 describes technical details of the chosen method. In section 4.3.1 the projected fringe patterns are described and defined. In section 4.3.2 the details of the coordinate translation from camera coordinates to projector coordinates are described. Finally section 4.4 evaluates the accuracy of the performed stereo calibration.

4.1 Related work

A large number of calibration procedures have been developed however most fail to satisfy both of the following two key requirements in this project:

1. Easy-to-perform for non-technical staff
2. Enabling very accurate 3D reconstructions

Several methods use a pre-calibrated camera for finding and extracting control points on a physical calibration target and then assigning projector correspondences to these points (see e.g. [23–28]). These methods are easy to perform however all of them suffer from a fundamental flaw in there design. They rely on an accurate camera calibration to be performed beforehand and even a small error in this calibration will be propagated to projector calibration and potentially result in large errors in the projector parameters.
Others have adopted a different approach where neither a calibrated camera nor an assumed planar printed calibration target is needed (See e.g. [29–31]). The methods are based on having the user move the projector to several locations while projecting a pattern onto a fixed wall or other structure. By using a stationary camera to observe the changes in the pattern as the projector moves around a calibration is performed of both the camera and the projector simultaneously. These methods are not usable in situations where the camera and projector are mounted on a rigid platform and these methods are not able to produce metric reconstructions as their results are only up-to-scale. Since the setup in the VideometerLab is a rigid platform and metric reconstructions are needed these calibration approaches are regarded as unusable for this project.
Iterative methods have also been proposed where a projected pattern is iteratively adjusted until they overlap with a printed pattern (see e.g. [32–35]). An uncalibrated camera is used to measure the overlap. Since both patterns must be clearly identifiable the classical black and white checkerboard pattern is replaced with a colour version of it which makes a colour camera mandatory. In practice colour calibration is also needed as the printed colour and the observed colour in the camera seldom match [36] imposing an extra undesirable step in the calibration. As the VideometerLab uses a monochromatic camera these methods are as well regarded as unusable for this project.
Another commonly used approach in projector calibration is to compute a single homography transformation between the world points on the calibration target and the projector image plane (see e.g. [25, 29, 30, 32, 34]). As homographies are linear operators they can not model non-linear distortions such as the lens distortion in the projector optics.

Finally, S. Zhang et al. [37], and others [29, 38], have developed methods using structured light patterns however they do not compute projector image points directly. Instead they create synthetic images at the projectors resolution and then feed these images to a standard calibration algorithm. As most projectors have lower resolution then the used camera this results in a high risk of discarding important information and thereby lowering the quality of the calibration.

4.2 Overview of the chosen method

To overcome the shortcomings of the above mentioned methods a three step approach is applied[1]. The used method robustly extract projector coordinates directly at camera sub-pixel resolution and no synthetic images are used.

1. Checkerbord corner coordinates are found in camera coordinates and using radial basis function (RBF) interpolation translated to projector coordinates [18]. Then the camera and the projector are calibrated individually (as descried in step 2 and 3) and simultaneously after which a combined stereo calibration is performed to further refine the found parameters. The details of this step is elaborated in section 4.3 and schematically illustrated in Figure 4.1 on the next page.

2. The intrinsics and extrinsics parameters are solved for in closed form, assuming that lens distortion is zero [40].

3. The estimated parameters are refined and the distortion coefficients are simultaneously estimated using non-linear least-squares minimization. (Using the Levenberg–Marquardt algorithm.) [40, 41].

The used method overcome all of the above mentioned shortcomings while still complying with the two key requirements of being easy-to-perform for non-technical staff and enables very accurate 3D reconstructions. However as step two and three is based on the very well known and widely used method of Zhang [40] and Heikkilä [41] the use of a perfect planar calibration target is assumed. Due to their optimization forces the control points to be in the xy-plane by hardcoding their z-coordinate to zero. To comply with this planar assumption a laser printed checkerboard spray glued on a flat 1mm thick metal plate is used as calibration target.

[1] Step two and three is done using the calibration tool in the Matlab 2015b “Computer Vision System Toolbox” which is based on the widely used Bouguet “Camera Calibration Toolbox for Matlab” [39]. Bouguets toolbox is included in OpenCV and may e.g. be used when integrating the final solution into the existing VideometerLab software.

According to the Z. Wang et al. [42] the assumption of a perfect planar calibration target can be avoided using bundle adjustment to simultaneously optimize all camera parameters *and* the possible defects of the calibration target. However since it is necessary to prioritize project time between different activities it is assessed that the benefits of their work do not justify a possibly time-consuming implementation. Their approach is therefore referred to the future work section.

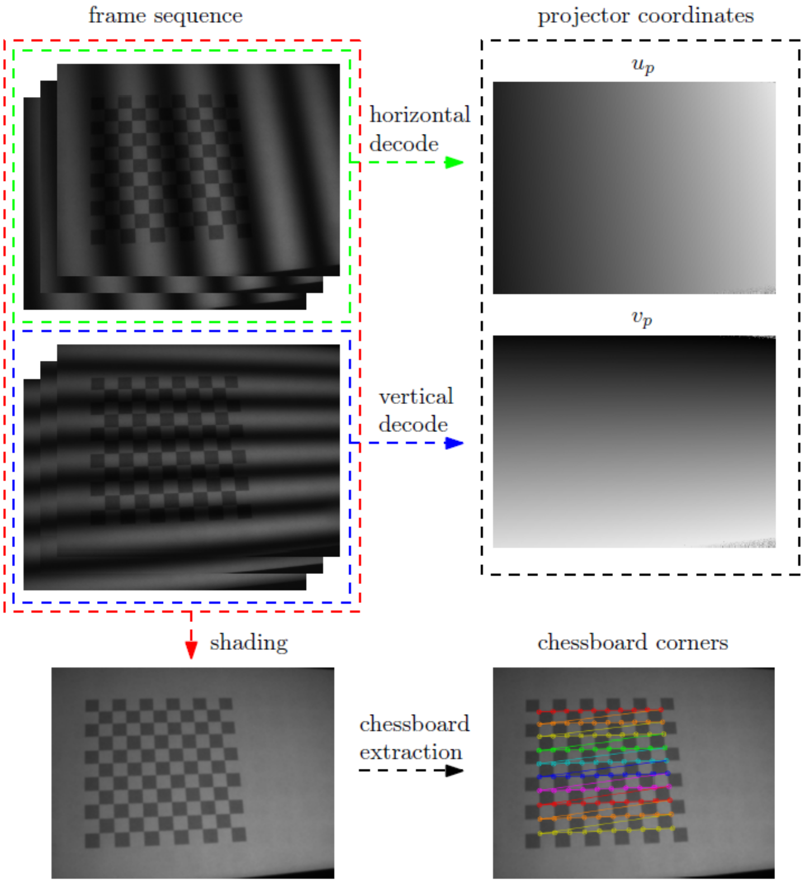


Figure 4.1: *Flow diagram of the used calibration procedure. The patterns from Figure 4.4 are projected onto the checkerboard and thus encoding the horizontal and vertical coordinate of the projector. By unwrapping (decoding) the captured images maps of projector coordinates are obtained. From these patterns a general shading or intensity image is also extracted which is used to extract the coordinates of the chessboard corners automatically. The coordinates are found in the camera reference frame and by using Gaussian radial basis functions translated into projector coordinates. The figure is from [18].*

4.3 Stereo calibration with RBF

By combining both stereo calibration and phase shifting profilometry techniques a very accurate calibration procedure is obtained. In [18] a light gray checkerboard is used as calibration target and represents a trade off between robustness in corner detection and encoding of projector coordinates. A black and white checkerboard is the ideal for robust corner detection and is therefore used in this project. In order to still get a robust projector coordinate encoding only the white parts of the checkerboard are used. In this way optimal (or nearly optimal) results can be obtained both in corner extraction and projector coordinate encoding as opposed to the compromise in [18]. The projector used in this project is a DLP LightCrafter with the technical specifications described in appendix D. The DMD[2] of the projector has a diagonal or diamond pixel array as illustrated below in Figure 4.2. The indexing of the mirrors (which can be thought of as projector "pixels") geometry has unique coordinates for every horizontal row and every other vertical column. Since a linear mapping of pixel indices to pixel positions are needed the diamond pixel array are handled by assigning unique indices to every row and every column. In this way every other coordinate is used to index a "pseudo pixel" and therefore in turn only every other coordinate are encoded. The used indexing scheme is illustrated in Figure 4.3 on the following page.

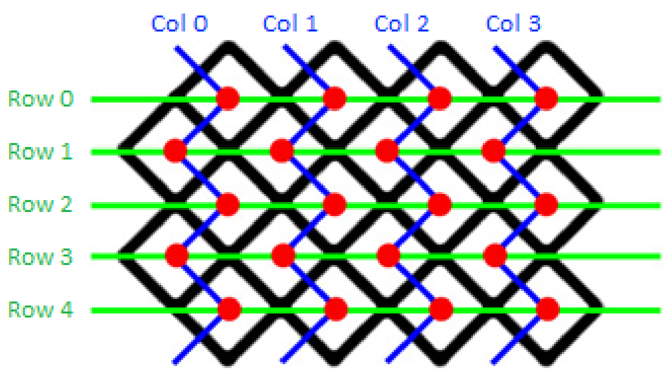


Figure 4.2: *A illustration of the projectors diamond pixel array. Figure from kguttag.com.*

[2] A projectors image is created by a matrix of microscopically small mirrors known as a digital micromirror device (DMD). Each of these mirrors represent one or more pixels in the projected image. By rapidly turning the individual mirror a light ray can be projected either through the lens or onto a heat sink. A certain image can be projected by turning the mirrors in a specific way derived from the image [8].

The employed calibration procedure overall consists of the following steps:

1. Use the (not calibrated) camera to capture images of a physical checkerboard while a series of fringe patterns are projected upon it.
2. Use a normal VideometerLab image (single band) for checkerboard corner extraction.
3. Translate the checkerboard corner coordinates from camera coordinates to projector coordinates.
4. Change the physical orientation and angle of the checkerboard.
5. Repeat step 1-4 roughly 10-20 times for different orientations and angles.
6. Translate the checkerboard corner coordinates into projector coordinates using the approach described in section 4.3.2.
7. Perform a standard stereo calibration by regarding the projector as the other camera.

Step 5 is a time consuming and manual process while all other steps are relatively fast and fully automated. The precise orientation and angle of the checkerboard need not to be known and thus this manual process may be automated by a mechanical device without considering mechanical backlash.

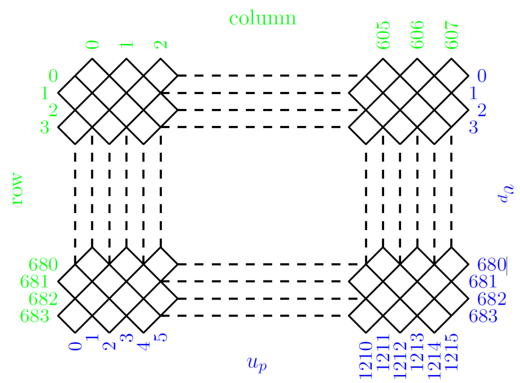


Figure 4.3: *The diamond mirror/"pixel" array of the DLP LightCrafter. The indexed screen coordinates are shown in green on top and to the left while the introduced "pseudo pixels" coordinates are shown in blue on the bottom and to the right. Note that in the introduced "pseudo pixel" coordinate system every other coordinate does not index a pixel. There is no need for a dense pixel array for any of the following steps. Figure from [18].*

4.3.1 The projected patterns

As a direct consequence of the "pseudo pixel" indexing of the projectors DMD the patterns are rendered with twice the needed columns and then every other pixel is discarded (without low pass filtering). In every even row all the even numbered columns are discarded and in every odd row all the odd numbered columns are discarded. This is illustrated in Figure 4.3. The projected patterns are rendered according to equation 3.6 on page 21 with $A^p = B^p = 0.5$ giving the equation

$$I_n^p(x^p, y^p) = \frac{1}{2} + \frac{1}{2}cos\left(2\pi fi - \frac{2\pi n}{N}\right) \tag{4.1}$$

where (x^p, y^p) is the row and column coordinate of a "pseudo pixel" on the projectors DMD, I_n^p is the light intensity of that pixel in a projector dynamic range from 0 to 1, $i = y^p \cdot 1216^{-1}$ if columns are encoded and $i = x^p \cdot 684^{-1}$ if rows are encoded[3], f is the frequency of the sine wave, n represents the phase shift index and N is the total number of phase shifted patterns. In order to reduce quantification error and noise three patterns with eight phases are used ($f = 8$) to encode the row and column coordinates respectively. By using $f \neq 1$ the computed phase needs to be unwrapped to remove the phase ambiguity. The unwrapping is done using the method of two wavelength phase unwrapping described in section 3.4.1. Three images are used as the phase cue sequence. For each of the four sequences $N = 3$ is used. Thus in total 12 patterns are projected to encode both projector row and column coordinates. The 12 patterns used are seen in Figure 4.4. From either image (a)-(f) or (g)-(l) in Figure 4.4 the B^c term from equation 3.14 on page 23 can be calculated and seen as a general intensity or shading image which can be used for checkerboard corner extraction. However in this project B^c is only used as a shadow noise filter and the diffuse homogeneous lighting of a normal VideometerLab image is used to capture a (single band[4]) image that is used for robust[5] checkerboard corner extraction.

[3] The term i is the normalized row (or column) coordinate and the numbers 1216 and 584 are thus specific to the used projector and may vary if another projector is used.

[4] Simply using whatever band that gives the most contrast between the white paper background and the black squares of the chessboard.

[5] Both the algorithm implemented in the Matlab 2015b "Computer Vision System Toolbox", the widely used Bouguet "Camera Calibration Toolbox for Matlab" [39] which in a C++ implementation is the default algorithm in OpenCV, have been experienced to fail in even detecting the checkerboard in the image if the light intensity increases slightly across the image. This is the experience made during the work with this project and by J. Wilm corresponding author of [18] during his work on his Ph.D.

