Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Electronic Systems

Leik Lima-Eriksen

A Distortion-resilient Pattern Codification Strategy for Structured Light 3D Cameras

Master's thesis in Electronics Systems Design and Innovation
Supervisor: Kimmo Kansanen, IES NTNU
Co-supervisor: Martin Ingvaldsen, ZIVID AS
July 2022

NTNU
Norwegian University of Science and Technology

ZIVID

Leik Lima-Eriksen

A Distortion-resilient Pattern Codification Strategy for Structured Light 3D Cameras

Master's thesis in Electronics Systems Design and Innovation
Supervisor: Kimmo Kansanen, IES NTNU
Co-supervisor: Martin Ingvaldsen, ZIVID AS
July 2022

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Electronic Systems

NTNU
Norwegian University of
Science and Technology

Abstract

In today's modern industries, robots are ubiquitous. Their applications range from lifting cargo off pallets to assembling consumer products. Unfortunately, most robots are blind, significantly restricting their use cases. The introduction of 3D cameras makes robots able to work in new and more challenging environments. Zivid AS is a company that specializes in making such cameras, and their cameras utilize structured light. Although the technique offers great accuracy, it struggles in environments that contain highly reflective objects giving rise to so-called interreflections. Considering that most consumer products are wrapped in reflective plastic and that many industrial parts are made of reflective metals, this poses a challenge to increase the utilization of 3D camera systems.

This thesis introduces a new signal processing chain for structured light systems to make them more resilient to reflective objects, without degrading the systems' performance. The processing chain consists of both a geometric constraint and a novel pattern codification strategy. The constraint exploits a restriction on the valid solutions seen by the camera, which makes signal processing faster and more reliable in the presence of interreflections. The pattern codification strategy is a new way to code structured light. It is based on the combination of sequences with good correlation properties and temporally randomly shuffled cosines which whiten out the distortions from shiny objects.

When tested in the presence of multiple shiny objects, the novel pattern codification strategy performs better than the state-of-the-art (GCPS). More than 70% of the camera pixels have residuals within the desired 0.1% target, compared to 60% for GCPS. Moreover, the residuals are spread evenly over the entire camera, instead of being centered around the edges of objects, which could be useful for applications. The codification strategy performs worse at close distances, and it is suggested that it could be caused by an incorrect choice of parameters for these distances.

A limitation of the pattern codification strategy is that it suffers from periodic systematic errors, and these contribute to approximately 50% of the residuals in scenes with a lot of interreflections. Several improvements are suggested, which could reduce these errors. Nevertheless, the codification strategy also suffers from an acquisition time 4.1 times longer compared to the state-of-the-art and an unknown higher computational complexity. This could for many use-cases be a deal breaker and should be taken into consideration.

Sammendrag

Roboter finnes overalt i dagens moderne industrier, og deres bruksområder inkluderer alt fra å løfte kargo av paller til å sette sammen forbrukervarer. Dessverre er de aller fleste roboter blinde, og dette begrenser deres bruksområder betraktelig. Ved å ta i bruk 3D-kameraer kan man bruke roboter i nye og mer utfordrende sammenhenger. Zivid AS er et selskap som spesialiserer seg i å lage slike kameraer, og teknikken som kameraene bruker kalles for strukturert lys. Selv om denne teknikken gir høy nøyaktighet, yter den ofte dårlig i miljøer som inneholder sterkt reflektive objekter som gir opphav til såkalte interrefleksjoner. Med tanke på at de fleste forbrukervarene er pakket inn i reflektiv plast og at mange industrielle deler er laget av reflektive metaller, er dette et hinder mot å øke bruken av 3D-kameraer.

Denne oppgaven presenterer en ny signalbehandlingskjede for systemer som bruker strukturert lys for å gjøre dem mer tolerante mot reflektive objekter, uten å gå på bekostning av deres ytelse. Signalbehandlingskjeden består av både en geometrisk beskrankning og en ny mønster-kodifiseringsstrategi. Beskrankningen utnytter en restriksjon for gyldige løsninger sett fra kameraet, og gjør signalbehandlingen både raskere og mer pålitelig når interrefleksjoner er tilstede. Mønster-kodifiseringsstrategien er en ny måte å kode strukturert lys. Den er basert på kombinasjonen av følger med gode korrelasjonsegenskaper og temporale tilfeldig omstokkede cosinus-bølger som demper forstyrrelsene fra reflektive objekter.

Denne nye mønster-kodifiseringsstrategien yter bedre enn dagens standard (GCPS) når den testes på scener med mange reflektive gjenstander. Mer enn 70% av kamerapikslene har residualer innenfor målet på 0.1%, sammenliknet med 60% for GCPS. Residualene er i tillegg spredt jevnt over hele kameraet, istedenfor å være sentrert rundt kantene på objekter, noe som kan være nyttig for applikasjoner. Teknikken yter verre på korte avstander, og det er foreslått at dette kan være forårsaket av dårlig valgte parametre for disse avstandene.

En svakhet ved denne mønster-kodifiseringsstrategien er at den lider av periodiske systematiske feil, og disse bidrar til omlag 50% av residualene i scener med mange interrefleksjoner. Flere forbedringer for å redusere disse feilene er foreslått. I tillegg lider også teknikken av en 4.1 ganger lengre bildetakingstid sammenliknet med dagens standard, og en ukjent høyere utregningskompleksitet. Dette er egenskaper som potensielt kan hindre bruken av mønster-kodifiseringsstrategien for mange bruksområder, og må tas i betraktning.

Preface

This thesis is submitted in fulfillment of the requirements as part of a Masters of Science degree within Electronics at the Norwegian University of Science and Technology. The thesis was produced in collaboration with Zivid AS and is a continuation of the project thesis.

The author would like to thank the main supervisor of the thesis, Kimmo Kansanen, for great advice and support throughout the work. The author extends his gratitude to co-supervisor Martin Ingvaldsen at Zivid AS for giving the opportunity to work on an exciting challenge of high relevancy, and for providing innovative ideas on which directions to take with the work. Lastly, the author would like to thank Aleksandar Babic at Zivid AS for providing help on how to use a structured light simulator and providing testing scenes of high relevance to the work.

Leik Lima-Eriksen
Trondheim, July 2022

Table of Contents

List of Figures

List of Tables

Acronyms

ACF auto-correlation function.

API application programming interface.

CCF cross-correlation function.

CDF cummulative distribution function.

CFPPS Correlation-Fringed Permuted Phase Shifts.

CIF correlation-identified fringes.

CLI command-line interface.

DFT discrete fourier transform.

DOF Depth of Field.

GCPS Gray-Coded Phase Shifts.

GUI graphical user-interface.

LTI linear time-invariant.

MIMO multiple input multiple output.

PPS permuted phase shifts.

PSF point-spread function.

RGB red, green and blue.

SD standard deviation.

SNR signal-to-noise ratio.

Glossary

capture The raw data originating from the camera inside a structured light system when taking a picture using a particular pattern.

code matrix A matrix $\mathbf{C}$ of dimensions $N_P \times X_P$ describing a particular pattern sequence containing N_P patterns. Each row $(\mathbf{C})_{i\cdot}$ stores the intensity values for all projector columns x_p for a particular pattern i as $(\mathbf{C})_{ix_p}$.

fringe A horizontal section of a pattern consisting of a certain number of projector columns W_F.

pattern A matrix $\mathbf{P}$ of dimensions $Y_P \times X_P$ where each entry $P_{y_p x_p}$ stores the intensity of a particular projector pixel (x_p, y_p).

pattern codification strategy A method for uniquely identifying the corresponding projector column x_p for each of the camera pixels (x_c, y_c).

point cloud A set of data points in space. Might resemble 3D shapes or objects.

pose The relative orientation of the structured light system with respect to the scene.

residual The difference between the estimated originating projector column and the ground truth obtained for a particular camera pixel.

structured light A method of projecting known patterns of light, and using their deformations to reconstruct a 3D point cloud.

Nomenclature

[1] Used to denote a unit-less quantity.

[cpx] Unit for camera pixel.

[ppx] Unit for projector pixel.

$\mathbf{C}$ Matrix of dimensions $N_P \times X_P$ used for storing a code matrix. The element $(\mathbf{C})_{ix_p}$ stores the intensity of the projector column x_p for the i-th pattern.

$\mathbf{E}$ Residual matrix of dimensions $Y_C \times X_C$.

$\mathbf{M}$ Tensor of dimensions $Y_C \times X_C \times N_P$ used to store a sequence of captures. The entry $\mathbf{M_{y_c x_c i}}$ stores the intensity of the camera pixel (x_c, y_c) using the i-th pattern of a particular code matrix.

$\mathbf{P}$ Matrix of dimensions $Y_P \times X_P$ used for storing a pattern. The element $\mathbf{P_{y_p x_p}}$ specifies the intensity of the projector pixel (x_p, y_p).

$\mathbf{R_n}$ $n \times n$ circular right-shift matrix

σ Standard deviation.

$\mathbf{e_i}$ Elementary column-vector with a 1 at the i-th position and 0 otherwise. Let $\mathbf{A}$ be a matrix. Then $\mathbf{e_i A}$ denotes the i-th *row* of the matrix, whereas $\mathbf{A e_i}$ denotes the i-th *column*.

f_s Spatial frequency

f_t Temporal frequency

N_F Number of fringes in a pattern.

N_P Number of patterns, typically in a code matrix.

W_F Width of a fringe.