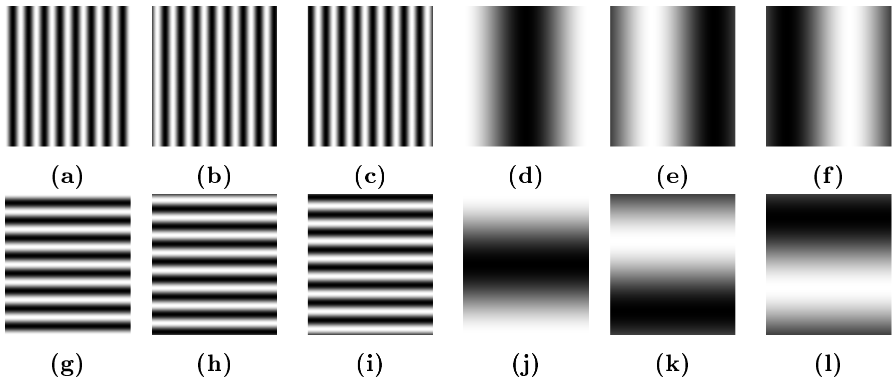


Figure 4.4: *The 12 projected patterns used to perform the stereo calibration. Pattern* ***(a)*** *to* ***(f)*** *are used to encode columns in the projector coordinate system and pattern* ***(g)*** *to* ***(l)*** *are used to encode the rows. Pattern* ***(d)*** *to* ***(f)*** *and pattern* ***(j)*** *to* ***(l)*** *are the cue sequences and are only using a single period and no unwrapping is therefore necessary for these patterns.*

4.3.2 Coordinate translation

The checkerboard corner coordinates are extracted in camera coordinates and are robustly translated to projector coordinates using interpolation of gaussian radical basis functions (RBF) [43]. The extracted corner coordinates are extracted with subpixel accuracy in the camera coordinate system and by treating the translation as an interpolation problem the computed corner coordinates in the projectors coordinate system are also found with subpixel accuracy. The underlying function is expected to be close to linear however in the presence of non-linear lens distortion a more complex function is needed to explain the data. Hence justifying a RBF model. The corner coordinates are located in regular grid sites however gridded data interpolation is not applied as data points are removed based on a threshold in B^c so the method must handle missing data. Using regular data interpolation would be much faster to compute but in this application the computational cost of using RBF interpolation is not prohibitive considering the added robustness and the possibility to allow for lens distortion.

Each corner from each sequence is translated individually and this process can be parallelized to speed up these computations. As with regular RBF interpolation for each corner point a set of equations is solved to find out how each corner point is explained as a linear combination of basis functions. As the underlying function is expected to be close to linear a linear polynomial term is added which means that the RBF will effectively describe the local deviation from a global linear trend. A regularization constant k is added to the interpolator to account for noisy data. Regularization simply relaxes the interpolation requirement. If the system is ill-conditioned regularization also improves the condition number of the system of equations needed to be solved. From a statistical point of view this type of regularization can be regarded as trading a small bias for a reduction in variance [43]. In practice the regularization is done simply by adding a constant to the diagonal of the linear system so the interpolator becomes

$$f_i = \sum_j \lambda_j \psi ||\mathbf{x}_i - \mathbf{x}_j|| + P(\mathbf{x}_i) + k\lambda_i \tag{4.2}$$

where f_i is the i'th corner point translated into projector coordinates, $\mathbf{x}_i$ is the subpixel coordinate of the i'th corner point decoded from the phase map found as in chapter 3, $\mathbf{x}_j$ is the regular position of every pixel in neighbourhood of $\mathbf{x}_i$, $P(\mathbf{x}_i)$ is the polynomial term described below, k is the regularization constant and is found using a smoother matrix as described below and λ_i is the regularization parameter. Note that all λ and the coefficients of the plane P is unknown at this point.

A gaussian radial basis function, $\psi(r) = exp(-\alpha r^2)$, is used to give the highest weight to the closest neighbours while being robust against noise in the estimated projector coordinates. An influence region of 121 neighbours (a 11×11 grid) seems reasonable and $\alpha = 1/5$ has been shown to provide good results [18].

The parameters needed to compute equation 4.2 for each corner point individually is found by forming and solving equation 4.2 in matrix form

$$\begin{bmatrix} \mathbf{\Psi} + k\mathbf{I} & \mathbf{P}^T \\ \mathbf{P} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \mathbf{c} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{0} \end{bmatrix} \tag{4.3}$$

where $\mathbf{\Psi}$ contains the basis functions and $\mathbf{P}$ is the polynomial term of the form

$$\mathbf{P} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ r_1 & r_2 & r_3 & \cdots & r_n \\ c_1 & c_2 & c_3 & \cdots & c_n \end{bmatrix} \tag{4.4}$$

containing the row and column coordinates of all pixels in the defined influence region. If no pixels are removed by the shadow noise filter then $n = 121$ however if some pixels are removed then $n < 121$. The k in equation 4.3 is a scalar parameter controlling the regularization. I.e. balancing between the exact RBF interpolator and the simple linear polynomial. As mentioned a linear trend is expected as without lens distortion there is a direct perspective relationship between the camera and the projector coordinates.

As the used RBF model can be thought of as a regularized regression model as in [18] a smoother matrix $\hat{\boldsymbol{H}}$ is defined according to

$$\hat{\boldsymbol{H}} \begin{bmatrix} \mathbf{\Psi} + k\mathbf{I} & \mathbf{P}^T \\ \mathbf{P} & \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{\Psi} & \mathbf{P}^T \end{bmatrix} \tag{4.5}$$

and the trace of $\hat{\boldsymbol{H}}$ is used as a measure of the effective degrees of freedom in the model [44]. In [18] a value of k is chosen such that the effective degrees of freedom are 10 with the argument that 10 is "a little more than the 3 defining a plane, but much less then the 100 data points underlying each model." In this project a regularization constant of $k = 0.525$, corresponding to ≈ 12.6 effective degrees of freedom, is used as it was experimentally found to minimize the mean reprojection error. Figure 4.5 shows the experimental results.

By solving the system of equations in 4.3 a set of coefficients $\boldsymbol{\lambda}$ and $\mathbf{c}$ are found. Knowing the coefficients $\boldsymbol{\lambda}$ and $\mathbf{c}$ and inserting these into equation 4.2 the coordinate of one checkerboard corner can be translated from camera coordinates to projector coordinates while maintaining subpixel accuracy in both coordinate systems. By repeating the described coordinate translation for each individual corner point for each angle and orientation of the checkerboard a combined set of corresponding camera- and projector coordinates can be found and used to perform a standard stereo calibration.

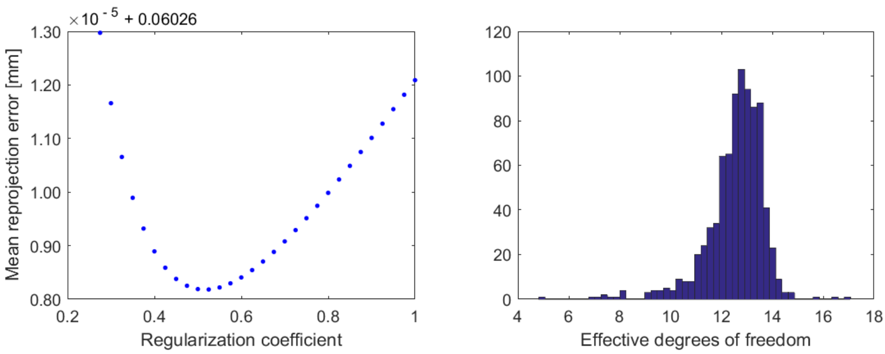


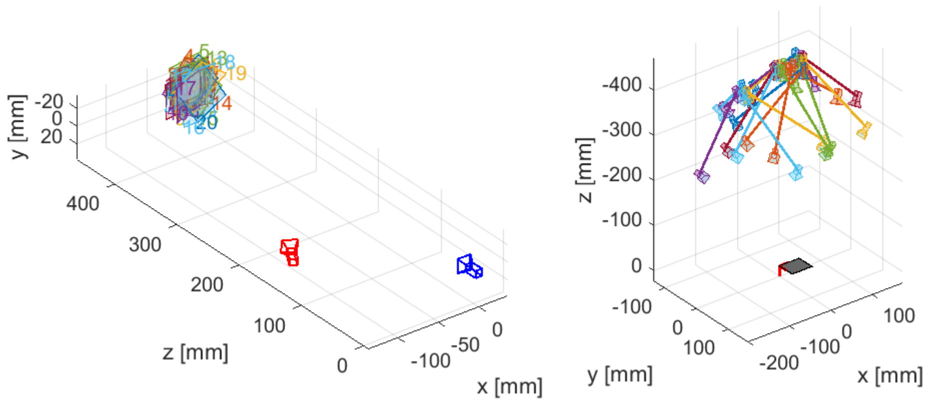
Figure 4.5: *Experimental results of varying the regularization constant and measuring the corresponding effective degrees of freedom individually for each corner in each of 20 individual angles and orientations of the checkerboard. Note the scale on the y-axis is 10^{-5}mm so the exact choice of regularization constant has little influence on the mean reprojection error. However the difference between not using regularization and using the found regularization constant of k=0.525 is a 10.2% drop in the mean reprojection error. This drop further justifies the use of regularization.*

4.4 Calibration accuracy

During the stereo calibration both the camera and projector intrinsic and extrinsic parameters are estimated. A number of ways exist to evaluate the accuracy of the estimated parameters and in this section the following are examined

- The relative positions of the camera, the projector and the different positions of the calibration pattern have been plotted and analysed.
- The reprojection errors been computed and analysed.
- Estimation uncertainties have been computed for all parameters.

Extrinsic parameters visualization
By visualizing and analysing the extrinsic parameters it can quickly be determined if any obvious errors were made. For example if one of the calibration patterns appear behind the camera or if a pattern is too far away from or too close to the camera, etcetera. Figure 4.6 shows the relative positions of the camera, the projector, and the different positions of the calibration pattern. It is shown both in the camera's coordinate system and relative to the calibration pattern's coordinate system. No obvious errors are discovered by analysing the figures.



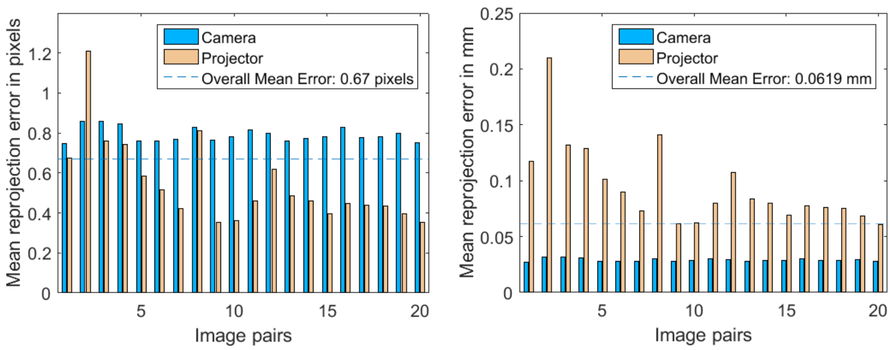
(a) *Camera centric.* **(b)** *Pattern centric.*

Figure 4.6: *Visualization of the extrinsic parameters. In the camera centric view the camera is shown in blue and the projector is shown in red.*

Reprojection Errors

The reprojection errors provide a qualitative measure of the calibration accuracy. The reprojection error is defined as the difference between where a corner point is detected in an image and where the corresponding world point ends up after being projected into the same image using the calibrated parameters. Figure 4.7 visualize the reprojection errors of each of the 20 image pairs used in the calibration as well as the overall mean reprojection error. The reprojection errors are shown in units of both pixels and millimetres. Note that the reprojection errors are all but one less then one pixel confirming that sub-pixel accuracy is obtained. Also note that the physical size of a pixel is different for the camera and the projector and thus the reprojection errors measured in pixels can not be directly compared between the camera and the projector. The size of a camera pixel is 0.0366mm and the size of a projector pixel is estimated to be 0.1737mm. This difference in size is the reason that the camera seems to have the largest error when looking at Figure 4.7a but seems like having the smallest error when looking at Figure 4.7b.

The figures do not give rise to suspecting any calibration errors and confirm that a very accurate calibration has been obtained with sub-pixel accuracy.



(a) *Measured in pixels.* **(b)** *Measured in millimetres.*

Figure 4.7: *Visualization of the reprojection errors of each of the 20 image pairs used in the calibration.*

Estimation uncertainties

The uncertainty of the estimated intrinsic parameters for the camera and the projector are listed below together with their internal relative position. A full list of the computed uncertainties for all 20 translation vectors and 20 rotation vectors are given in appendix E. The errors listed are standard errors and may be used to compute confidence intervals for each parameter. For example there is a 95% probability that the true value for a given parameter is within 1.96 times the listed standard error for any given parameter. It is noted that all standard errors are relatively small when compared to the estimated value which indicate a successful and very accurate calibration.

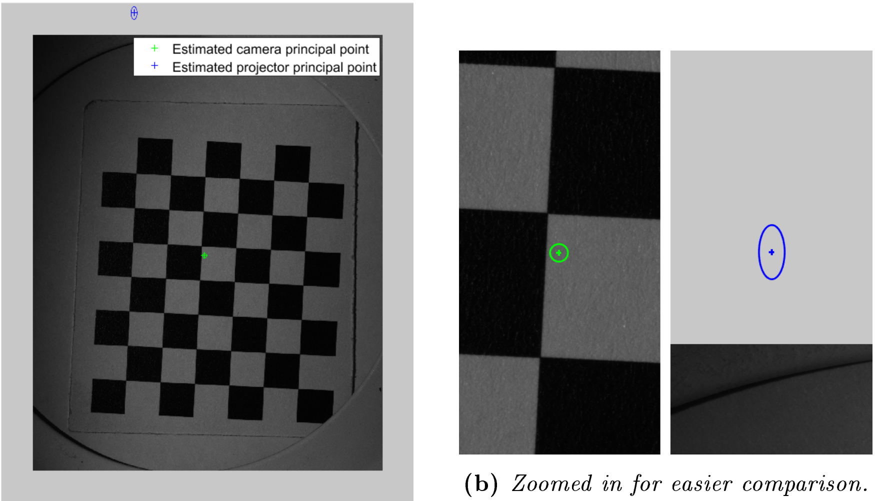
The principal point is the optical center of the camera and the point where the optical axis intersects the image plane. The standard error of the estimated principal point is visualized in Figure 4.8 on the next page together with an ellipse whose radii are equal to 1.96 times the estimation errors. The actual principal point is therefore contained in this ellipse with 95% probability.

```
Camera intrinsics
-----------------
Focal length (pixels):   [12472.2115 +/- 49.7326   12472.2797 +/- 49.7385 ]
Principal point (pixels):[ 1076.7276 +/- 6.7271     1387.0016 +/- 6.6363  ]
Radial distortion:       [    1.0075 +/- 0.0744      -76.3058 +/- 9.4058  ]

Projector intrinsics
--------------------
Focal length (pixels):   [ 2059.8182 +/- 16.0432    2047.9162 +/- 16.4507 ]
Principal point (pixels):[  634.2298 +/- 9.7288     -135.6540 +/- 20.5165 ]
Radial distortion:       [    0.0211 +/- 0.0299       -0.2951 +/- 0.1222  ]

Position and orientation of the camera relative to the projector
----------------------------------------------------------------
Rotation of the projector:
        [   0.8898 +/- 0.0099     0.0323 +/- 0.0045    -0.4552 +/- 0.0019 ]

Translation of projector (mm):
        [ 171.7069 +/- 0.7663   -11.7599 +/- 0.8716   -49.9412 +/- 3.0683 ]
```



(a) *Full view.*

(b) *Zoomed in for easier comparison.*

Figure 4.8: *Visualization of the principal point error for both the camera and the projector. As the projectors principal point is outside the cameras field of view its position is not in the camera image. A gray frame has therefore been added to the picture and the projectors principal point plotted onto this frame. As the uncertainty in the principal points is very small it is hard to see the uncertainty ellipse and a zoomed in view is therefore also shown.*

Recall from section 3.5 that the pinhole camera model is

$$\mathbf{q_i} = \mathbf{A}[\mathbf{R}\ \ \mathbf{t}]Q_i = \mathbf{P}Q_i\ , \quad \mathbf{P} = \mathbf{A}[\mathbf{R}\ \ \mathbf{t}] \tag{4.6}$$

where Q_i is a 3D point with the projection $\mathbf{q_i}$ in a camera defined by $\mathbf{P}$.
Now let $\mathbf{q_c}$ and $\mathbf{q_p}$ be the known projections of the 3D point Q in the camera and projector respectively and let $\mathbf{P_c}$ and $\mathbf{P_p}$ be the camera matrices of the projector and the camera respectively. Then the calibrated parameters of $\mathbf{P_c}$ and $\mathbf{P_p}$ can be explicitly written as

$$\mathbf{P_c} = \begin{bmatrix} 12472 & 0 & 1077 \\ 0 & 12472 & 1387 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{4.7}$$

$$\mathbf{P_p} = \begin{bmatrix} 2060 & 0 & 634 \\ 0 & 2048 & -136 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.90 & 0.03 & -0.46 & 172 \\ 0.06 & 0.98 & 0.19 & -12 \\ 0.45 & -0.20 & 0.87 & -50 \end{bmatrix} \tag{4.8}$$

Note that the projectors principle point is outside of the DMD chip. This is due to the fact that the projector projects images along an off-axis direction[6] [8].

[6] The projector has a 100% vertical offset as listed in appendix D meaning that it projects an image whose bottom edge is in line with the center of the lens and extends upwards.

Chapter 5

Applications

This chapter will showcase two specific applications that benefit from being able to combine 3D and multispectral imaging in the VideometerLab.

When used on a multispectral image canonical discriminant analysis (see [1] or [45]) have vast discriminatory power that are first utilized to make a very accurate, fast and easy segmentation into background and foreground objects. Then using 3D differential geometry the geometric surface characteristics of each foreground object is analysed allowing for effective classification into dorsal or ventral grains. Analysing the surface characteristics also allows for improved segmentation of granular products such as rice, grains and seeds.

Section 5.1 elaborates on the classification of dorsal and ventral grains and section 5.2 examines how algorithms for segmentation of granular products can benefit from the integrated 3D information.

There are numerous additional possible applications of combining 3D, multispectral, and fluorescence imaging and several of Videometers customers have confidential applications of integrated 3D information. The two applications showed in this chapter should therefore only be seen as a few examples.

5.1 Classification of dorsal and ventral grains

When working scientifically with grains one use the terminology of the “dorsal” and “ventral” side of the grain. The term dorsal originates from the Latin “dorsum” that means “back” and dorsal therefore refers to the back of the grain. The term ventral originates from the Latin “venter” that means “belly” and ventral therefore refers to the front of the grain. This work has been done specifically on wheat grains. The dorsal side of the wheat grain is smoothly rounded and the ventral side has a deep crease. Figure 5.1 shows eight examples of wheat grains where the first four from the left are dorsal and the last four are ventral. Note how the ventral wheat grains all have a deep crease running from top to bottom and all the dorsal wheat grains have a smooth appearance.

As some fungal infections are only visible on the dorsal side of the grain an automatic classification method is desired to ease detection of such fungal infections. Such a method has been developed by Videometer using only 2D images. This section explains and reviews a novel approach for dorsal/ventral classification combining both multispectral imaging and 3D information. It will also be shown that superior results can be obtained using this novel method.

Figure 5.1: *Eight examples of wheat grains. The first four from the left are dorsal and the last four are ventral. The wheat grains are cropped out of a multispectral VideometerLab image and segmented from the background using canonical discriminant analysis (as described in e.g. [45] or [1]). The wheat grains shown are magnified for clarification and the average size of the wheat grains used in this work is ≈6.6mm long and ≈3.5mm wide.*

5.1.1 Automatic classification

A method for automatic classification between dorsal and ventral wheat grains have been developed to illustrate one of the ways that 3D information can assist a VideometerLab analysis and even outperform it. Figure 5.3 on page 66 shows six examples of 3D scanned wheat grains and Figure 5.4 shows their corresponding height maps. The classification is made by analysing the local shape of the surface of the grains and thus tools from differential geometry where needed. For theory on differential geometry the reader is referred to e.g. [46] or for information on computational differential geometry

Bærentzens et al. book [43]. Classical curvature measures like the Gaussian curvature or the mean curvature are informative of the curvature of a specific point on a surface but are however not very informative about the local shape. The principal curvatures of a surface are much more informative about local shape and can be combined into a single number called the *shape index*. This shape index is both a good shape indicator and is size invariant i.e. invariant to the amount of curvature but only affected by the type of curvature [47]. The shape index is a number in the range $[-1; 1]$ that describes the local shape as one of the categories listed in Table 5.1 and illustrated in Figure 5.2 on the following page.

The shape index at any given point on a surface is defined as

$$s = \frac{2}{\pi} arctan\left(\frac{\kappa_2 + \kappa_1}{\kappa_2 - \kappa_1}\right), \quad \kappa_1 \geq \kappa_2 \tag{5.1}$$

where κ_1 and κ_2 are the principal curvatures of that point of the surface. The principal curvatures are the eigenvalues of the Hessian matrix and thus found by simply solving $|H - \kappa I| = 0$ for κ where I is the identity matrix and H is the Hessian matrix

$$H(x) = \begin{bmatrix} L_{cc}(x) & L_{cr}(x) \\ L_{cr}(x) & L_{rr}(x) \end{bmatrix} \tag{5.2}$$

where $L_{cc}(x)$ is the images second partial derivative along the columns and where $L_{rr}(x)$ is the images second partial derivative along the rows and $L_{cr}(x)$ is the images mixed partial derivative in both directions. This means that the shape index can be written

$$s = \frac{2}{\pi} arctan\left(\frac{-L_{cc}(x) - L_{rr}(x)}{\sqrt{(L_{cc}(x) - L_{rr}(x))^2 + 4L_{cr}(x)^2}}\right) \tag{5.3}$$

and computed for the entire image all at once.

Surface type	Shape index range
Spherical cup	$s \in [-1,\ -7/8)$
Trough	$s \in [-7/8,\ -5/8)$
Rut	$s \in [-5/8,\ -3/8)$
Saddle rut	$s \in [-3/8,\ -1/8)$
Saddle point	$s \in [-1/8,\ +1/8)$
Saddle ridge	$s \in [+1/8,\ +3/8)$
Ridge	$s \in [+3/8,\ +5/8)$
Dome	$s \in [+5/8,\ +7/8)$
Spherical dome	$s \in [+7/8,\ +1]$

Table 5.1: *The shape index ranges for different surface types.*

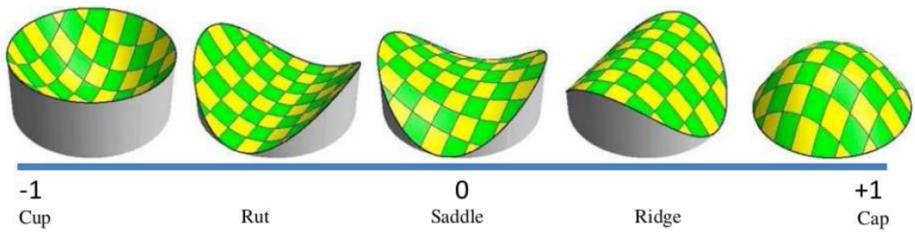


Figure 5.2: *The shape index ranges for different surface types [48].*

By using the idea of scale space representations presented by T. Lindeberg in [49] and [50] a shape index can be represented either in a multi-scale fashion or in just one scale different from the original image. This is used to represent and measure the shape of the grains at a scale adapted to the grains physical size instead of on a pixel by pixel level. The actual scale used is determined by measuring the median of the minor axis of the first e.g. 20 grains. By “minor axis” is meant the minor axis of the ellipse that has the same normalized second central moments as the grain itself. The scale is then set to 15^{-1} times the found median. This factor of 15^{-1} is found experimentally to give the best results. As the first 20 grains had a median minor axis length of ≈ 87 pixels a scale of ≈ 5.8 was used in this project[1]. By setting the scale in this way the geometry is made independent of the physical size of a pixel or of a specific number of pixels and thus the camera can be replaced or the images scaled down without changing the algorithm. The ability to seamlessly scale down the images is considered an advantage that allows for faster computations and faster development of algorithms that rely on a prior classification between dorsal and ventral grains.

The shape index of the six wheat grains in Figure 5.3 is seen in Figure 5.5 on page 67. If a grain is ventral then a large part of the grain will have a rutted surface. Thus by thresholding the shape index at $-2/8$ a binary map is obtained showing rutted areas on the surface of the grain. The result after thresholding is seen in Figure 5.6 on page 68 and by analysing the shape of any present rutted areas a classification can be made on whether the grain is dorsal or ventral. If no rutted areas appear then the grain is dorsal. If a rutted area appears and its major axis is longer then one third of the major axis of the grain itself then the grain is classified as ventral. By “major axis” is meant the major axis of the ellipse that has the same normalized second central moments as the rutted area. If more then one rutted area are present on the grain then the one with the biggest area is used and the rest are ignored. Figure 5.6 shows the thresholded shape index images in gray with an ellipse fitted in red to the grain itself and to the largest rutted area. The blue line is the major axis in the ellipse of the largest rutted area and the blue dots are the endpoints of grains own major axis.

[1] Note that in the computation of the shape index the scale is rounded to the nearest integer towards infinity and decimals of the scale value do therefore not matter.

A total of 114 wheat grains were scanned and manually annotated by an expert as either dorsal or ventral. The above described method is developed on the first 17 grains, evaluated and slightly modified on the next 41 grains and finally tested on the remaining 56 grains. As the method was not altered between the second and third run overfitting is ruled out as roughly half the grains were used as an independent test set. The developed method is able to correctly classify every single grain when comparing with the expert annotations. Giving the classifier a splendid accuracy of 100%. This is to be compared with Videometers existing 2D method capable of providing an accuracy of 94.48% based on 798 grains.

As dorsal/ventral classification is a binary classification problem it can be modelled as a binomial experiment and it can be shown that the classifications empirical accuracy also has a binomial distribution. A confidence interval for the *true* accuracy can thus be written using the binomial distribution, but is however often approximated by the normal distribution when the number of samples (grains in this case) is sufficiently large. Based upon the normal distribution the confidence interval for the true accuracy can be expressed as [51]

$$\frac{2Na + Z_{\alpha/2}^2 \pm Z_{\alpha/2}\sqrt{Z_{\alpha/2}^2 + 4Na - 4Na^2}}{2\left(N + Z_{\alpha/2}^2\right)} \tag{5.4}$$

where N is the number of samples used (grains in this case), a is the empirical accuracy, and $Z_{\alpha/2}$ is the standard normal inverse CDF at the probability 0.5α. Table 5.2 lists the confidence intervals for the true accuracy for both the 2D and 3D approach. A value of $\alpha = 0.05$ was used corresponding to using a confidence level of 95%. It is noted that the confidence intervals do not overlap and it is therefore concluded that the difference in empirical accuracy is statistically significant at a 95% confidence level. In other words that the 3D algorithm *is* performing better then the 2D alternative.

Algorithm	Number of grains	Empirical accuracy	Confidence interval of true accuracy
2D	798	94.48%	92.67% – 95.86%
3D	114	100%	96.74% – 100.00%

Table 5.2: *Confidence intervals of true accuracy for the 2D and 3D approach.*

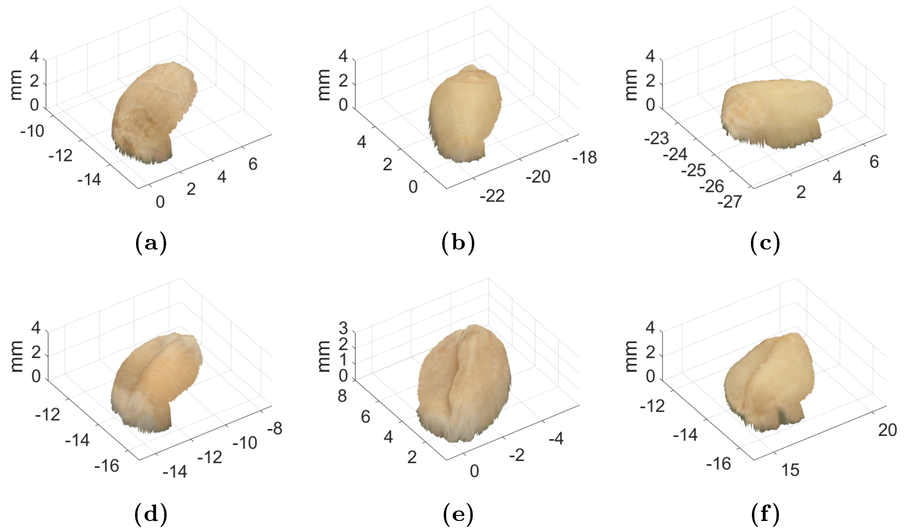


Figure 5.3: *Examples of 3D scanned wheat grains. Grain* ***(a)-(c)*** *are dorsal and grain* ***(d)-(f)*** *are ventral. The grains are coloured by texture mapping from a true colour image generated by combining the red, green and blue images from a standard VideometerLab image.*