1 Introduction

In today's modern industries, robots are ubiquitous. Their applications range from lifting cargo off pallets (depalletization) to assembling consumer products. Unfortunately, most robots are blind. This, in turn, means that each maneuver has to be pre-programmed to follow specific paths, and the objects with which robots interact need to be positioned at exact known locations for the robot to be able to do its job.

Zivid AS is a company that develops 3D cameras for industrial robots. These camera systems make it possible for robots to work in changing environments, which in turn increases their areas of use. Their imaging technique, known as *structured light*, offers great accuracy and is tolerant of many of the distortions that are commonly present in the environments. However, it struggles in environments that contain highly reflective objects. Considering that most consumer products are wrapped in reflective plastic and that many industrial parts are reflective metallic, this poses a challenge to the further adoption of 3D camera systems. The aim of this thesis is to improve the signal processing chain in structured light systems to make it more resilient to these reflective objects, without degrading its performance.

1.1 Background and motivation

The first and second industrial revolutions (1733 to 1913) introduced the world to the concept of factories and machine manufacturing (Engelman 2022). By providing streamlined production services, factories significantly increased the throughput per capita (Zeidan 2021). Furthermore, the usage of machines such as the Spinning Jenny meant that the employees had to do less labor-intensive work (Zeidan 2021).

Several years later in 1961, the world's first industrial robot named Unimate saw its light (Wallén 2008). Unimate transported die castings from a General Motor's assembly line and welded these to the body of cars (Mickle 1999). Traditionally, industrial machines could perform simple tasks, such as printing news papers and weaving textiles. Industrial robots meant a leap in machine usage, as they allowed for the execution of pre-programmed complex mechanical tasks. For these reasons, the introduction of robots during the latter half of the 20^{th} century has later been known as the third industrial revolution (Rifkin 2011).

Since their introduction in 1961, the world has embraced the use of robots, particularly during the 21^{st} century. Today, more than 2.7 million industrial robots are employed around the world in factories and logistics centers (International Federation of Robotics 2019). Here, they perform dull, dangerous, and repetitive tasks, such as spray painting cars and lifting heavy objects. By offloading these tasks from humans, robots significantly improve the lives of workers.

One particular sector that has truly embraced robots is the automotive industry. For every 10 workers in the industry, there are 1.3 industrial robots employed, which is markedly higher than the general average of 0.3 industrial robots per 10 workers

(International Federation of Robotics 2021). Robots move heavy parts around, do spray painting, screwing, and drilling, only to name a few. Another industry sector that has benefited from the employment of robots is the logistics sector. Approximately 10% of all new industrial robots are deployed in this sector (Graat 2020). Machines automate the process of moving and storing goods in logistics centers (Association for Advancing Automation 2020).

Robots have to interact with objects in their environment to perform their tasks. Therefore, a main limiting factor for a robot's performance is to what degree it is able to observe and adapt to its environment. Today, most industrial robots are blind(Golnazarian and Hall 2000). Without observing their environment, robots require that all objects with which they interact are placed at predetermined locations. Typically, this would mean that their interacting objects are fed through reels, such as in PCB pick-and-place machines. The disability significantly restricts the tasks a robot can perform to repetitive tasks only, and exact programming of each movement is required for every task (Association for Advancing Automation 2017).

Equipped with 3D vision capabilities, robots can perform multiple tasks without reprogramming. Changing environments is also less of a problem, as the vision allows for the adaptation to this through the recognition of objects and their positions. This allows for greater flexibility, making robots faster, and increasing their return on investment.

In particular, pick-and-place operations have been notoriously difficult for robots to perform. According to (Association for Advancing Automation 2017), "blind robots could only pick objects from predetermined positions and 2D camera systems could not pick out a part from its background". However, with 3D vision, the system can be taught to recognize the objects, e.g. by inputting 3D CAD files beforehand. These operations then become much easier to perform since the introduction of depth knowledge allows one to measure how far away the objects are. Additionally, pick-and-place operations can be defined through object recognition, which reduces the amount of programming required. These operations are a ground pillar within both logistics and manufacturing, and so it is a big milestone to make robots capable of mastering them.

One company that specializes in making 3D vision systems for industrial robots is Zivid AS. Located in Oslo, Norway and founded in 2015, the company develops hardware and software for 3D vision systems. Their portfolio of cameras include the Zivid One, Zivid One+ and Zivid Two. A picture of Zivid Two, their latest product which was unveiled in 2021, has been provided in Figure 1. All of their cameras are using structured light as a technique for creating the 3D point clouds. This technique will be explained in more detail in Section 2.2. Its main advantages compared to other 3D imaging techniques such as stereovision are that " it has fast measuring speed, high resolution, and high precision" (Association for Advancing Automation 2017).

Unfortunately, structured light systems do not work well in environments consisting of a lot of shiny objects. Shiny objects such as metallic parts and products packaged in plastic wrap are quite common in both factories and warehouses. For that reason

Figure 1: The *Zivid Two* camera mounted on a robot arm (Borgan 2022).

there is an increasing demand to make 3D cameras better at capturing point clouds when such objects are present.

1.2 Previous work

The existing work done in the field of structured light has mainly focused on improving the technique in other fields rather than the interreflection issue. An overview of the existing pattern codification strategies is given in (Salvi et al. 2004). As stated in this overview, existing strategies focus on either short acquisition time, real-time acquisition, or high spatial resolution.

An attempt to address the issue of interreflection has been done in (Harding 2019). The paper focuses on modifying the hardware by applying a polarizing filter in front of the camera. As the author states, this is only applicable to laser-based structured light systems, and thus is not relevant for the portfolio of Zivid cameras. Moreover, the system did not achieve satisfactory results, particularly in the case of multiple reflections.

The application of a dual monocular structured light system in which two cameras are used instead of one has been applied and analyzed in (He et al. 2020). Here, the purpose is to use these two cameras to handle occlusion and high surface reflectance in the presence of shiny objects. The method achieves satisfactory results. However, the work is limited by the fact that it focuses only on isolated shiny objects rather than multiple shiny objects with interreflections occurring between them. The codification strategy used is Gray-Coded Phase Shifts (GCPS). As shown later in Section 3.1.4 the premises for the strategy do not allow for the filtering of such interreflections.

A novel approach to address interreflections was taken in the work by (Lima-Eriksen 2022) in his project thesis. The application of correlation-based patterns significantly reduced the distortions caused by interreflections. While the pattern codification strategies showed promising results, it required the system to be within focus

in order to work. Also, it did not achieve high accuracy due to the discrete nature of the patterns.

1.3 Problem statement

There is an increasing demand for structured light systems that work in the presence of shiny objects in the scene. At the same time, little research has been done to try to address this issue. The correlation-based pattern codification strategies introduced in the project thesis (Lima-Eriksen 2022) showed promising results when it comes to handling interreflections. It is therefore of interest to investigate why this codification strategy sometimes fails, and how it can be modified or extended to mitigate its limitations.

As mentioned in the project thesis, the pattern codification strategies fail outside of focus. Little is known about how patterns are distorted depending on the distance and thereby the focus. Therefore, system identification will be applied to the Zivid Two camera to obtain the point-spread function and the frequency response of the system.

The insights obtained from the system identification will then be applied to investigate the limiting properties of correlation-based patterns. These insights are used to make modifications and extensions to the patterns to address the limitations. Ultimately, the goal is to develop a pattern codification strategy that works in the presence of interreflections at a wide range of distances while still achieving high accuracy. The work will focus on optimizing the performance for a Zivid Two camera. According to Zivid AS, most industrial applications that use a 3D camera require estimate errors less than 0.1% of the camera distance. This corresponds to 0.2 ppx (projector pixels) at the focal distance for the Zivid Two camera, and will be the target accuracy for the pattern codification strategy developed in the thesis.

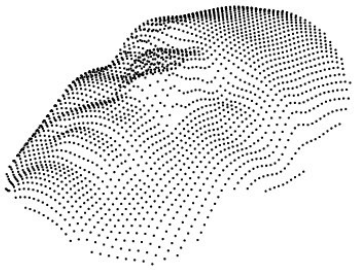
2 Theoretical background

This chapter gives an overview of the theory that is relevant to the work. In the first half, a general structured light system is described from a top-down perspective. An overview of the working principles and models behind structured light is given first. The camera and projector inside the system are then described in a geometric framework using the pinhole model.

The latter half of this chapter introduces common sources of distortions and how they degrade the performance of the system. The first source of distortions originates from the system itself and is caused by the lenses. Following is an overview of reflections from within the scene, which is known as the second source of distortions.

2.1 3D cameras

3D cameras are systems which can create a 3D representation of its field of view. Conventional cameras depict their field of view through a 2D projection in a rectangular grid array of pixels; 3D cameras typically give each pixel a (x, y, z) coordinate specified in spatial coordinates relative to the camera position. These collections of coordinates are known as point clouds. A simple example point cloud that stores only the coordinates of each pixel is given in Figure 2a. For this particular case, it resembles the shape of a face. More advanced 3D cameras, such as the Zivid Two camera, store the values for the RGB color channels for each of the points, and the point clouds become more like a 3D picture. An example of such a point cloud captured by the Zivid Two camera is given in Figure 2b.



(a) Coordinates only (Fabry et al. 2010).

(b) With colors, using Zivid Two (Zivid 2020).

Figure 2: Examples of different types of point clouds.

In contrast to conventional cameras, 3D cameras need a way to know where in space a particular pixel originated from. There are several means of accomplishing this, and most use either time-of-flight or multiview geometry. Time-of-flight-based systems are very fast, but at the expense of resolution. Multiview geometry is the principle used in structured light, and gives very accurate measurements. In its most basic sense, a 3D camera using multiview geometry must be able to see its

field of view from multiple viewpoints, and then find correspondences between them in order to triangulate the XYZ-coordinates of the pixels.