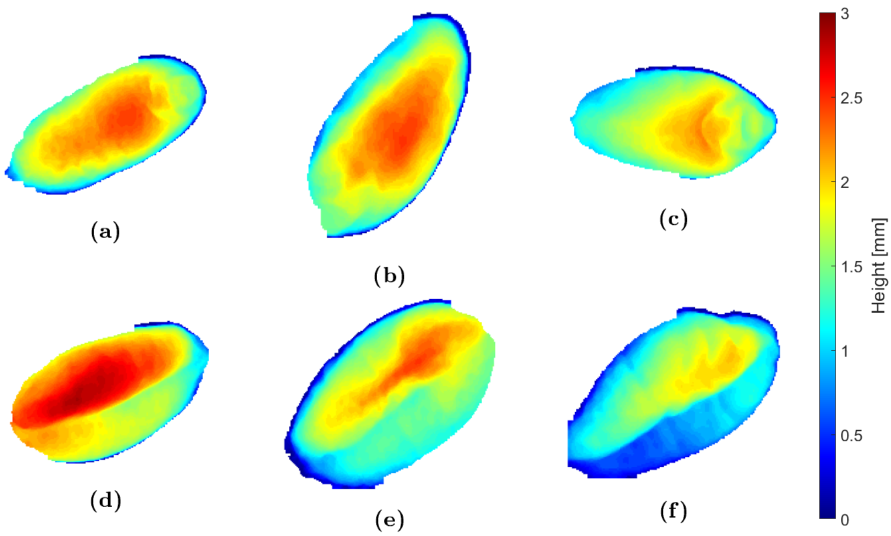


Figure 5.4: *The height maps of the six grains seen in Figure 5.3.*

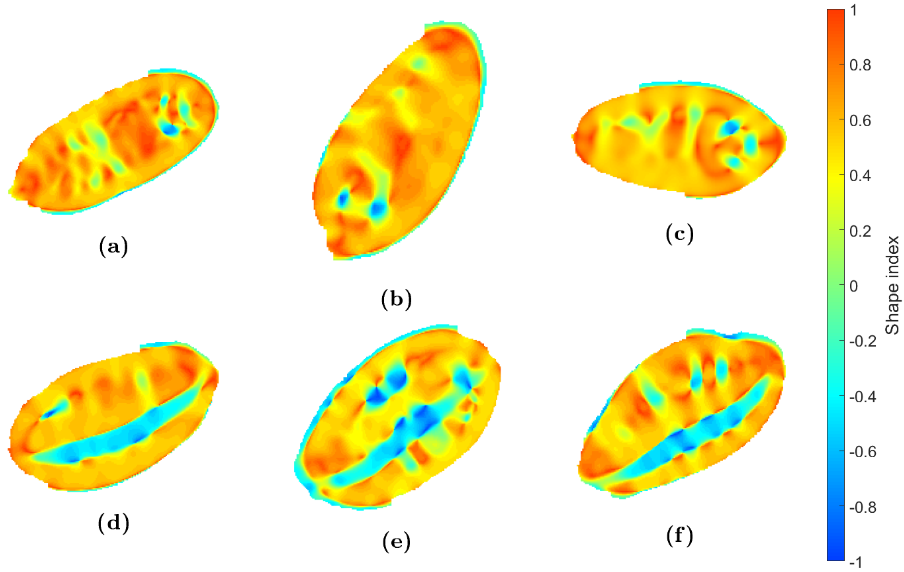


Figure 5.5: *The shape index of the six grains seen in Figure 5.3.*

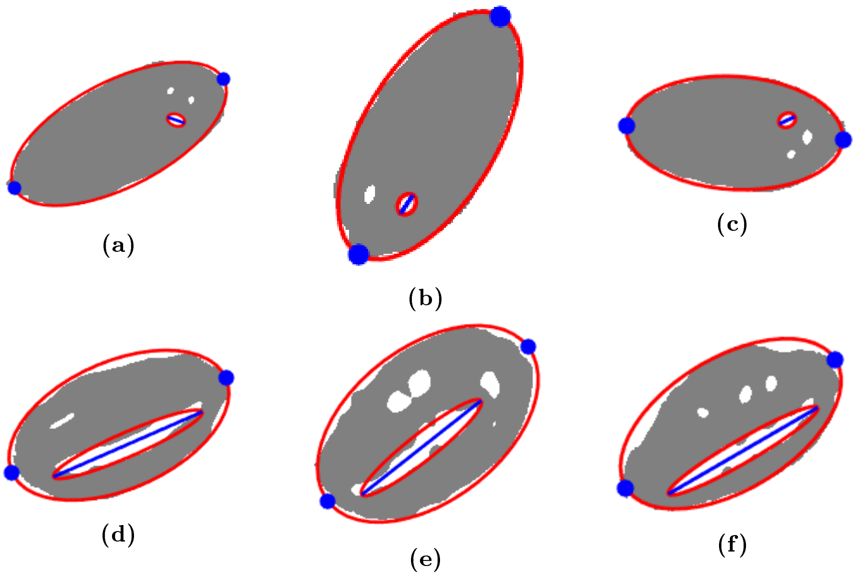


Figure 5.6: *The thresholded shape index maps of the six grains seen in Figure 5.3. The gray areas are the areas of the index maps that are greater than the threshold and the white holes are the areas that are smaller then the threshold. An ellipse has been fitted in red to the grain itself and to the largest white hole corresponding to the largest rutted area. The blue line is the major axis in the ellipse of the largest rutted area and the blue dots are the endpoints of the major axis of the grain itself. It is noted that for the ventral grains **(d)-(f)** the blue line is longer then a third of the distance between the blue dots. This corresponds to the major axis of the largest rutted area being longer then a third of the major axis of the grain itself consequently classifying the grain as a ventral.*

5.2 Segmentation of granular products

Computational image analysis of touching and possibly overlapping objects, like the grains in Figure 5.7, almost always starts with finding the relevant objects. That is figuring out which pixels in the image that make up each object and thereby separating the objects from the background as well as from each other. This process is referred to as segmentation. In the widely used textbook “Digital Image Processing” [52] it is written that “segmentation of nontrivial images is one of the most difficult tasks in image processing. Segmentation accuracy determines the success or failure of computerized analysis procedures.” A reliable algorithm for segmentation of granular products like rice, grains or seeds is therefore essential for many of Videometers projects within the cereals sector. Some of Videometers current algorithms for segmentation of granular products rely on a watershed transformation on a 2D image. It is however believed that the accuracy of the existing algorithms can be improved when combining the existing 2D multispectral image with 3D height data.

This section first introduces the basic watershed segmentation followed by the introduction and motivation of the widely used extension called marker-controlled watershed segmentation in section 5.2.2. Examples of its use on both 2D multispectral images and 3D height data is shown and evaluated. In section 5.2.3 a novel measure of segmentation error is developed and a set of optimal parameters found that minimizes this measure of segmentation error. Then in section 5.2.4 the shortcomings of the 2D and 3D approach is analysed in order to finally in section 5.2.5 present a combined algorithm for segmentation of granular products like rice, grains or seeds utilizing the combined multispectral image and the 3D height data.

The following work on segmentation algorithms was done on touching barley grains as these grains are the most difficult to segment using the traditional methods and thereby ensuring that the developed method is challenged.

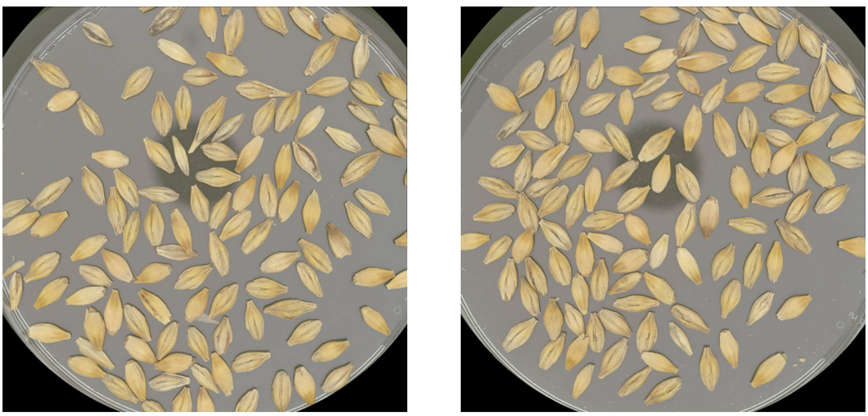


Figure 5.7: *Two examples of images of touching barley grains.*

5.2.1 Watershed segmentation

Intuitively a watershed segmentation can be explained using a metaphor based on the behaviour of rain water in a hilly landscape like the one illustrated in Figure 5.8. When rain water hits a hillside in the landscape the rain drops will start to run downhill ending up in the bottom of valleys. So each valley will have a region of hillsides from which all rainwater drains into the valley. That is each valley is associated with a catchment basin and each point in the hilly landscape belongs to either exactly one unique basin or is a hilltop not belonging to any catchment basin. The watershed lines between individual catchment basins are referred to as ridge lines. In the 2D case using images it is these ridge lines that separate the individual objects.

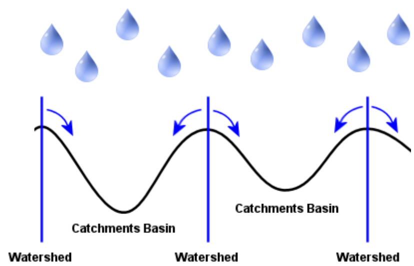


Figure 5.8: *Illustration of a hilly landscape with marked catchment basins and watershed points. The figure is from imagemet.com.*

As indicated by the above example one needs to think of an image as a surface in order to understand what the watershed transform does. Consider e.g. the image to the left in Figure 5.9 showing two synthetically generated dark blobs. If one thinks of the bright areas as being high ground and the dark areas as being low ground then the image becomes the surface to the right in 5.9. The two catchment basins and watershed line between them is marked on the figure and it is noted that this watershed line separates the two dark blobs.

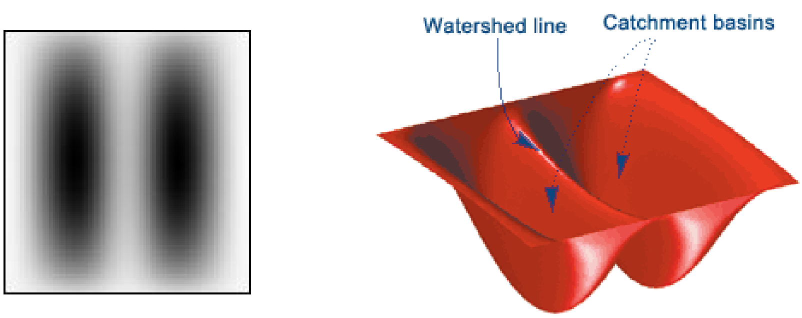
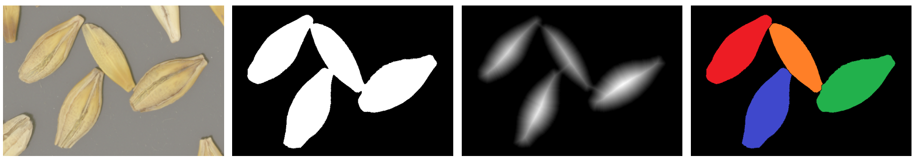


Figure 5.9: *A visualization of the watershed principle for a 2D image [53].*

It is important to note the difference between the watershed *segmentation* and the watershed *transform*. The term "watershed segmentation" is a common name for a family of algorithms that are all based on the watershed transform. The watershed transform is not a full segmentation method on its own except in very special cases. The main idea in using the watershed transform for segmentation is to first modify your image into an other image where the catchment basins are the objects you want to segment.

If one has an image of e.g. grains and the image is segmented into foreground and background then a distance transform is typically used to generate an image where each catchment basin represents one of the grains needed to be segmented. For images of roughly circular touching blobs, like e.g. grains, a standard euclidean distance map provides good results. The different steps of the segmentation process is visualized in Figure 5.10. For a multispectral image the segmentation into background and foreground objects can be done very accurately using e.g. canonical discriminant analysis as described in [1] and [45] it is however beyond the scope of this thesis to cover the details hereof.



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

Figure 5.10: ***(a)*** *Touching barley grains.* ***(b)*** *A segmentation into background and foreground objects have been performed. The five only partially visible grains have been ignored. Note that the four grains are touching.* ***(c)*** *An euclidean distance map of the image in (b).* ***(d)*** *A watershed segmentation have been performed on (c) and the four barley grains have been separated and colour coded for clarity.*

As the watershed algorithm is implicitly designed to find catchment basins and ridge lines in a 3D hilly landscape it is natural to examine how well the algorithm performs on a height map derived from real 3D data. Watershed segmentation is however known for its tendency to perform "oversegmentation" as every single local minimum, no matter how small, becomes a catchment basin. To overcome this challenge one modifies the distance transformed image in such a way that most of the unwanted local minima are removed. This technique is called minima imposition and leads to what is called marker-controlled watershed segmentation. To understand and further motivate this technique consider the two examples in Figure 5.11 and 5.12 on the next page showing the watershed segmentation used on a 1D curve as well as on the height data measured for the four grains from Figure 5.10.