Consider first the case of a single pixel in a camera. As illustrated in Figure 3, there exists one ray that specifies where light could possibly originate. The introduction of a second camera further restricts this. Figure 4 shows the situation in which two cameras point toward an object. Considering *Camera L* isolated, the system only knows a ray from which the light originates for a particular pixel. But with additional knowledge of the ray seen by *Camera R*, only the intersection between these two can be the correct origin. The intersection between two rays forms a point in space, and thus the XYZ coordinates of the corresponding pixel can be found through geometric considerations of this intersection. Thus, the 3D coordinates of the object can be calculated relative to the system by means of triangulation. For this particular example using two cameras, the technique is known as stereo vision and is fairly common in applications (Manuel 2020).

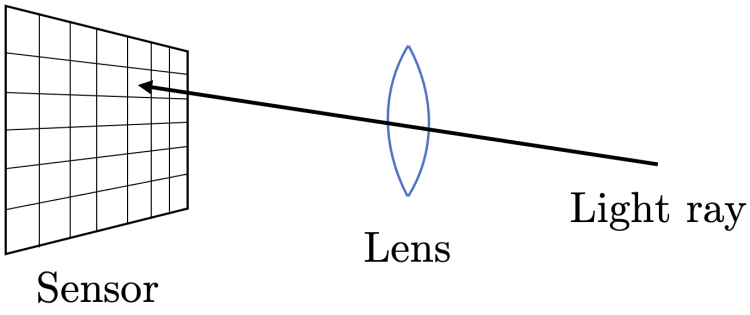


Figure 3: Simplified model of how a light ray impedes onto a camera image sensor.

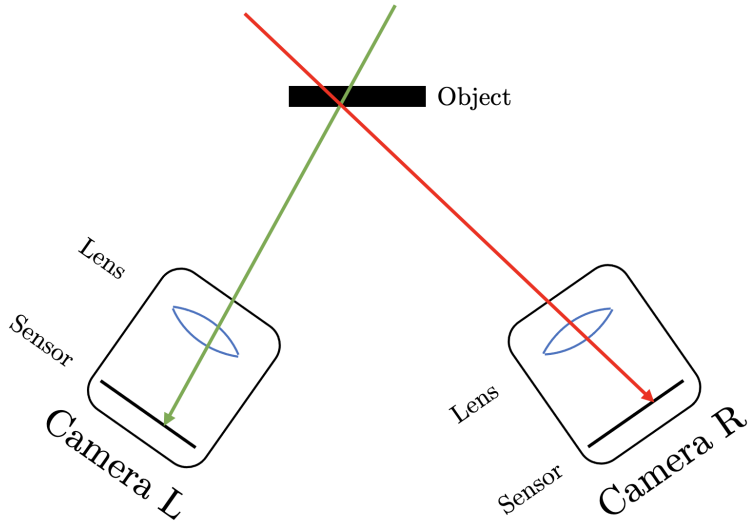


Figure 4: Simplified model of triangulation using multi-view geometry.

2.2 Structured light

The central problem to be solved in 3D cameras is the so-called correspondence problem; To perform the triangulation shown in Figure 4, the system needs to know which pixel in each of the cameras captures the same part of the object. For stereo-vision systems, this would typically involve finding areas in each of the cameras that appear similar. Surfaces that do not have any texture would then pose a challenge, as large areas in the pictures are difficult to distinguish (Szeliski 2011). For instance, consider a scene that consists of a plane that has the same color all over it. Each of the cameras would then capture large areas with the same color. It is impossible to find out which pixels from each of the cameras correspond to each other, as they are all the same color and cannot be distinguished.

Structured light systems address this problem by replacing one of the cameras in Figure 4 with a projector. For each of the pixels in the projector, there will exist a ray pointed out in space that specifies the trajectory of the light originating from the pixel. This is equivalent to the situation depicted in Figure 3 with one of the cameras replaced with a projector, only the direction of the ray from the projector is opposite. Therefore, the correspondence problem now changes to finding out which projector pixel is seen in each of the camera pixels. Since each of the projector pixels can be illuminated independently, the projector is capable of adding textures to objects artificially. This alleviates the problem of surfaces lacking textures. The correspondence problem can be further simplified in the case of structured light systems. Consider the case where the projector is configured in such a way that the intensity of the light that it emits is constant along the y_p direction. Then each projector column $(x_p, \cdot)$ forms a plane in the 3D world, as illustrated by the green triangle in Figure 5. As before, each camera pixel (x_c, y_c) forms a ray in 3D world space, indicated by the red line. The plane and the ray will intersect at a point, making triangulation possible as before. Therefore, the camera only needs to distinguish between all different projector *columns* rather than all projector *pixels*. This means that the structured light system must be able to construct the surjective mapping $(x_c, y_c) \mapsto x_p$, which maps each camera pixel (x_c, y_c) to a projector column x_p. The reader is referred to (Hartley and Zisserman 2003) for a more rigorous introduction to how the multiview triangulation itself is performed. This thesis only considers how to construct the mapping $(x_c, y_c) \mapsto x_p$ without considering the 3D reconstruction.

2.2.1 Patterns and temporal encoding

The correspondence problem is solved by uniquely identifying each projector column x_p for all camera pixels (x_c, y_c). There are primarily two disjoint methods for accomplishing this, and the preferred way depends on the kinetics of the scene itself. First, define a pattern as a matrix $\mathbf{P}$ of dimensions $Y_P \times X_P$ where each entry $(\mathbf{P})_{y_p x_p}$ stores the intensity for a particular projector pixel (x_p, y_p). A scene is said to be dynamic when there are moving objects in the field of view. Similarly, a scene is static if all objects within the field of view are fixed. For dynamic scenes, the projector columns are typically coded spatially (Kawasaki et al. 2009). This means that the

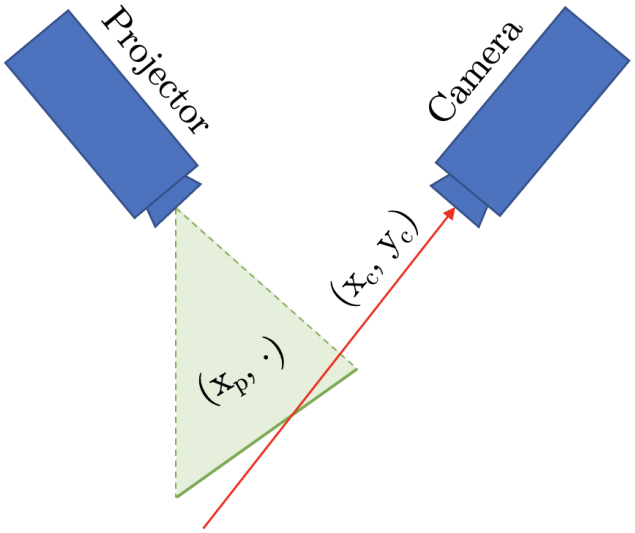
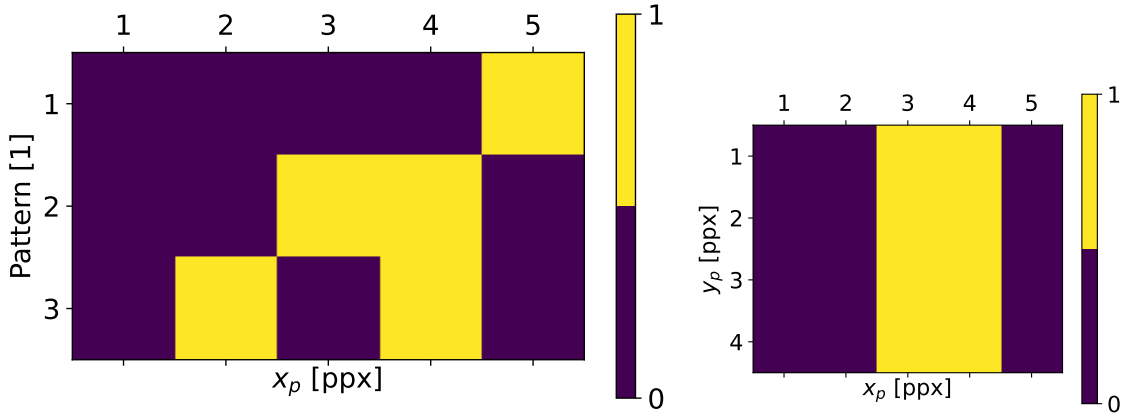


Figure 5: Projector column simplification of structured light systems.

neighborhoods of projector pixels are coded in such a way that the columns can be identified in the camera capture by observing the pixels in close proximity. This pattern codification strategy takes advantage of the fact that pixels that are spatially close in the projector pattern tend to be spatially close in the camera capture as well.

If the scene is static, the camera can take multiple pictures, known as captures, without the scene changing between. Therefore, the system can project multiple patterns and analyze the sequence of captures. This is known as temporal encoding, and now each projector column is uniquely identified by the sequence of intensities captured in the temporal dimension for every camera pixel. The patterns in the pattern sequences are typically constant along the y_p axis, so a pattern sequence can be uniquely identified by the so-called code matrix. This matrix is constructed in such a way that a particular entry $(\mathbf{C})_{ix_p}$ stores the intensity of a projector column x_p for the i-th pattern.