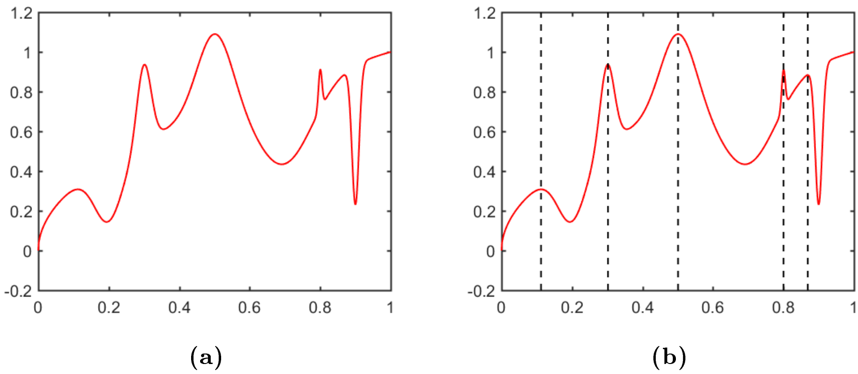


Figure 5.11: ***(a)*** *A 1D function representing a profile of a 2D image.* ***(b)*** *A watershed segmentation of the line in (a). Note that every single local minimum has become a catchment basin and every local maximum has therefore become part of a ridge line. The dotted black vertical lines mark the ridge points separating the catchment basins.*

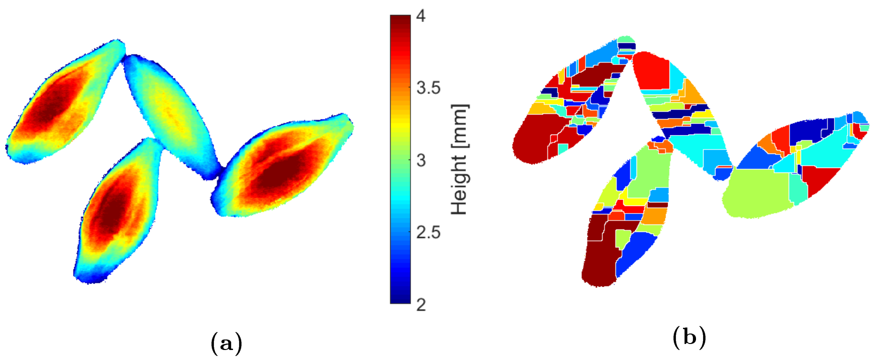


Figure 5.12: ***(a)*** *The measured height data of the four barley grains shown in Figure 5.10a.* ***(b)*** *A watershed segmentation computed on the basis of the height data in (a). The natural textured surface of the grains cause massive oversegmentation as every single local minimum, no matter how small, become a catchment basin.*

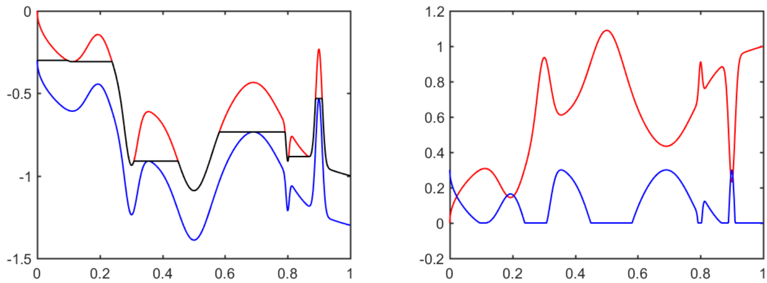
5.2.2 Marker-controlled watershed segmentation

As seen in Figure 5.11 and 5.12 the use of a watershed segmentation without the use of markers causes evident oversegmentation. To understand how to remove the unwanted local minima one has to understand the concept of marker-controlled watershed segmentation where the position of the initial lakes are determined by a marker image instead of the local minima. The unwanted local minima can be removed from the image/curve donated y as follows

1. Compute a *morphological reconstruction* of the marker image $-y-h$ under the mask $-y$ where h is a constant height.
2. Compute a *h-dome transform* by subtracting the above found morphological reconstruction from $-y$.
3. Find the marker by simply thresholding the h-domes.
4. Modify y so it only has local minima wherever the marker is nonzero.

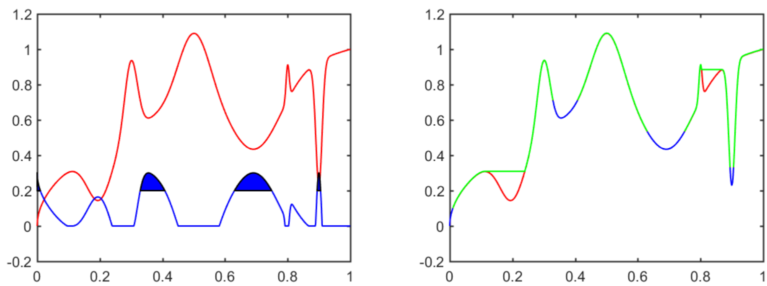
To understand what this means in practice let's use the curve from Figure 5.11a as an example and donate its y-values for y. First a morphological reconstruction is computed as illustrated in Figure 5.13a. It may help to conceptually think of a morphological reconstruction as a series of repeated dilations of the marker image until the contour of the marker image fits under the mask image. Each successive dilation is constrained to lie beneath the mask. This process "spread out" the peaks of the marker image as indicated by the vertical parts of the black line in Figure 5.13a. The final dilation is the reconstructed image and is illustrated in Figure 5.13a where the red curve is mask, $-y$, the blue curve is the marker, $-y-h$, and the black curve is the morphological reconstruction. Next the h-domes are computed by simply subtracting the morphological reconstruction from $-y$. This is illustrated in Figure 5.13b where the red curve is y and the blue curve is the corresponding h-domes for a value of $h = 0.3$. The marker is now found by simply thresholding the h-domes. The filled in blue parts of figure 5.14a show the parts of the h-domes higher then a threshold of 0.2 and these parts make out the marker. The green curve in Figure 5.14b shows the original red curve (from Figure 5.11a) after it has been modified to only have local minima wherever the marker is nonzero. That is the green curve does only have minima at the locations where the marker curve is above 0.2 in Figure 5.14a. Now finally using watershed transform on the green curve in Figure 5.14b (the blue sections are considered part of the green curve) will provide the desired result of avoiding oversegmentation as seen in Figure 5.15a. In the hilly landscape now only valleys deeper then 0.2 compared to their surroundings are considered individual catchment basins. This controls the segmentations sensitivity and this is precisely the core of marker-controlled watershed segmentation allowing for more accurate and useful segmentations.

Using the technique of marker-controlled watershed segmentation the extreme over-segmentation seen in Figure 5.12b of the four barley grains from Figure 5.10a can be avoided. The segmentation is seen in Figure 5.15b on the facing page and was obtained using the h-domes seen in Figure 5.16a and the marker seen in Figure 5.16b. The only two parameters in a marker-controlled watershed segmentation are the height of h and the threshold value used to threshold the h-domes to find the marker. The segmentation seen in Figure 5.15b was made with $h = 1.69$ and a threshold value of 0.45. Section 5.2.3 describes how an optimal set of these two parameters can be found and compares the optimal 3D based segmentation to the existing 2D segmentation.



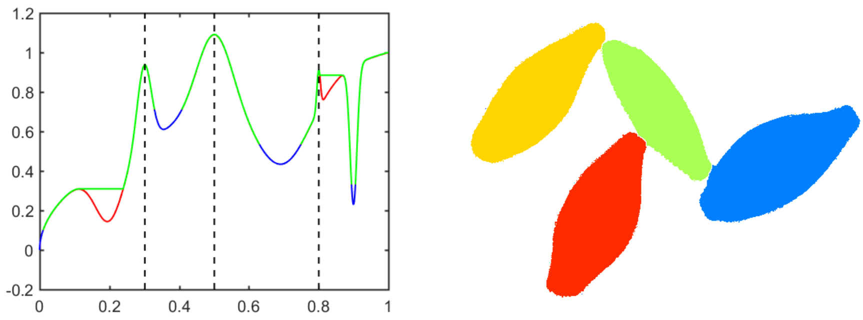
(a) (b)

Figure 5.13: *(**a**) The black line is a morphological reconstruction made using the red curve as mask and the blue line as marker. (**b**) The blue curve is the red curves h-domes for a value of h = 0.3. The h-domes are computed simply by subtracting the morphological reconstruction (black line in (a)) from $-y$ (red curve in (a)).*



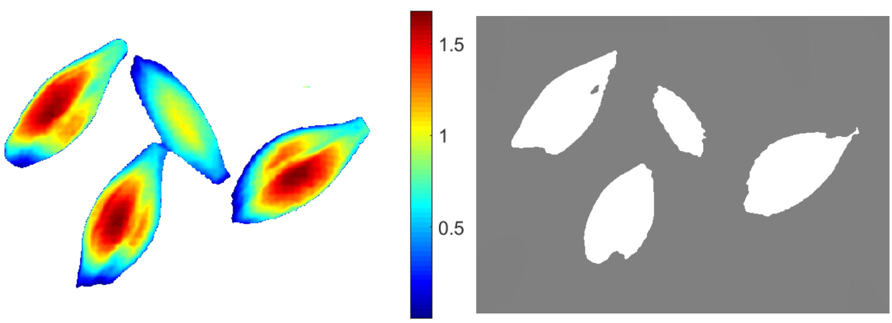
(a) (b)

Figure 5.14: *(**a**) The filled in blue parts show where the h-domes are higher then a threshold of 0.2 and it is these parts that make out the marker. (**b**) The green curve shows the original red curve (from Figure 5.11a) after it has been modified to only have local minima wherever the marker is nonzero that is only have minima at the locations where the h-domes are above 0.2.*



(a) (b)

Figure 5.15: ***(a)*** *Using the watershed transform on the green curve (the blue sections are considered part of the green curve) will avoid oversegmentation as only valleys deeper then 0.2 compared to their surrounding are still present in the landscape. The dashed black vertical lines mark the separation between catchment basins.* ***(b)*** *A marker-controlled watershed segmentation of the four barley grains from Figure 5.10a based on their 3D measured height map from Figure 5.12a. The h-domes used are seen in Figure 5.16a and the marker is seen in Figure 5.16b.*



(a) (b)

Figure 5.16: *The h-domes* ***(a)*** *and the marker* ***(b)*** *used in the segmentation of the four barley grains from Figure 5.10. The gray parts of (b) are zeros and the white holes are the actual markers.*

5.2.3 Optimal parameters

There are as mentioned only two parameters in a marker-controlled watershed segmentation. The value of h used for computing the h-domes and the threshold value used for defining the markers. As the 3D height data is representing a naturally textured surface a smoothing step is introduced as preprocessing. The amount of smoothing then becomes a third variable. The smoothing is done by mean filtering and the third parameter is therefore the side length of the smoothing kernel.

A novel segmentation error measure was defined and computed in the following way by comparing the automatic segmentation result to an expert annotation. The expert annotation was made by manually segmenting four different images of a total of 467 barley grains.

1. Compute an automatic watershed segmentation with a certain set of the three parameters:
   - Size of the smoothing filter.
   - Size of h used for h-domes.
   - Threshold used on the h-domes.
2. Loop through all the grains in the expert annotation and for each one do the following:
   (a) Find the biggest object in the automatic segmentation result that overlaps with the expert annotation of the grain.
   (b) Count the number of pixels that are wrongly found to be part of the grain and refer to these pixels as "extra pixels".
   (c) Count the number of pixels that are missing from the grain and refer to these pixels as "missing pixels".
   (d) Convert the two above found numbers into a percentage of the grains area in the expert annotation.
   (e) The error measure is defined as a combination of these two numbers: $\sqrt{(\%\text{ extra pixels})^2 + (\%\text{ missing pixels})^2}$

For each grain in the expert annotation this gives a scalar error measure indicating how well the segmentation works with the specific set of parameters. As an exhaustive high resolution search for the optimal parameters would take too long the following approach was taken

- A value of $h = 1$ and a threshold of 0.5 was used as these values provided good preliminary results. Then a sweep was made through different kernel sizes of the mean filter. This was done individually for each of the four images. The results are seen in Figure 5.17 (left). The error measure is seen to flatten out as the kernel size increases. As oversmoothing is not desired a kernel size of 6 pixels was chosen as this is where the plateau begins.

- Now using a fixed kernel size of 6 pixels and a fixed threshold of $0.5h$ a sweep was made through different values of h. The resulting data is seen in Figure 5.17 (middle) and is fitted with a cubic function (the red line) as this gives the best fit without visible overfitting. The minimum of the red line is at $h = 1.69$ and this value is therefore considered the optimal.
- Finally using a fixed kernel size of 6 pixels and a fixed value of $h = 1.69$ a sweep was made through different values of the threshold. The results are seen in Figure 5.17 (right) and the graph is again fitted with a cubic function as this best fits the underlying data without visible overfitting. The minimum of the red line is at 0.45 and this value is therefore considered to be the optimal threshold. Note that 0.45 is in procent of the chosen value of h, so the actual threshold value is $0.45h = 0.76$.

In summary the optimal parameters were found to be using a smoothing kernel of size 6 pixels on each side, using a value of $h = 1.69$ for computing the h-domes, and using a threshold of 0.76 to define the markers.

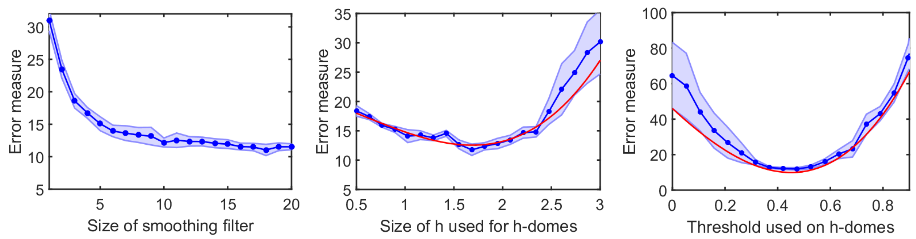


Figure 5.17: *The error measure as a function of segmentation parameters. The blue dots are the average error of the four images segmented using a given set of parameters. The shaded pale blue region represents the standard error of the mean and may be regarded as the uncertainty for each measurement. The last two graphs have each been fitted with a cubic function as this best fits the underlying data without visible overfitting. The minimum of these cubic functions are considered to be the optimal values.*

Figure 5.18: *Examples of where the 2D approach have cut grains in half across the middle. Note that this error is happend to ≈ 1.5% of the total 467 grains.*

5.2.4 Comparison of 2D and 3D

A comparison of the measured error for each of the 467 barley grains for both the 3D approach and the 2D approach is visualised in Figure 5.19. Five important properties become apparent when analysing the figures:

- The 2D approach is for the most part outperforming the 3D based method.
- The points furthest to the right represent chunks of a few grains that the segmentation was not able to separate. The 3D approach is better able to handle these chunks of grains.
- The 3D approach oversegments resulting in serrated edges giving many grains a high value of missing pixels.
- The 2D approach biggest problem is missing pixels.
- The 2D approach has a tendency to sometimes cut grains in half across the middle. Further analysis have revealed that is what happened to the seven grains that are missing between approximately 30% and 60% of their pixels. Examples can be seen in Figure 5.18 on the previous page. Note that the seven grains that were cut in half correspond to ≈ 1.5% of the total 467 grains.

Figure 5.20 and 5.21 on the facing page show a selection of 36 typical segmentation results for 2D and 3D approach respectively. Note how the edges of the grains from the 2D approach are smoother and not as jagged as the edges around the grains from the 3D approach. Also as the 3D approach is based on a height map in a few cases it cuts a grain in half vertically through the middle of the grain. This happens as the grain on its ventral side are naturally having a groove whose sides are acting as a catchment basin. For most grains this groove is deeper then the threshold of 0.76mm used to define the markers and the groove is therefore not filtered away in the marker image.

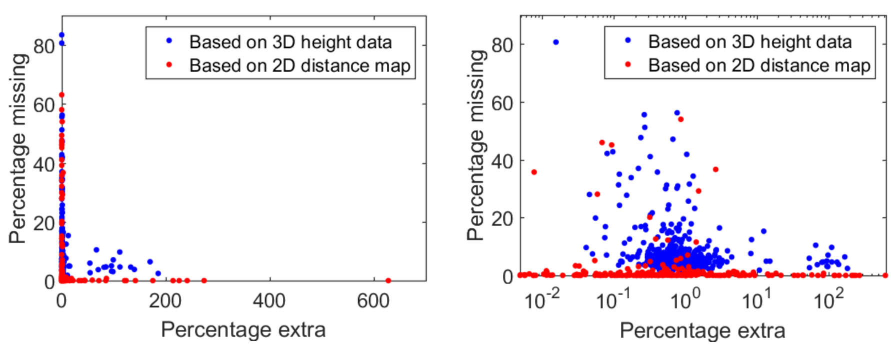


Figure 5.19: *The measured error for each of the 467 grains for both the 2D and 3D approach. The plot to the right has a logarithmic scale on the x-axis.*

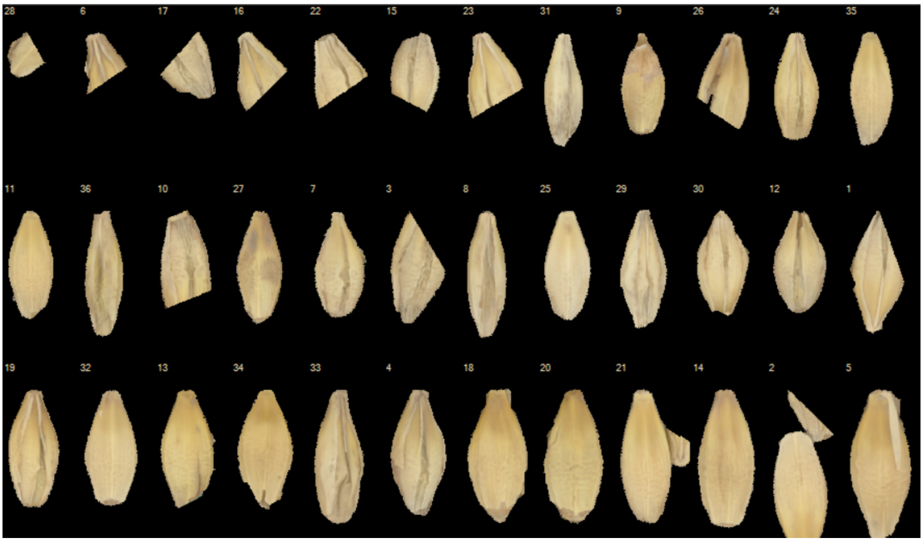


Figure 5.20: *A selection of 36 typical segmentation results from the **2D** approach based on a distance map.*

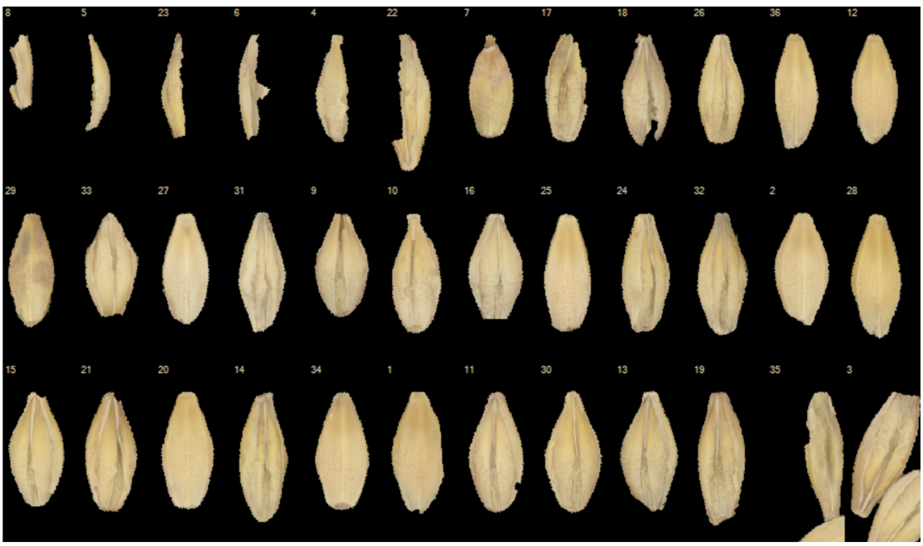


Figure 5.21: *A selection of 36 typical segmentation results from the **3D** approach based on a 3D hight map.*

5.2.5 A combined spectral and 3D approach

It was natural to examine how well a 3D segmentation performed on its own. However it was found that the segmentation only based on the 3D height map is outperformed by the segmentation only based on the 2D distance map. However both have their benefits and drawbacks and this section proposes a way to combine the benefits from both approaches while still excluding most of their respective disadvantages. The proposed approach is outlined below.

Segment into foreground and background
: The 2D approach uses canonical discriminant analysis (details can be found in [1] or [45]) to make a very accurate segmentation between the background and the grains. This means that the edges of grains appear sharp and are smoothly following the grains natural shape. Furthermore there are no holes in the foreground mask. This can not be achieved based on the 3D data, as there are missing values in the data due to occlusions. A combined approach should therefore use canonical discriminant analysis to do the initial segmentation into background and foreground.

Compute the h-domes and the marker
: The optimal marker marks the entire area of all the individual grains except the outermost pixels around the border of each individual grain. This would allow the watershed transform to only find ridge lines at the correct separations between the individual grains. However if one had such a marker one would already have solved the segmentation problem. Of the 2D and 3D approach the 3D approach provides the best h-domes and consequently also the best marker. This is claimed on the basis of experimental results and the fact that two touching objects lying side by side forming a convex shape can be distinguished in the 3D height map, but would be perceived as a single object in the 2D distance transform.
A combined approach should therefore use the h-domes and marker computed on the basis of the 3D height map. It is though also a possibility to use both the marker computed on the 2D and 3D data. This might provide even better results as none of them are excessive, but are just mostly overlapping.

Perform a watershed segmentation
: Compute a marker-controlled watershed segmentation using the marker just described.

As the marker is computed from the 3D height map the middle of each grain is represented in the marker and as such no grains should be cut in half across the "waist" in the way the existing 2D approach has a tendency of doing. The examples of this error seen in Figure 5.18 should therefore be eliminated using the above described combined approach. As the problem of grains that were cut in half across their "waist" was observed to happen to $\approx 1.5\%$ of the grains segmented an improvement of at least $\approx 1.5\%$ is expected from this combined algorithm. The first seven grains from the left in the top row in Figure 5.20 on the previous page would therefore not be cut in half and presumably segmented correctly.

Chapter 6

Conclusion

An integrated structural light scanner has been designed for use in the VidemeterLab instrument allowing for combined 3D, multispectral, and fluorescence imaging. The specifications and requirements for the system have been drawn up and listed in chapter one. Possible approaches for designing such an integrated 3D measurement system was examined by analysing the pros and cons of selected types of structured light solutions and time-of-flight technology.
Three different approaches to phase shifting profilometry were selected, comprehensively documented and reported. An approach based on Fourier analysis was selected to reconstruct the 3D topology both quickly and accurately. The problem of phase unwrapping was also studied and a dual-wavelength solution selected. Algorithms for triangulation of points in 3D space were discussed and a computationally effective algorithm derived. The accuracy of the systems 3D reconstructions were estimated and analysed. The lateral accuracy is determined by camera resolution to 25.9 micrometers and the height error was found to be normally distributed around zero with a standard deviation of just 34.7 micrometers and so 95.5% of the height errors are expected to be below 69.4 micrometers and thus well under the 100 micrometer specification stated in the introduction of this thesis.
It was discussed and explained how to perform accurate and robust stereo calibration of the camera and the projector in order to estimate their intrinsic and extrinsic parameters. The used method does not rely on an initial calibrated camera and therefore does not propagate camera calibration error into the projector calibration. By using radical interpolation to translate projected features into the projectors internal coordinate system sub-pixel accuracy was achieved for both the camera and the projector using ordinary stereo calibration. Estimation uncertainties have been computed for all

calibrated parameters and are all seen to be small which further indicate a successful and very accurate calibration. The used 3D calibration is consistent with the existing 2D geometric calibration on a flat surface.
Lastly two specific applications that benefit from combining 3D and multispectral imaging were introduced and evaluated. First a novel algorithm was presented for classification of grains orientation into the categories of either dorsal or ventral. It was shown that the introduced approach is statistically significantly outperforming the 2D alternative. It was then studied how algorithms for segmentation of granular products, such as rice, grains or seeds, benefit from combining the multispectral image with 3D data. A novel supervised method was introduced to find optimal parameters for marker-controlled watershed segmentation. Then the shortcomings of the existing 2D and the novel 3D approach was analysed. Finally a modification to the existing 2D approach was presented that are expected to increase the number of correctly segmented grains by $\approx 1.5\%$.

The designed system is able to perform 3D measurements while the VideometerLabs sphere is down regardless if the VideometerLab is mounted over a conveyor belt or not and only extends the acquisition time by ≈ 0.6 seconds (corresponding to $\approx 10\%$) depending on the used settings. The developed system has the desired measuring volume of $\varnothing$110mm lateral and a height of 30mm. Both the 3D measurement and the calibration are able to handle varying intensity of objects in the same manner as the light setup[1] handles the dynamics of the spectral image. The measured 3D topographical map has the same sampling as the spectral image and all pixels in the 3D topographical map are handled properly including occluded and unobserved pixels. The added production cost of the VideometerLab is minimized allowing for the 3D measurement system to become a regular feature rather than an optional one. The designed system can be integrated into the VideometerLab instrument and software in a way that works smoothly with the existing hardware and software and a topographical map can then be provided as an additional band in the spectral image.

The goal of this thesis was to design:

A system for measuring 3D simultaneously with spectral image recording in the VideometerLab4 instrument, analysis algorithms to exploit the combined 3D and spectral information, and to demonstrate that this can be utilized efficiently in 1-3 applications.

Based on the above conclusion this goal is considered successfully fulfilled.

[1] An integrated software that controls the strobe time for the individual LEDs based upon the reflective properties of the sample. In this way saturated pixels are avoided independently of the sample.

6.1 Future work

This section lists aspects of future work that would be relevant to consider doing if it was decided to further develop the integrated 3D measurement system. The listed points are independent and can be implemented separately in any order. The list is sorted by what the author considers most essential. The first two items are very quick to implement and expected to give an immediate improvement in the accuracy of 3D measurements.

- Rotate the projector in such a way that its projection is centred in the cameras field of view. This will minimize the influence of non-linear lines distortion in the projector optics and thereby increase the accuracy of the measured 3D data.
- Modify the projected images to be black outside the region that hits the sample area. In the current setup the full image is projected with the consequence that approximately half of the projected light is hitting the inner surface of the sphere before it is bounced around inside the sphere before finally hitting the sample area. These unintended internal light reflections cause the left part of the 3D reconstruction to have a significantly decreased signal to noise ratio. An example of this can be seen as the turquoise colour in the left side of Figure 3.11 on page 40.
- It was described in section 3.6.4 on page 34 that a noisemask could be computed based on thresholding the signal to noise ratio for each individual camera pixel. This noise mask was used to discard pixels from the 3D reconstruction on an individual basis in case of excessive noise. The used value for the threshold was manually chosen in this project. It is hypnotized that the first peak in the histogram of the noisemask (see e.g. Figure 3.11 on page 40) is always the occluded pixels and therefore that a usefull value for the threshold can be found by computing the middle between the first and second peak in the referred histogram.
- Build a mechanical device that allows to set, adjust and maintain a fixed projector focus. Find the optimal focus and measure how much it improves the accuracy of the 3D measurements.
- Apply a linear diffuser in extension of the projector. This would lead to each point in the scene being illuminated by an extended light source rather than a point source and thereby resulting in a reduction of both specularities and shadows. See Figure 6.1 on the next page for an illustration of the methods potential. The theory behind the idea and further explanation can be found in [54].
- It is unknown how the 3D accuracy is affected by the surface colour of the scanned object. It may therefore be interesting to 3D scan an object which gradually change colour. Such an object could be made by printing a colour gradient on a normal printer.
- It remains to verify the calibration and 3D accuracy on two or more different instruments to prove transferability of the technology.

- Replace the projectors optics allowing the projector to project its entire image in the camera field of view. This would better utilize all the projectors pixels allowing for a more accurate height estimation.
- Phase shifting profilometry requires a linear camera-projector response. That is a linear correlation between the projected colour and the cameras perceived colour. All images were taken with 0EV exposure compensation, no gamma correction and no added sharpening or other in camera postprocessing. A linear response is therefore expected but has not been verified. It is therefore recommended to verify and possibly adjust the camera-projector response.
- As the stereo calibration performed is based on the very well known and widely used method of Zhang [40] and Heikkilä [41] the use of a perfect planar calibration target is assumed. Due to their optimization forces the control points to be in the xy-plane by hardcoding their z-coordinate to zero. According to the Z. Wang et al. [42] the assumption of a perfect planar calibration target can be avoided using bundle adjustment to simultaneously optimize all camera parameters *and* the possible defects of the calibration target. The work done by Z. Wang et al. was not implemented as the benefits of their work was not considered to justify a possibly time-consuming implementation. An improvement of the 3D accuracy is expected from implementing their work but it remains to assess whether the improvement is worth the implementation time.
- A 3D viewer combining the spectral and 3D information could be made.

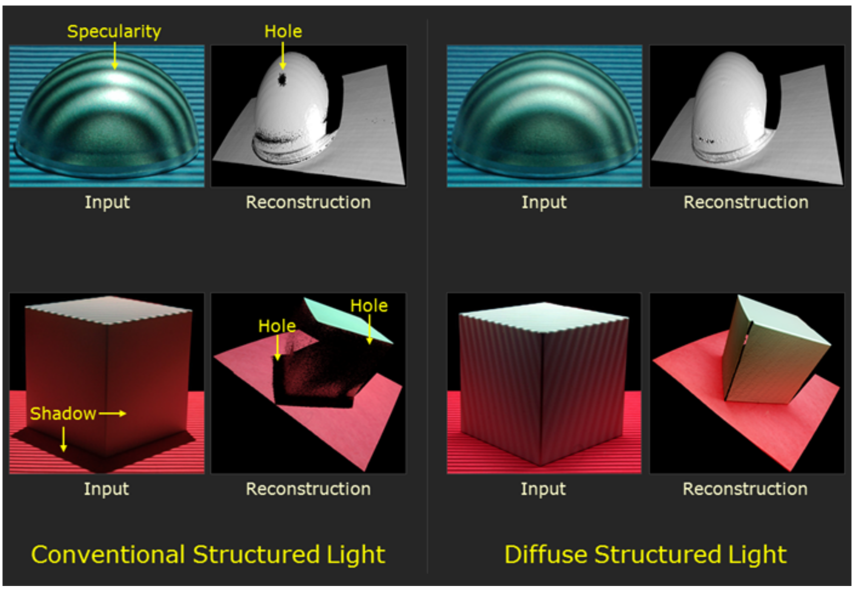


Figure 6.1: *An illustration of the benefits of using diffuse structured light [55]. Note that this figure appears more clearly when viewed on a screen. A link to an electronic version of this report can be found in the preface.*

Appendix A

Technical specifications of the VideometerLab4

This appendix contains the technical specifications of the VideometerLab4 instrument described in section 1.2 on page 3. Further information can be found online at http://videometer.com or be acquired by contacting Videometer.