An example of such a matrix is illustrated in Figure 6a. This particular code matrix $\mathbf{C_B}$ contains three patterns corresponding to the rows in the matrix. The second pattern, $(\mathbf{C_B})_2$, is illustrated in Figure 6b with $Y_P = 4\,\text{ppx}$. Consider the projector column $x_p = 3\,\text{ppx}$ in $\mathbf{C_B}$. It is encoded by the intensities $\mathbf{C_B e_3} = \begin{bmatrix}0 & 1 & 0\end{bmatrix}^T$. This particular code matrix uses a binary codification strategy. Thus, by viewing the sequence of patterns as a binary sequence, $\mathbf{C_B e_3}$ would be decoded to 2. If these patterns were used in a structured light system, the camera pixels which captured this sequence would then correspond to the projector column $x_p = 3\,\text{ppx}$. It is apparent that temporal encoding allows for the unique identification of projector columns in the camera only by considering the sequence of intensities captured in each particular camera pixel. This observation will be used extensively throughout the thesis.



(a) Example of a code matrix.

(b) Example of a pattern.

Figure 6: The relationship between a code matrix and its corresponding patterns. The second row in the code matrix corresponds to the pattern illustrated to the right.

2.3 Camera and Projector Geometry

This section describes the geometry of the camera and projector inside the structured light system. The mapping between pixels and world coordinates is derived, which makes it possible to estimate the spatial light trajectories corresponding to each of the pixels. Lastly, the mapping between camera and projector world coordinates is explained. The section is based on the corresponding section from the project thesis (Lima-Eriksen 2022) with minor additions and corrections.

2.3.1 The pinhole model

A camera can be thought of as a projection of the 3D world onto a 2D pixel array known as an image. Similarly, a projector maps a 2D image onto the 3D world. When it comes to structured light systems, it is of interest to be able to estimate these mappings so that correspondences between 2D images and the 3D world can be made. The simplest model which allows such an estimation is called the pinhole model (Moreno and Taubin 2012). It is valid for both cameras and projectors.

Consider the geometry of a general pinhole model depicted in Figure 7. From here on, the subscript $c|p$ indicates that the subscript c should be substituted for the camera pinhole model and the subscript p should be substituted for the projector pinhole model. Assume homogeneous world coordinates $\mathbf{X_{c|p}} = \begin{bmatrix} X_{c|p} & Y_{c|p} & Z_{c|p} & 1 \end{bmatrix}^T$ with origin in $\mathbf{C}_{c|p}$ and homogeneous pinhole coordinates $\mathbf{x_{c|p}} = \begin{bmatrix} x_{c|p} & y_{c|p} & 1 \end{bmatrix}^T$ with origin in $\mathbf{p_{c|p}}$. The world coordinates then describe a point in space relative to the camera / projector origin $\mathbf{C}_{c|p}$, and the pinhole coordinates correspond to a particular pixel in the camera / projector relative to its origin $\mathbf{p_{c|p}}$.

The rightmost part in Figure 7 is the projection of the pinhole model onto the YZ-plane. From the relationships of similar triangles in this projection, it follows that

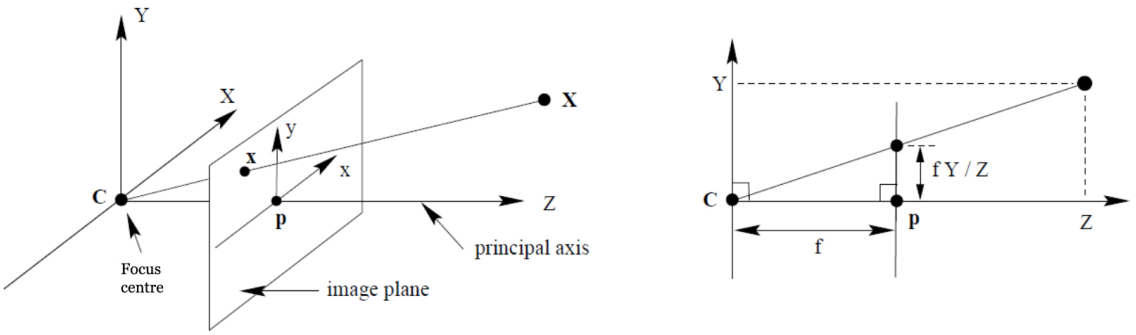


Figure 7: Pinhole model geometry (Hartley and Zisserman 2003).

for a given $Z_{c|p}$,

$$y_{c|p} = \frac{1}{Y_{c|p}} \left(f_{c|p,y} Y_{c|p} + p_{c|p,y} \right)$$

A similar relationship also holds for $y_{c|p}$ by considering the same geometry in the XZ-plane from the leftmost model in Figure 7. The relationships give rise to the mapping $\mathbf{X_{c|p}} \mapsto \mathbf{x_{c|p}}$ through

$$\mathbf{x}_{c|p} = \mathbf{K}_{c|p} \mathbf{X}_{c|p} \tag{1}$$

where

$$\mathbf{K}_{c|p} = \begin{bmatrix} f_{c|p,x} & 0 & p_{c|p,x} & 0 \\ 0 & f_{c|p,y} & p_{c|p,y} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{2}$$

is either the camera matrix or the projector matrix. The terms $p_{c|p,x}$ and $p_{c|p,y}$ account for the fact that the origin of pinhole coordinates is in the upper left corner of the image plane, instead of the center in which the Z_C-axis intersects. The terms $f_{c|p,x}$ and $f_{c|p,y}$ are the horizontal and vertical focal lengths. These numbers are measures of the distance between the lens $\mathbf{p}$ and the camera / projector sensor $\mathbf{C}$.

Unfortunately, neither cameras nor projectors work in this linear fashion in the real world. Lenses are used to solve the issue of getting enough light to pass through the pinhole. But these introduce non-linearities due to geometric distortion known as radial distortion. The effect of these distortions is that straight lines appear curved if the distortions are not corrected for. The experimental setup used in this thesis does not suffer from such distortions, and therefore they will not be considered to keep things simple. In real-world scenarios, the distortions need to be taken into account by introducing the radial distortion vectors $\mathbf{k_{c|p}}$ and including them in the above equations as shown in (OpenCV 2021).

2.3.2 Relative orientation

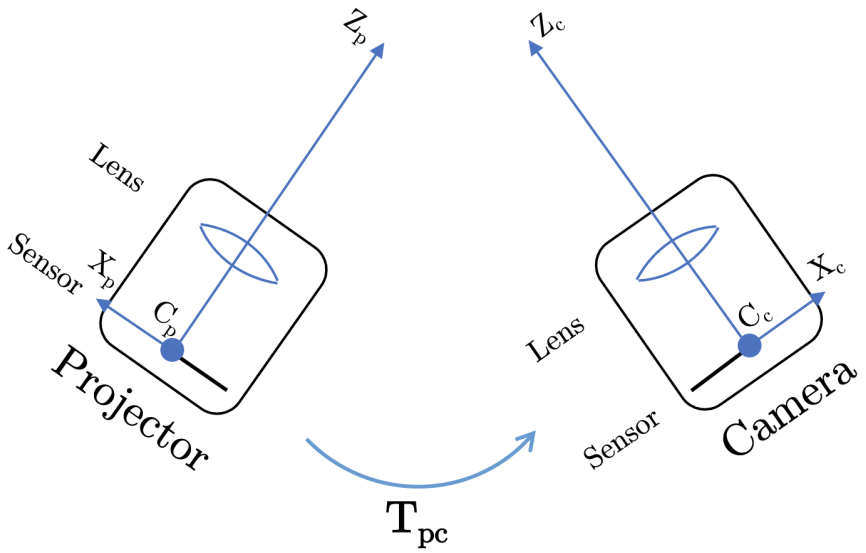


Figure 8: The relative orientation of the camera and projector with their respective world coordinate systems.

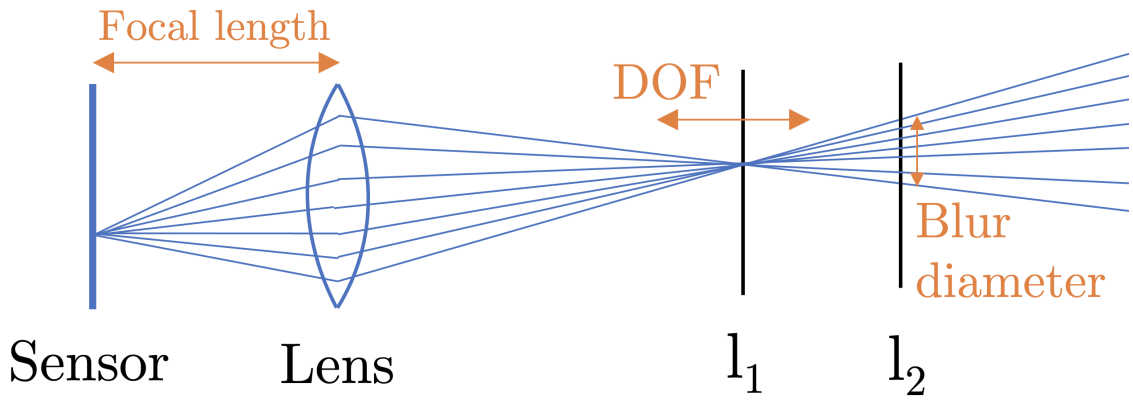
The camera and projector are displaced and rotated relative to each other as described in Section 2.2. Figure 8 shows how the world coordinate systems described in Section 2.3.1 do not have the same origin and basis. This comes from the fact that the origins of these coordinate systems are placed in the sensor centers with their respective $Z_{c|p}$-axis pointing towards the lens center. Let $\mathbf{T}_{pc}$ denote the homogeneous combined translation and rotation matrix from the projector world coordinate system to the camera world coordinate system. Then

$$\mathbf{X}_c = \mathbf{T}_{pc}\mathbf{X}_p \tag{3}$$

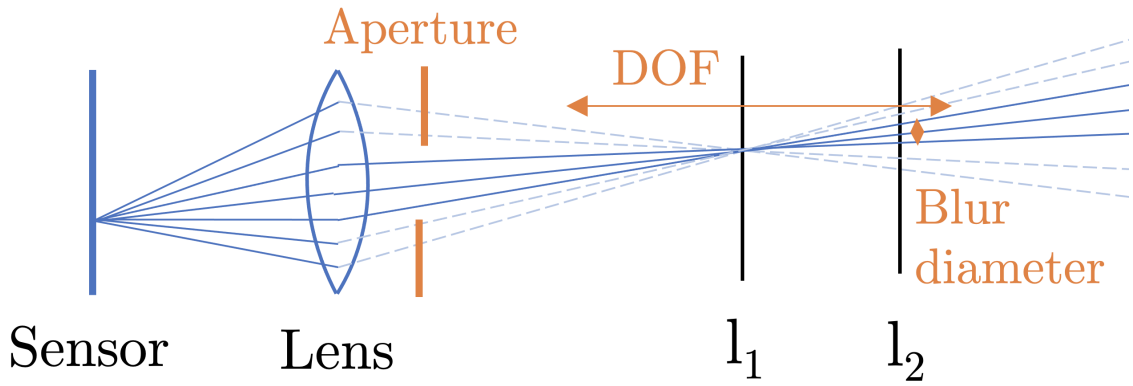
From the above equation it is possible to map a point from the projector world coordinate system to camera world coordinate system and vice versa. This closes the gap between the projector and camera, and allows for mapping a projector pixel to a camera pixel given a certain distance. The geometric considerations will later be exploited in Chapter 5 in order to improve on the performance of structured light systems.

2.4 Lens optics

The pinhole model stated above gives a good model on the mapping between the 3D world coordinates and 2D image coordinates in cameras and projectors. Nevertheless, it is a simplification of what happens and lacks the ability to capture the spatial distortions caused by lenses.



(a) Without aperture.



(b) With aperture.

Figure 9: Model of how a lens focuses light in cameras and projectors at various distances.

To focus light from the outside world onto the sensor in a projector or camera, a convex lens is placed in front of the sensor (Fiete 2010). Such a lens makes light rays converge towards the center of the lens. Typically, a camera or projector will utilize a combination of convex and concave lenses to get optimal optical properties. In order to keep things simple, a single convex lens will be used here, as the principles stay the same. Also, the principles will be explained for a camera, but the same arguments will also apply to a projector.

Figure 9a depicts an imaging sensor, a lens, and two possible objects l_1 and l_2 placed in its field of view at different distances. Due to the bending of light rays caused by the lens, there are multiple ways a light can travel through the lens and be incident onto the same pixel. In the figure, these possible paths are illustrated by the blue lines for one particular pixel. The optical properties of the lens as well as the focal length specifies a distance to which all these lenses cross each other in a single point, and this happens in what is known as the *focusing distance*. For this example, it happens at the distance in which object l_1 is placed. Notice that at this distance, a single pixel will have its incident light rays from a single point in space only. This in turn makes an image sharp.

Objects may also be placed at distances different from the focusing distance and l_2 is an example of this. Here, the light incident to a particular pixel will originate from an area of circular shape on l_2. Therefore, the light that is captured by this pixel will be a weighted average of the light within this circle. Depending on the relative distance of l_2 from the focusing distance, the diameter of the circle will change as indicated by the figure. The phenomenon of this weighted average over a circle is known as blur, since it smoothes out the image. The diameter of the circle is known as the blur diameter. A scene captured by a camera will never have all of its objects within the focusing distance, and so many objects will only be slightly blurred. The distance between the nearest and furthest objects in acceptably sharp focus is known as Depth of Field (DOF).

For the model depicted in Figure 9a, the DOF is fixed by the optical properties of the lens as well as the focal length, meaning that it cannot be adjusted once the camera is constructed. There is however another way of modifying the DOF. Figure 9b introduces what is known as an *aperture*. This is an adjustable circular opening of the lens that restricts the light rays incident to the lens. It is measured in f-number, where an increase in number corresponds to a narrower aperture. For the aperture depicted in this particular figure, the two uppermost and lowermost light rays cannot pass through the aperture, indicated by the dashed lines. This effectively reduces the blur diameter outside of focus, as seen, e.g., for the object l_2. With a reduced blur diameter, the DOF increases. As a result, objects at a wider range of distances from the focusing distance will appear sharp. The aperture is often adjustable through a mechanism known as *iris*. Decreasing the aperture increases the DOF, but it comes with its drawbacks. Since less light is allowed through the lens, the exposure time must be increased to compensate for this. Exposure time is the time in which light is allowed onto the sensor to form an image. If objects move during the exposure, motion blur will be present, leading to an unsharp image. Also, the increase in DOF with decreasing aperture will not continue indefinitely, as other optical phenomena will occur (Conrad 2018).

The choice of aperture ultimately becomes a compromise between having a large DOF and a short exposure time. This compromise means that not all blurring can be avoided. The physical effects of this blur are quite complex to model, as it depends on multiple optical aspects. However, a common simplification is to approximate the blur as a convolution of a perfectly sharp image with a Gaussian kernel with a certain SD to obtain the blurred image (Strasburger et al. n.d.).

From the signal processing domain, convolution with a Gaussian kernel is known as a low-pass filter. Therefore, high frequency content will be dampened due to this effect. In structured light, one wants to identify the mapping from projector pixels to camera pixels in order to exploit multiview geometry. If the light in the projector is coded with high frequency content, the details may be lost due to this blurring effect, and caution has to be taken when designing the patterns to mitigate the issue.

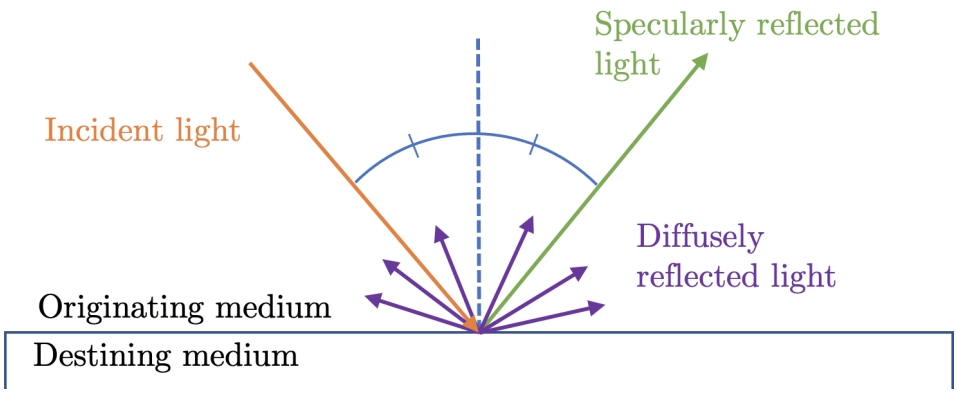


Figure 10: Illustration of the different types of physical reflections. Based on an illustration from Sergiyenko 2010.

2.5 Reflections

Reflection is the phenomenon in which parts of a light ray bounce off the incidence between two different media, and return to the originating medium (Lekner 1987). This section first describes reflections from a purely physical perspective, where they are separated on the basis of how they spread back into the originating medium. In the latter half, reflections are described from the perspective of a structured light system in terms of how they cause distortions.

2.5.1 Diffuse and specular reflections

When a light ray enters the incidence between two media, a fraction of the wave will be reflected back into the originating medium. Depending on the destining medium, the light ray may be reflected in different ways.

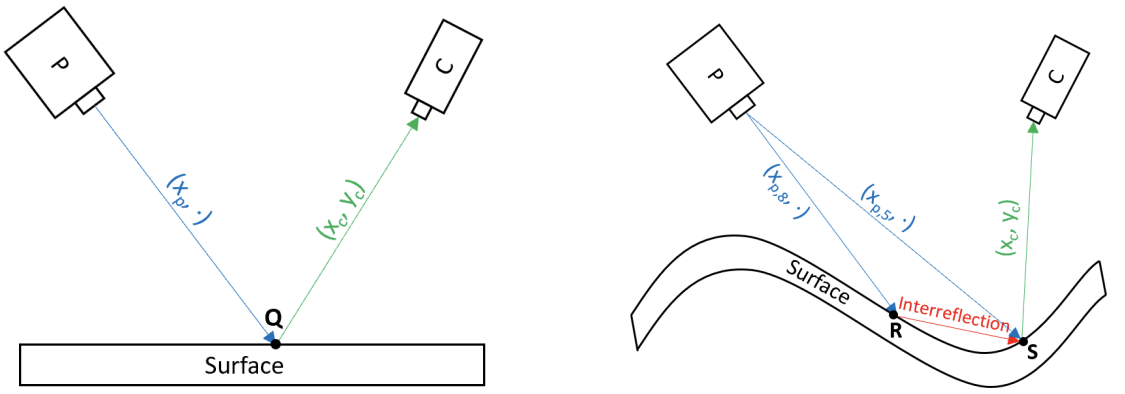
For some destining media such as polished metal, glass, or transparent plastics, the reflections will be mostly specular. Such reflections behave in a mirror-like fashion, meaning that all of the light is reflected back into the originating medium at the angle of incidence in accordance with Snell's law of refraction (Steyerl et al. 1991). The reflected light ray will have the same intensity as the originating ray because all of the reflected light travels in the same direction. This is illustrated in Figure 10, where the incident light ray (orange) is reflected back as a single ray (green).