Light sources	19 high power LED sources with a range from 365 nm to 970 nm. One optional external light source.
Image size	2192 × 2192 pixels (optional 2704 × 2704).
Resolution	≈ 41µm / pixel (optional ≈ 33µm).
Dynamic range	Optimized according to the application using autolight setup.
Calibration	Absolute reflectance calibration using two reflectance calibration targets and one geometric calibration target. Simple calibration wizard procedure that takes 3 minutes.
Sample size	Free height max. 90 mm, diameter of inspection opening 110 mm.
Time of complete analysis	5-10 seconds per sample.
Dimensions	490-585 mm(h) × 420 mm(w) × 590 mm(d).
Weight	14.1 kg (net), 26.6 kg (gross).
Power supply	100 – 240 VAC, 50/60 Hz.
Ambient temperature	Operation: 5 – 40 °C, Storage; -5 – 50 °C.
Ambient humidity	20-90 % RH non-condensing.
PC requirements	Minimum configuration: Intel i7 or better, 16 GB RAM, USB2 port, USB3 SuperSpeed port.
Software	Microsoft Windows 7/8.1/10 Professional 64 bit, full windows update.
Hardware options	Darkfield/brightfield backlight. Standardized backgrounds. Flight case, 57 cm(h) × 50 cm(w) × 71 cm(d). Automatic sample handling.
Software options	Image processing toolbox (IPT). Spectral imaging toolbox (MSI). Blobs toolbox.

Appendix B

Proof of special case of 3.12

This appendix contains proof that the three step equation 3.3 on page 21 to compute the phase ϕ based on only three images is in fact a special case of the derived equation 3.12 on page 22 that used an optional $N \geq 3$ images to compute the phase with higher accuracy.

Equal phase steps of fixed size $\alpha = 2\pi/3$ are used with $\delta_k = \{-2\pi/3, 0, 2\pi/3\}$ for $k = \{1, 2, 3\}$. If inserted into equation 3.2 on page 19 the following general solution for ϕ was found

$$\phi(x,y) = tan^{-1}\left(\sqrt{3}\frac{I_1 - I_3}{2I_2 - I_1 - I_3}\right) \tag{B.1}$$

The derived N step equation is

$$\phi(x,y) = tan^{-1}\left(\frac{\sum_{n=0}^{N-1} I_n^c sin\left(\frac{2\pi n}{N}\right)}{\sum_{n=0}^{N-1} I_n^c cos\left(\frac{2\pi n}{N}\right)}\right) \tag{B.2}$$

An expression of a similar structure where only the images subscript differ can easily be achieved by setting N = 3 and simplifying B.2. However if an identical expression is wanted the same naming convention is needed for the images used in both equations. To determine subscripts consistent with equation 3.6 on page 21 the subscripts are interchanged as follows

δ_k	Name in three step method	Name in N step method
$-2\pi/3$	I_1	I_1
0	I_2	I_0
$2\pi/3$	I_3	I_2

Table B.1: *Subscript alterations in accordance with equation 3.6 on page 21.*

With $N = 3$ the expression in B.2 becomes

$$\phi(x,y) = tan^{-1}\left(\frac{\sum_{n=0}^{3-1} I_n^c sin\left(\frac{2\pi n}{3}\right)}{\sum_{n=0}^{3-1} I_n^c cos\left(\frac{2\pi n}{3}\right)}\right) \tag{B.3}$$

$\Downarrow$

$$\phi(x,y) = tan^{-1}\left(\frac{I_0^c sin\left(\frac{2\pi 0}{3}\right) + I_1^c sin\left(\frac{2\pi 1}{3}\right) + I_2^c sin\left(\frac{2\pi 2}{3}\right)}{I_0^c cos\left(\frac{2\pi 0}{3}\right) + I_1^c cos\left(\frac{2\pi 1}{3}\right) + I_2^c cos\left(\frac{2\pi 2}{3}\right)}\right) \tag{B.4}$$

$\Downarrow$

$$\phi(x,y) = tan^{-1}\left(\frac{I_0^c sin(0) + I_1^c sin\left(\frac{2\pi 1}{3}\right) + I_2^c sin\left(\frac{2\pi 2}{3}\right)}{I_0^c cos(0) + I_1^c cos\left(\frac{2\pi 1}{3}\right) + I_2^c cos\left(\frac{2\pi 2}{3}\right)}\right) \tag{B.5}$$

$\Downarrow$

$$\phi(x,y) = tan^{-1}\left(\frac{\frac{\sqrt{3}}{2} I_1^c - \frac{\sqrt{3}}{2} I_2^c}{I_0^c - 0.5 I_1^c - 0.5 I_2^c}\right) \tag{B.6}$$

$\Downarrow$

$$\phi(x,y) = tan^{-1}\left(\frac{\frac{\sqrt{3}}{2}(I_1^c - I_2^c)}{I_0^c - 0.5 I_1^c - 0.5 I_2^c}\right) \tag{B.7}$$

$\Downarrow$

$$\phi(x,y) = tan^{-1}\left(\frac{\sqrt{3}(I_1^c - I_2^c)}{2I_0^c - I_1^c - I_2^c}\right) \tag{B.8}$$

$\Downarrow$

$$\phi(x,y) = tan^{-1}\left(\sqrt{3}\frac{I_1^c - I_2^c}{2I_0^c - I_1^c - I_2^c}\right) \tag{B.9}$$

Using the subscript alterations from table B.1 the deduced equation is seen to be identical to equation B.1. Thus proving that equation B.1 is simply a special case of equation B.2 where $N = 3$.

$\square$

Appendix C

Remaining plots for the 3D accuracy analysis

This appendix contains the remaining plots referred to in the analysis of the 3D height accuracy made in section 3.7 on page 42 in chapter three.

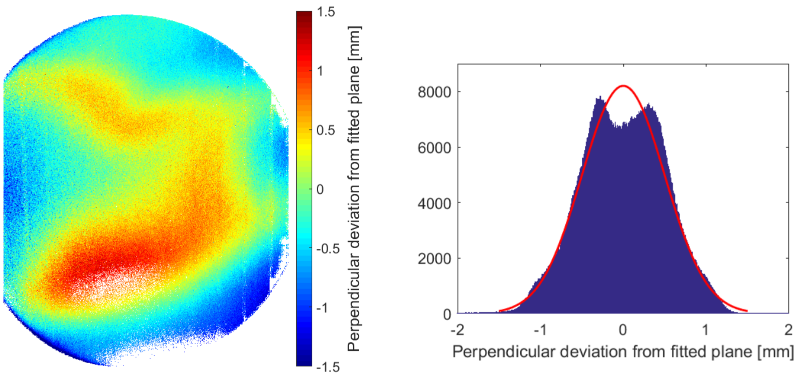


Figure C.1: *The image to the left shows the perpendicular deviation for each pixel across the surface of the glass plate when a mat spray was applied on top and a white piece of paper manually held in place under the glass. A histogram of the deviations is seen to the right. A normal distribution has been fitted to the histogram and shown in red for comparison.*

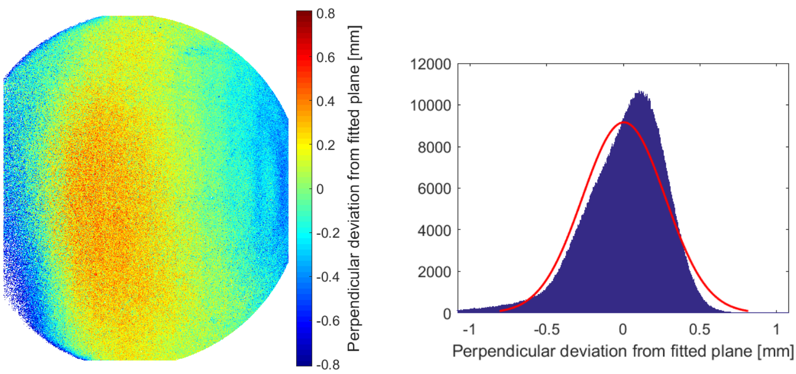


Figure C.2: *This figure illustrates the same as Figure C.1 however with a white spray applied under the glass during scanning. Notice the decrease in error compared with Figure C.1.*

Appendix D

Technical specifications of the DLP LightCrafter

The projector used in this project is a DLP LightCrafter Evaluation Module produced by Texas Instruments and seen in Figure D.2. From now on simply referred to as the LightCrafter. The factual information and figures in this appendix is from the official "DLP LightCrafter User's Guide (Rev. E)".

The light engine in the LightCrafter consists of the optics, the red, green, and blue LEDs, and a 608 × 684 diamond pixel 0.3 inch WVGA DMD. An illustration of the diamond shaped pixel array is seen in Figure D.1 on the following page. The light engine is capable of 20 lumens out-of-the-box with support to 50 lumens with the addition of active cooling. If only short blinks of light is needed the full 50 lumens can be used without active cooling. Tabel D.1 lists relevant specifications of the light engine.

f-number	2.2
Throw radio	1.66
Vertical offset	100
Focus range	364mm - 2169mm
Image diagonal size	24.5mm - 152.4mm

Table D.1: *Specifications of the light engine.*

The LightCrafter also contains a unit called MSP430 that monitors the light engine's thermistor to provide a safety shutdown of the LightCrafter if excessive heat is measured on the green LED (above 70°C). The LightCrafters optical engine is mounted on top of a thermal plate to provide passive cooling. Passively cooled systems (no extra heat sinks or fans) have a thermal limit resulting in LED currents under 633mA. Actively cooled systems (extra heat sink and fan) have a thermal limit resulting in LED currents under 1.5A. However as only 12 short bursts of monochrome light are needed followed by a break of at least six seconds a LED current of 1.98A is safely used[1] on one of the LED's while the other two are turned off.

The entire LightCrafter has the dimensions of 116.5mm long, 65mm wide, and 23mm tall. Figure D.3 on the next page shows the LightCrafter dimensions.

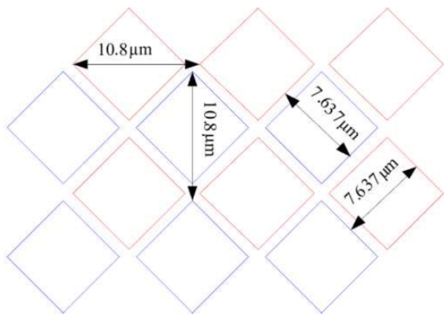


Figure D.1: *Schematic illustration of the diamond shaped pixel array.*

[1] According to a Texas Instruments employee answering this question in their online E2E community forum.

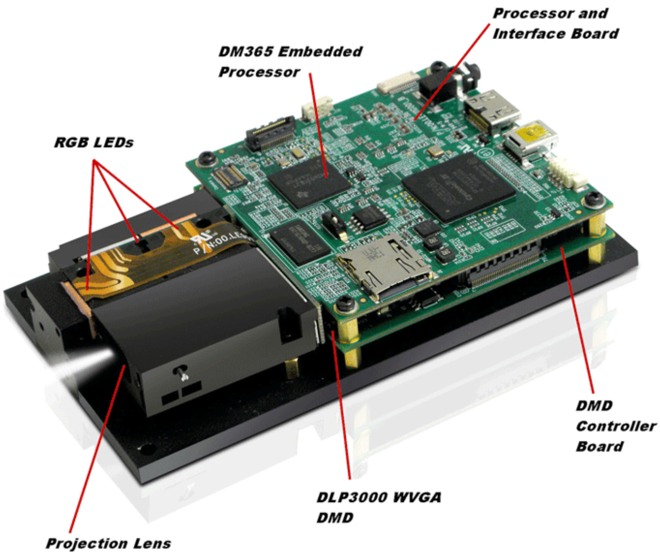


Figure D.2: *The LightCrafter and its key components.*

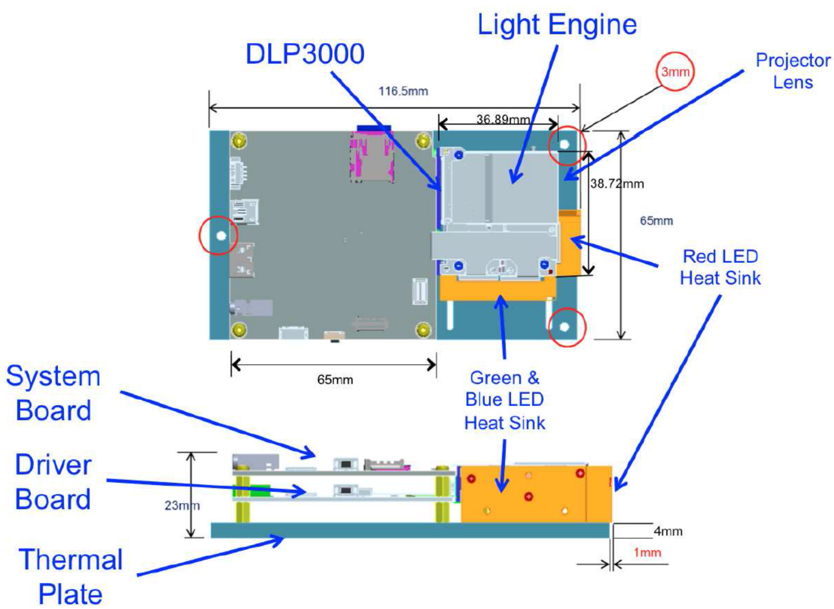


Figure D.3: *The dimensions of the LightCrafter.*

Appendix E

Uncertainty estimates of the extrinsic parameters

This appendix contains a table of uncertainty estimates of the cameras extrinsic parameters. The intrinsic parameters can be found in section 4.4 on page 57. The extrinsic parameters for the projector are found by applying the rotation and translation shown in section 4.4 and for clarity also reproduced in the table directly below.

```
Position and orientation of the camera relative to the projector
----------------------------------------------------------------
Rotation of the projector:
        [   0.8898 +/- 0.0099     0.0323 +/- 0.0045    -0.4552 +/- 0.0019 ]

Translation of projector (mm):
        [ 171.7069 +/- 0.7663   -11.7599 +/- 0.8716   -49.9412 +/- 3.0683 ]
```

```
Rotation vectors:
-----------------
[    0.0309 +/- 0.0024       -0.0094 +/- 0.0025        -1.5203 +/- 0.0001  ]
[   -0.0682 +/- 0.0022        0.0153 +/- 0.0021        -0.5750 +/- 0.0001  ]
[   -0.0683 +/- 0.0021       -0.0064 +/- 0.0020        -0.0332 +/- 0.0001  ]
[   -0.0065 +/- 0.0024       -0.0415 +/- 0.0021         0.6599 +/- 0.0001  ]
[   -0.0124 +/- 0.0026       -0.0183 +/- 0.0026         1.4421 +/- 0.0001  ]
[   -0.0306 +/- 0.0027       -0.0281 +/- 0.0029         2.2229 +/- 0.0002  ]
[   -0.0466 +/- 0.0033       -0.0201 +/- 0.0038         3.1117 +/- 0.0002  ]
[    0.0380 +/- 0.0028        0.0051 +/- 0.0032        -2.4110 +/- 0.0001  ]
[   -0.0466 +/- 0.0022       -0.0849 +/- 0.0021        -1.5631 +/- 0.0002  ]
[   -0.0307 +/- 0.0020       -0.1046 +/- 0.0018        -1.1462 +/- 0.0002  ]
[    0.0076 +/- 0.0018       -0.1269 +/- 0.0015        -0.0232 +/- 0.0001  ]
[    0.0171 +/- 0.0018       -0.1190 +/- 0.0015         0.1773 +/- 0.0001  ]
[    0.0635 +/- 0.0021       -0.0937 +/- 0.0021         1.6541 +/- 0.0002  ]
[   -0.1082 +/- 0.0031       -0.0006 +/- 0.0036        -3.0021 +/- 0.0002  ]
[   -0.0786 +/- 0.0027       -0.0511 +/- 0.0029        -2.3050 +/- 0.0002  ]
[   -0.1844 +/- 0.0012       -0.2580 +/- 0.0011        -1.5487 +/- 0.0002  ]
[    0.0185 +/- 0.0010       -0.3148 +/- 0.0009        -0.0583 +/- 0.0002  ]
[    0.2290 +/- 0.0011       -0.2459 +/- 0.0011         1.5641 +/- 0.0002  ]
[    0.3105 +/- 0.0012       -0.1680 +/- 0.0014         2.2188 +/- 0.0002  ]
[   -0.2535 +/- 0.0013       -0.2054 +/- 0.0013        -2.0321 +/- 0.0002  ]

Translation vectors (mm):
-------------------------
[  -18.1111 +/- 0.2478       28.8443 +/- 0.2442      460.3735 +/- 1.8305  ]
[  -28.3665 +/- 0.2458       -4.3046 +/- 0.2428      457.9543 +/- 1.8224  ]
[  -24.3809 +/- 0.2459      -19.4661 +/- 0.2425      457.8048 +/- 1.8235  ]
[   -4.8317 +/- 0.2468      -33.0413 +/- 0.2434      458.6623 +/- 1.8271  ]
[   15.5034 +/- 0.2472      -31.4609 +/- 0.2438      459.8023 +/- 1.8324  ]
[   32.6195 +/- 0.2477       -8.6864 +/- 0.2445      461.3297 +/- 1.8356  ]
[   25.8141 +/- 0.2477       18.0015 +/- 0.2445      461.3321 +/- 1.8341  ]
[    3.5570 +/- 0.2478       32.0772 +/- 0.2443      460.8945 +/- 1.8328  ]
[  -21.3896 +/- 0.2433       27.9275 +/- 0.2400      451.6206 +/- 1.8112  ]
[  -31.7069 +/- 0.2430       18.1641 +/- 0.2396      451.0786 +/- 1.8111  ]
[  -34.9164 +/- 0.2424      -15.6011 +/- 0.2393      450.6897 +/- 1.8137  ]
[  -28.1751 +/- 0.2424      -24.3200 +/- 0.2392      450.5988 +/- 1.8123  ]
[   17.9114 +/- 0.2431      -26.4134 +/- 0.2398      451.8703 +/- 1.8134  ]
[   25.1043 +/- 0.2438       26.9967 +/- 0.2404      452.6857 +/- 1.8164  ]
[    6.4406 +/- 0.2435       37.6834 +/- 0.2405      452.4541 +/- 1.8142  ]
[  -20.5466 +/- 0.2365       32.4018 +/- 0.2331      439.6882 +/- 1.7989  ]
[  -30.4926 +/- 0.2358      -15.2463 +/- 0.2329      438.9709 +/- 1.7961  ]
[   18.0904 +/- 0.2367      -26.7995 +/- 0.2330      440.1412 +/- 1.8014  ]
[   32.3154 +/- 0.2368       -9.4365 +/- 0.2334      440.7493 +/- 1.8050  ]
[   -3.0713 +/- 0.2366       33.4811 +/- 0.2337      440.2218 +/- 1.8015  ]
```

Bibliography

[1] Jens Michael Carstensen and Nette Schultz. Dermatological conditions assessed by spectral imaging and normalized canonical discriminant analysis. Paper intended for the International Conference on Image Analysis and Processing 2013.

[2] Safety of laser products - Part 1: Equipment classification and requirements. Technical Report IEC 60825-1:2014, International Electrotechnical Commission, 2014.

[3] Ritchie Logan. Machine vision with depth: Industrial imaging with high-resolution time-of-flight cameras. Talk at Visionday 2015, DTU Compute, May 2015.

[4] Joaquim Salvi, Sergio Fernandez, Tomislav Pribanic, and Xavier Llado. A state of the art in structured light patterns for surface profilometry. *Pattern recognition*, 43(8):2666–2680, 2010.

[5] Henrik Aanæs. Lecture notes on computer vision. Technical University of Denmark, September 2011.

[6] Jason Geng. Structured-light 3d surface imaging: a tutorial. *Advances in Optics and Photonics*, 3(2):128–160, 2011.

[7] Hugh D. Young, Lewis S. Ford, and Roger A. Freedman. *University Physics*. Pearson Education, 12st edition, 2007.

[8] Song Zhang. *High-resolution, real-time 3-D shape measurement*. PhD thesis, Citeseer, 2005.

[9] Seung-Woo Kim, Min-Gu Kang, and Geon-Soo Han. Accelerated phase-measuring algorithm of least squares for phase-shifting interferometry. *Optical Engineering*, 36(11):3101–3106, 1997.

[10] C. J. Morgan. Least-squares estimation in phase-measurement interferometry. *Optics Letters*, 7(8):368–370, 1982.

[11] Daniel L Lau, Kai Liu, and Laurence G Hassebrook. Real-time three-dimensional shape measurement of moving objects without edge errors by time-synchronized structured illumination. *Optics letters*, 35(14):2487–2489, 2010.

[12] Kai Liu, Yongchang Wang, Daniel L Lau, Qi Hao, and Laurence G Hassebrook. Gamma model and its analysis for phase measuring profilometry. *JOSA A*, 27(3):553–562, 2010.

[13] María F Salfity, Pablo D Ruiz, Jonathan M Huntley, Martin J Graves, Rhodri Cusack, and Daniel A Beauregard. Branch cut surface placement for unwrapping of undersampled three-dimensional phase data: application to magnetic resonance imaging arterial flow mapping. *Applied optics*, 45(12):2711–2722, 2006.

[14] Thomas J Flynn. Two-dimensional phase unwrapping with minimum weighted discontinuity. *JOSA A*, 14(10):2692–2701, 1997.

[15] Hussein Abdul-Rahman. *Three-dimensional Fourier fringe analysis and phase unwrapping*. PhD thesis, Citeseer, 2007.

[16] Jin-Jung Chyou, Shean-Jen Chen, and Yi-Kuang Chen. Two-dimensional phase unwrapping with a multichannel least-mean-square algorithm. *Applied optics*, 43(30):5655–5661, 2004.

[17] Henrik O. Saldner and Jonathan M. Huntley. Temporal phase unwrapping: application to surface profiling of discontinuous objects. *Applied optics*, 36(13):2770–2775, 1997.

[18] Jakob Wilm, Oline V. Olesen, and Rasmus Larsen. Accurate and simple calibration of DLP projector systems. In *SPIE MOEMS-MEMS*. International Society for Optics and Photonics, 2014.

[19] Yongchang Wang. Novel approaches in structured light illumination. 2010.

[20] Jakob Wilm, Oline V Olesen, and Rasmus Larsen. Slstudio: Open-source framework for real-time structured light. In *Image Processing Theory, Tools and Applications (IPTA), 2014 4th International Conference on*, pages 1–4. IEEE, 2014.

[21] Michael Spivak. *Calculus on Manifolds: A Modern Approach to Clasical Theorems of Advanced Calculus*. WA Benjamin, 1965.

[22] Robert J. Valkenburg and Alan M. McIvor. Accurate 3D measurement using a structured light system. *Image and Vision Computing*, 16(2):99–110, 1998.

[23] J. Massich D. Fofi G. Falcao, N. Hurtos. "Projector-camera calibration toolbox". Technical report, University of Burgundy, 2009. Available at http://code.google.com/p/procamcalib.

[24] Filip Sadlo, Tim Weyrich, Ronald Peikert, and Markus Gross. A practical structured light acquisition system for point-based geometry and texture. In *Point-Based Graphics, 2005. Eurographics/IEEE VGTC Symposium Proceedings*, pages 89–145. IEEE, 2005.

[25] Makoto Kimura, Masaaki Mochimaru, and Takeo Kanade. Projector calibration using arbitrary planes and calibrated camera. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–2. IEEE, 2007.

[26] Jiarui Liao and Lilong Cai. A calibration method for uncoupling projector and camera of a structured light system. In *Advanced Intelligent Mechatronics, 2008. AIM 2008. IEEE/ASME International Conference on*, pages 770–774. IEEE, 2008.

[27] Koichiro Yamauchi, Hideo Saito, and Yukio Sato. Calibration of a structured light system by observing planar object from unknown viewpoints. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE, 2008.

[28] Gao Wei, Wang Liang, and HU Zhan-Yi. Flexible calibration of a portable structured light system through surface plane. *Acta Automatica Sinica*, 34(11):1358–1362, 2008.

[29] Hafeez Anwar, Irfanud Din, and Kang Park. Projector calibration for 3d scanning using virtual target images. *International Journal of Precision Engineering and Manufacturing*, 13(1):125–131, 2012.

[30] Jamil Draréni, Sandip Roy, and Peter Sturm. Geometric video projector autocalibration. In *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*, pages 39–46. IEEE, 2009.

[31] Jamil Draréni, Sébastien Roy, and Peter Sturm. Methods for geometrical video projector calibration. *Machine Vision and Applications*, 23(1):79–89, 2012.

[32] Samuel Audet and Masatoshi Okutomi. A user-friendly method to geometrically calibrate projector-camera systems. In *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*, pages 47–54. IEEE, 2009.

[33] Ivan Martynov, Joni-Kristian Kamarainen, and Lasse Lensu. Projector calibration by inverse camera calibration. In *Image Analysis*, pages 536–544. Springer, 2011.

[34] Jérémie Mosnier, Francois Berry, and Omar Ait-Aider. A new method for projector calibration based on visual servoing. In *MVA*, pages 25–29, 2009.

[35] Soon-Yong Park and Go Gwang Park. Active calibration of camera-projector systems based on planar homography. In *20th International Conference on Pattern Recognition (ICPR)*, pages 320–323. IEEE, 2010.

[36] Daniel Moreno and Gabriel Taubin. Simple, accurate, and robust projector-camera calibration. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*, pages 464–471. IEEE, 2012.

[37] Song Zhang and Peisen S Huang. Novel method for structured light system calibration. *Optical Engineering*, 45(8):083601–083601, 2006.

[38] Zhongwei Li, Yusheng Shi, Congjun Wang, and Yuanyuan Wang. Accurate calibration method for a structured light system. *Optical Engineering*, 47(5):053604–053604, 2008.

[39] Jean-Yves Bouguet. Camera calibration toolbox for Matlab. California Institute of Technology, October 2004.

[40] Zhengyou Zhang. A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1330–1334, 2000.

[41] Jane Heikkilä and Olli Silvén. A four-step camera calibration procedure with implicit image correction. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 1106–1112. IEEE, 1997.

[42] Minh Vo, Zhaoyang Wang, Long Luu, and Jun Ma. Advanced geometric camera calibration for machine vision. *Optical Engineering*, 50(11):110503–110503, 2011.

[43] Jakob Andreas Bærentzen, Jens Gravesen, François Anton, and Henrik Aanæs. *Guide to Computational Geometry Processing*. Springer London, 2012.

[44] Trevor Hastie, Robert Tibshirani, Jerome Friedman, and James Franklin. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85, 2005.

[45] Kristian Ryder Thomsen. Måling af enzymatiske effekter med spektral billedanalyse. B.S. Thesis, Technical University of Denmark, DTU Compute, 2013.

[46] Andrew N Pressley. *Elementary differential geometry*. Springer Science & Business Media, 2010.

[47] Jan J. Koenderink and Andrea J. van Doorn. Surface shape and curvature scales. *Image and vision computing*, 10(8):557–564, 1992.

[48] Ujwal Bonde, Vijay Badrinarayanan, and Roberto Cipolla. *Multi scale shape index for 3d object recognition*. Springer, 2013.

[49] Tony Lindeberg. Scale-space theory: A framework for handling image structures at multiple scales. In *CERN School of Computing, Egmond aan Zee, The Netherlands, Sep. 8–21, 1996*, pages 27–38, 1996.

[50] Tony Lindeberg. *Scale-Space*. Wiley Online Library, 2008.

[51] Pang Ning Tan, Michael Steinbah, and Vipin Kumar. *Introduction to Data Mining*. Pearson International Education, 2006.

[52] R. Gonzalez and R. Woods. *Digital Image Processing*. Prentice Hall, 2002.

[53] Steve Eddins. The watershed transform: Strategies for image segmentation. *MathWorks: Technical Articles and Newsletters*, 2002.

[54] Shree K. Nayar and Mohit Gupta. Diffuse structured light. In *International Conference on Computational Photography (ICCP)*, pages 1–11. IEEE, 2012.

[55] Columbia University, Computer vision laboratory. http://www.cs.columbia.edu/CAVE/projects/DiffuseSL/.