A different phenomenon occurs when the destining medium is e.g. unfinished woods or paper. For such surfaces which are close to Lambertian, the light rays will be scattered away from the destining medium in a range of different angles (Lu 2016). The intensity of the reflected light will be of a lower intensity than the incident light, as it is spread of a larger area. This type of reflection corresponds to the purple rays in Figure 10.

In reality, surfaces will reflect light as a mixture of both diffuse and specular reflections, whereby the ratio between these two is subject to variation.

2.5.2 Direct reflections and interreflections

Structured light systems are based upon the principle of determining the correct projector column observed in each of the camera pixels. When light from such a projector column hits a surface, it is known as a reflection. Therefore, it is essential to have knowledge on how the nature of these reflections affect the performance of the system. From a structured light perspective, reflections are typically divided into direct reflections and interreflections. The distinction is made based on how many reflections occur from the light exiting the projector till it enters the camera.



(a) Direct reflection.

(b) Direct reflection and interreflection both entering the same (x_c, y_c).

Figure 11: The geometry of reflections in a structured light system. Camera: **C**; Projector: **P**. Courtesy of Lima-Eriksen 2022.

Direct reflections occur when the incident light (x_p, y_p) bounces off of a surface in a point **Q** and enters the camera directly in (x_c, y_c) without any further bounces (Deeb et al. 2017) as illustrated in Figure 11a. Since a single projector pixel (x_p, y_p) is captured by a particular camera pixel (x_c, y_c), there exists an injective mapping $(x_c, y_c) \mapsto x_p$. This is the premise behind structured light systems as stated in Section 2.2.

Things get more complicated in Figure 11b. Here, light rays from both $(x_{p,5}, \cdot)$ and $(x_{p,8}, \cdot)$ enters the camera in (x_c, y_c). Typically, this phenomenon occurs when the surface gives off a lot of specular reflections. A large proportion of the light reflected from $(x_{p,8}, \cdot)$ in **R** is reflected specularly to the point **S** through what is called an interreflection. This is the same point that the ray from $(x_{p,5}, \cdot)$ hits directly, and the light rays give off diffuse reflections in this point. Therefore, both $(x_{p,5}, \cdot)$ and $(x_{p,8}, \cdot)$ is seen by the camera in (x_c, y_c). As the mapping $(x_c, y_c) \mapsto x_p$ now is non-injective, decoding errors may occur. If the surface is of such shape that the light ray from $(x_{p,5}, \cdot)$ is partially occluded, the light ray from $(x_{p,8}, \cdot)$ might be the one with the highest intensity when captured in (x_c, y_c).

3 State of the Art

This chapter presents the state of the art in structured light from the perspective of signal processing. Gray-Coded Phase Shifts (GCPS) is a commonly used pattern codification strategy for static scenes due to its high accuracy. The codification strategy will be presented using a top-down perspective together with its strengths and weaknesses. Lastly, a method of filtering out invalid solutions are presented. The algorithm is particularly useful for pattern codification strategies which give multiple possible solutions to the mapping $(x_c, y_c) \mapsto x_p$.

3.1 Gray-Coded Phase Shifts

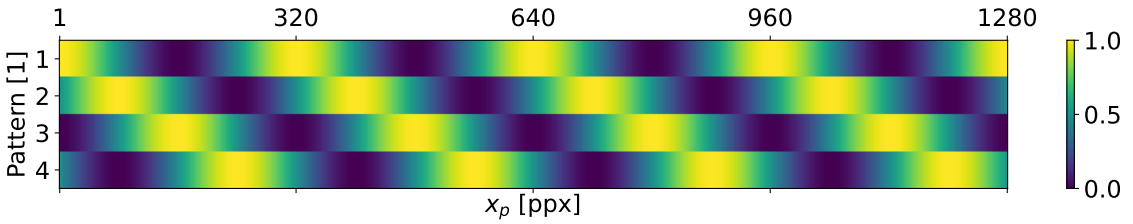
Gray-Coded Phase Shifts is a temporal pattern codification strategy, using a hierarchical column estimate. It consists of two types of patterns – phase shifts and gray codes. Phase shifts give high-accuracy estimates of the originating projector columns, but the estimates are periodic. To unwrap the periodicity of phase shifts, temporal gray codes are used. The combination of these two pattern types make a fast pattern codification strategy with high accuracy.

3.1.1 Working principles

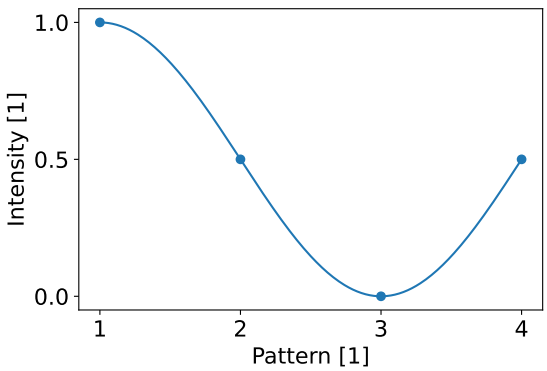
The first part of GCPS is the phase shifts. It consists of projecting four patterns with horizontal cosines, each with its phase shifted 90° relative to the previous. The code matrix $\mathbf{C_{PS(320)}}$ that belongs to the phase shift patterns with spatial period $W_F = 320\,\text{ppx}$ and $X_P = 1280\,\text{ppx}$ is shown in Figure 12a. Recall that the blur effects from defocus correspond to a convolution with a Gaussian kernel. This is a low-pass filter, so the overall shape of these cosines will be preserved at a wide range of camera distances Z_C.

An important observation from these phase-shifted cosines is the fact that the patterns resemble cosines in the *temporal* domain as well. To illustrate this, the samples for $x_p = 320\,\text{ppx}$ are available from the code matrix through $\mathbf{C_{PS(320)}} \cdot \mathbf{e_{320}}$, and are plotted as the dots in Figure 12b. The blue curve in this same figure is a cosine with its phase equal to the phase of the first sample when viewed as part of the cosine in the *spatial* domain of the first pattern. By estimating the phase of the cosines in the temporal domain, one gets the spatial phase of the first sample when viewing it as part of a cosine in the *spatial* domain.

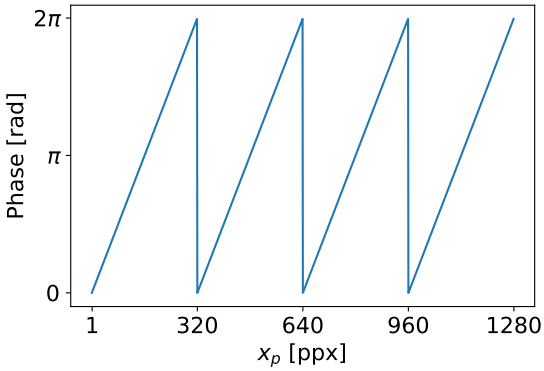
The phase of the temporal cosines can be found for each of the projector columns, and the result for this example is shown in Figure 12b. The phase can be used to identify each projector column through its phase shift. Unfortunately, the estimate is periodic with W_F due to the periodicity of arctan2. This in turn will lead to an ambiguity, since, for instance, the phase in $x_p = 200\,\text{ppx}$ is the same as in $x_p = 200\,\text{ppx} + W_F$. Decoding structured light requires the mapping $(x_c, y_c) \mapsto x_p$ be one-to-one, which means that these patterns cannot be used alone.



(a) $\mathbf{C_{CP(320)}}$



(b) $\mathbf{C_{CP(320)}} \cdot \mathbf{e_{320}}$



(c) ϕ

Figure 12: Phase stepping with $X_P = 1280\,\text{ppx}$ and $W_F = 320\,\text{ppx}$.

To solve the ambiguity in the projector column estimates of phase shifts, temporal gray codes are used. First, the x_p-axis of the projector is divided into sectors of equal width W_F called *fringes*. This means that exactly one period of the cosines in phase shifts fits within one such fringe. The purpose of the gray codes is to uniquely identify the originating fringe of each projector column. This is illustrated in Figure 13. Here, groups of four projector columns are identified as the same fringe using gray codes. Within each of these fringes, phase shifts make it possible to distinguish between the projector columns. Together, they allow for the unique identification of any projector column.

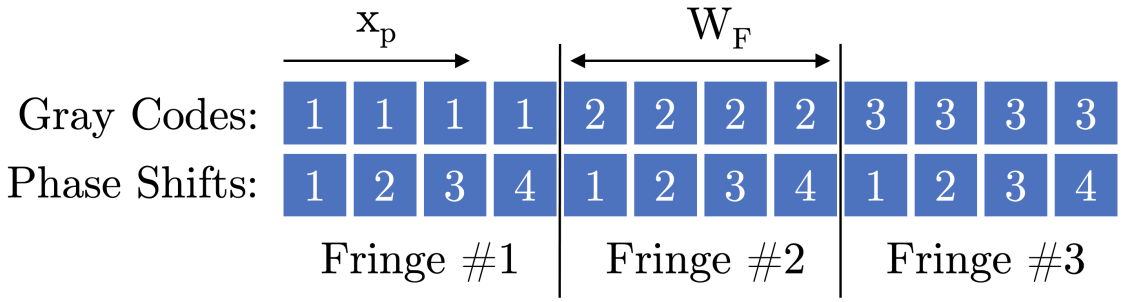
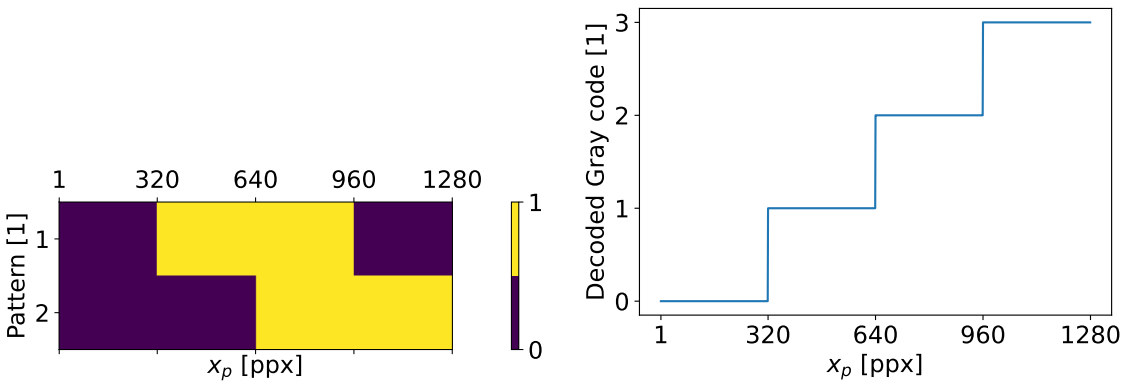


Figure 13: In GCPS, the projector columns are sectioned into fringes of width W_F.

Gray codes are binary codes that are constructed in such a way that only one bit changes when going from one code word to the next. They can be derived from its binary representation through the algorithm specified in (Gray 1947). For instance, the binary code $\{0, 1\}$ can be written as $\{1, 0\}$ in gray code representation . In GCPS, each fringe is given its unique gray code corresponding to its fringe



(a) The code matrix for the gray codes.

(b) The decoded gray codes for each x_p.

Figure 14: Gray codes with $X_P = 1280\,\text{px}$ and $W_F = 320\,\text{px}$.

number. This means that the first fringe is coded as the gray code representation for 0, the second fringe is coded as the gray code representation for 1, etc. In total $N_F = \lceil \frac{X_P}{W_F} \rceil$ fringes should be uniquely identified, requiring $N_{P,GC} = \lceil \log_2(\frac{X_P}{W_F}) \rceil$ bits to distinguish them. For this particular example, two bits are required. Gray codes are coded in the temporal dimension, so one pattern is needed per bit. The code matrix required for this example is shown in Figure 14a.

By decoding the gray codes for every projector column, one gets the result shown in Figure 14b. Notice that it is constant within each fringe of size $W_F = 320\,\text{ppx}$, as desired. The decoded phase shifts and gray codes can be scaled and combined to form the mapping $(x_c, y_c) \mapsto x_p$, and this will be shown in the next section.

3.1.2 Algorithm

As seen above, GCPS consists of multiple steps which together allow for the unique identification of projector columns for every camera pixel. The following section will explain the pattern codification strategy from an algorithmic perspective. It is in part based on the corresponding section found in the project thesis (Lima-Eriksen 2022), but it has been rewritten to use tensors and matrices in order to follow the same general format as for the other patterns presented in this thesis.

3.1.2.1 Normalization

In most scenes, there is a certain background lighting originating from either the sun or artificial lighting sources such as lamps. When a capture is made of a fully dark pattern, this results in each camera pixel (x_c, y_c) having a certain background illumination $I_0(x_c, y_c)$ instead of being fully black, as expected. In addition to this, objects have varying reflectance as further described in Section 2.5. Therefore, the total illumination of one particular camera pixel $(x_{c,1}, y_{c,1})$ will result in a different intensity captured compared to another camera pixel $(x_{c,2}, y_{c,2})$. For the algorithm to be able to correctly decode the captures, it is essential that the effects originating from background illumination and reflectance are eliminated. This is done through a

process known as normalization. In essence, it results in each camera pixel spanning the range $[0, 1]$ such that 0 corresponds to the intensity of no illumination from the projector and 1 corresponds to maximum illumination from the projector.

Let the reflectance observed in (x_c, y_c) be denoted as $R(x_c, y_c)$. If the camera pixel (x_c, y_c) is illuminated by the projector with a particular intensity $I_{cp}(x_c, y_c)$, then according to (Skotheim and Couwelleers 2004) the intensity captured by the camera can be represented as

$$I_c(x_c, y_c) = I_0(x_c, y_c)\left[1 + R(x_c, y_c) \cdot I_{cp}(x_c, y_c)\right] \tag{4}$$

When capturing a certain pattern, the values of $I_c(x_c, y_c)$ is what is obtained. However, the intensity $I_{cp}(x_c, y_c)$ that originates from the projector is of interest. To obtain $I_{cp}(\cdot)$, one needs to estimate $I_0(\cdot)$ and $R(\cdot)$. This is done by first capturing an image I_{c0} where $I_{cp}(x_c, y_c) = 0 \,\forall\, (x_c, y_c)$. In other words, an all-black pattern is used. Then (4) simplifies to the estimate

$$\widehat{I_0}(x_c, y_c) = I_{c0}(x_c, y_c) \tag{5}$$

For estimating $R(\cdot)$, a capture I_{c1} of the pattern $I_{cp}(x_c, y_c) = 1 \,\forall\, (x_c, y_c)$ is made. The estimate $\widehat{I_0}$ from (5) is inserted into (4), and the equation is solved for $R(\cdot)$:

$$\begin{aligned} I_{c1}(x_c, y_c) &= \widehat{I_0}(x_c, y_c)\left[1 + \widehat{R}(x_c, y_c) \cdot 1\right] \\ \widehat{R}(x_c, y_c) &= \frac{I_{c1}(x_c, y_c)}{\widehat{I_0}(x_c, y_c)} - 1 \end{aligned} \tag{6}$$

With the estimates $\widehat{I_0}(x_c, y_c)$ and $\widehat{R}(x_c, y_c)$ known, an estimate of $I_{cp}(\cdot)$ given $I_c(\cdot)$ can be made through the transformation

$$\widehat{I_{cp}}(x_c, y_c) = \frac{I_c(x_c, y_c) - \widehat{I_0}(x_c, y_c)}{\widehat{I_0}(x_c, y_c)\widehat{R}(x_c, y_c)} \tag{7}$$

It is shown through (7) that the projected light intensity in a certain camera pixel (x_c, y_c) can be reconstructed. This transformation is used later in the GCPS algorithm to estimate which part of the projected patterns is observed in the camera pixels.

3.1.2.2 Phase shifts

In this next step, the scene is illuminated with a series of horizontal cosine patterns, each with its phase shifted 90° relative to the previous. Let W_F denote the spatial period of the cosines. Also, define the phase shift vector

$$\mathbf{\Phi_N} = \begin{bmatrix} 0 \cdot \frac{2\pi}{N} & 1 \cdot \frac{2\pi}{N} & \cdots & (N-1) \cdot \frac{2\pi}{N} \end{bmatrix}^T \tag{8}$$

Then these patterns can be represented by its code matrix of dimensions $4 \times X_P$, where each element is defined through

$$\left(\mathbf{C_{CP(W_F)}}\right)_{nx_p} = \frac{1}{2}\left[1 + \cos\left(\frac{2\pi}{W_F}x_p - \left(\mathbf{\Phi_4}\right)_n\right)\right] \tag{9}$$

An example of $\mathbf{C_{CP(W_F)}}$ with $X_P = 1280\,\text{ppx}$ and $W_F = 320\,\text{ppx}$ was given in Figure 12a. The phase for each projector column can be calculated (Hung 2000) according to

$$\left(\phi\right)_{x_p} = \text{arctan2}\left\{\cos \mathbf{\Phi_4^T} \cdot \mathbf{C_{CP(W_F)}} \cdot \mathbf{e_{x_p}}, \quad \sin \mathbf{\Phi_4^T} \cdot \mathbf{C_{CP(W_F)}} \cdot \mathbf{e_{x_p}}\right\}$$

This would correspond to the phase plot in Figure 12c. Of course, the phases of the cosines have to be estimated from the camera captures. Let the tensor $\mathbf{M_{PH}}$ of dimensions $Y_C \times X_C \times 4$ store the captures of the phase shift patterns for each camera pixel (x_c, y_c) as $(\mathbf{M_{PH}})_{y_c x_c}$, normalized according to the normalization algorithm provided above. Then the phase can be estimated similarly for each camera pixel and stored in the $Y_C \times X_C$ matrix through

$$\left(\mathbf{Q_{PH}}\right)_{y_c x_c} = \text{arctan2}\left\{\cos \mathbf{\Phi_4^T} \cdot \left(\mathbf{M_{PH}}\right)_{y_c x_c}, \quad \sin \mathbf{\Phi_4^T} \cdot \left(\mathbf{M_{PH}}\right)_{y_c x_c}\right\} \tag{10}$$

3.1.2.3 Gray-coded fringes

In addition to the phase shift patterns, gray code patterns must be made. Let first $\text{bin}_N(n)$ be a function which takes a decimal number n and returns a vector of dimensions $N \times 1$ which is the binary representation of the number. As an example, $\text{bin}_4(5) = \begin{bmatrix} 0 & 1 & 0 & 1 \end{bmatrix}^T$. For a certain fringe width W_F and number of projector columns X_P, $N_{P,GC} = \lceil \log_2 \frac{X_P}{W_F} \rceil$ patterns are needed for the gray codes. Initialize the binary code matrix $\mathbf{C_{B(W_F)}}$ of dimensions $N_{P,GC} \times X_P$ as

$$\mathbf{C_{B(W_F)}} \cdot \mathbf{e_{x_p}} = \text{bin}_{N_{P,GC}}\left(x_p \bmod W_F\right)$$

Now, each column vector of $\mathbf{C_{B(W_F)}}$ is the binary representation of its fringe number. For instance, the first W_F column vectors are the binary representation of 0, etc. According to the conversion algorithm provided in (Gray 1947), this code matrix can be converted to its gray code representation using

$$\left(\mathbf{C_{GC(W_F)}}\right)_{nx_p} = \left(\mathbf{C_{B(W_F)}}\right)_{nx_p} \oplus \left(\mathbf{C_{B(W_F)}}\right)_{(n-1)x_p} \tag{11}$$

with the special case

$$\left(\mathbf{C_{GC(W_F)}}\right)_{1x_p} = \left(\mathbf{C_{B(W_F)}}\right)_{1x_p}$$

An example of such a code matrix with $W_F = 320\,\text{ppx}$ and $X_P = 1280\,\text{ppx}$ was provided in Figure 14b.

Let the tensor $\mathbf{M_{GC}}$ of dimensions $Y_C \times X_C \times N_{P,GC}$ store the captures of the gray code patterns, normalized according to the normalization algorithm given earlier. These captures store normalized light intensities in the range $[0, 1]$, whereas the codes are binary. In order to decode these captures to their corresponding fringe number, they should first be discretized to binary numbers. This is done through

$$\mathbf{M'_{GC}} = \lfloor \mathbf{M_{GC}} \rceil$$

which rounds each light intensity to the nearest integer (0 or 1). The captures can be converted from gray codes to binary codes using

$$\left(\mathbf{Q_B}\right)_{y_c x_c n} = \left(\mathbf{M'_{GC}}\right)_{y_c x_c 1} \oplus \left(\mathbf{M'_{GC}}\right)_{y_c x_c 2} \oplus \cdots \oplus \left(\mathbf{M'_{GC}}\right)_{y_c x_c (n-1)}$$

where the binary representations of the captures are stored in the tensor $\mathbf{Q_B}$ of dimensions $Y_C \times X_C \times N_{P,GC}$. Let the matrix $\mathbf{Q_{GC}}$ of dimensions $Y_C \times X_C$ store the decimal representation of the decoded gray codes. Converting from binary to decimal is trivial:

$$\left(\mathbf{Q_{GC}}\right)_{y_c x_c} = \left(\mathbf{Q_B}\right)_{y_c x_c 1} \cdot 2^{N_{P,GC}-1} + \left(\mathbf{Q_B}\right)_{y_c x_c 2} \cdot 2^{N_{P,GC}-2} + \cdots + \left(\mathbf{Q_B}\right)_{y_c x_c N_{P,GC}} \cdot 2^0$$

3.1.2.4 Phase unwrapping

The gray code solutions and the phase shift solutions should finally be combined to find the mapping $(x_c, y_c \mapsto x_p$. Phase shifts yield solutions between 0 and 2π, and can identify projector columns periodic to W_F. On the other hand, the gray codes decode the fringe number for each camera pixel. These can be combined by scaling the estimates and adding them, and the total solution is available through the $Y_C \times X_C$ matrix, where

$$\left(\mathbf{Q_{GCPS}}\right)_{y_c x_c} = \frac{W_F}{2\pi} \cdot \left(\mathbf{Q_{PS}}\right)_{y_c x_c} + W_F \cdot \left(\mathbf{Q_{GC}}\right)_{y_c x_c} + 1 \tag{12}$$

The +1 in the above equation comes from the fact that pixels start at 1.

3.1.3 Strengths

A major strength of GCPS is that it allows for a great accuracy in estimating the mapping $(x_c, y_c) \mapsto y_p$. The phase stepping increases in accuracy with decreasing

W_F (Salvi et al. 2004). Also, the phase stepping is quite resilient to the blur caused by lens defocus mentioned in Section 2.4, as the defocus behaves as a low-pass filter and thus only reduces the overall amplitude of the cosines. Some defocus on the projector lens is in fact advantageous because it blurs out the cosine and makes it appear continuous. Then the phase unwrapping allows for sub-pixel accuracy. On the other hand, the gray codes are a lot more susceptible to errors in the presence of blurring artifacts. In practice, a compromise has to be made which allows for high accuracy in phase stepping while still correctly decoding the grade codes when the lens is somewhat defocused.

Another advantage of GCPS is that it is really fast and scales well. As mentioned previously, the $N_F = \lceil \frac{X_P}{W_F} \rceil$ fringes can be coded with $\log_2 N_F$ patterns. Adding one pattern thus doubles the number of fringes that can be encoded. Normalization and phase shifts will always require only two and four captures, respectively. The whole decoding pipeline is parallelizable for each camera pixel (x_c, y_c) as shown by the equations, so the process of constructing the point cloud is fast.

3.1.4 Weaknesses

The GCPS pattern codification strategy does unfortunately only work correctly when the surfaces of the objects in the scene mostly give off scattering reflections. Both the gray code patterns and the phase shifts collapse in the presence of specular reflections because they give rise to interreflections as seen in Section 2.5.

The gray codes utilize the whole code space, meaning that the lowest hamming distance for all the code words to another valid code word is always one. Therefore, erroneously decoding only a single bit can lead to errors of integer multiples of W_F, which is significant. Such errors typically give rise to shadow planes and make the point clouds useless in most cases.

The decoding error introduced in phase shifts is easily shown by noting that the sum of two cosines with the same frequency ω is another cosine with the same frequency but a new phase shift. For simplicity, let the correct signal be denoted by $s(t) = A\cos(\omega t - \alpha)$. If an interreflection causes part of the pattern with a different phase shift $w(t) = B\cos(\omega t - \beta)$ to add to $s(t)$, then the sampled signal $i(t)$ will be a new cosine with the same frequency but a different phase shift:

$$\begin{aligned} i(t) &= s(t) + w(t) \\ &= A\cos(\omega t - \alpha) + B\cos(\omega t - \beta) \\ &= C\cos(\omega t - \zeta) \text{ , where } \begin{cases} C = \sqrt{[A\cos(\alpha) + B\cos(\beta)]^2 + [A\sin(\alpha) + B\sin(\beta)]^2} \\ \zeta = \arctan\left[\frac{A\sin\alpha + B\sin\beta}{A\cos\alpha + B\cos\beta}\right] \end{cases} \end{aligned} \tag{13}$$

The phase shift $\zeta \neq \alpha$ is perfectly valid and it is impossible to distinguish $s(t)$ from $w(t)$. Of course, following from (12) the errors $\epsilon_{ph} = \frac{W_F}{2\pi} \cdot [\alpha - \zeta] \in \left[-\frac{W_F}{2}, \frac{W_F}{2}\right)$, which

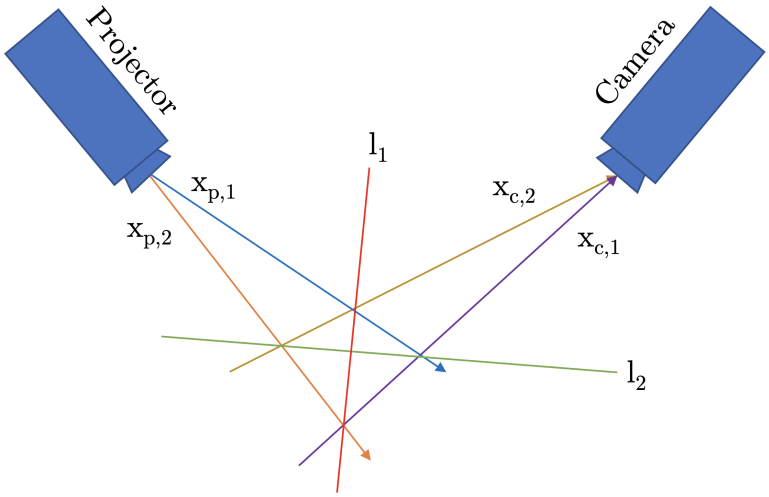


Figure 15: Physical interpretation of positive and negative gradient in projector column mapping.

are always of lower magnitude than the ones introduced by the gray code patterns. But errors in phase shifts can occur even for low magnitudes B of the noise. This is in contrast to gray codes, which require at least one bit in the code word to be decoded erroneously. For instance, if a zero is to be decoded, but interreflections increase the light intensity in the pixel from 0.1 to 0.4, the pixel is still decoded as a zero.

3.2 Gradient filter

The multiview geometry which is utilized in structured light places several restrictions on the mapping $(x_c, y_c) \mapsto x_p$, and one of them is a restriction in the projector column gradient. It will be shown that this constraint allows for filtering out invalid solutions, which typically originate from interreflections.

Consider the two-dimensional model of a structured light system depicted in Figure 15. The model is viewed from the top down. The columns in both the camera and the projector increase when looking from the sensor through the lens left-to-right. For this model, only two columns are depicted in the camera ($x_{c,1}$ and $x_{c,2}$) and the projector ($x_{p,1}$ and $x_{p,1}$). Assume that the scene captured by the system is constructed in such a way that each of the camera columns see either $x_{p,1}$ or $x_{p,2}$. Then there are four possible pairs of camera-to-projector mappings: $(x_{c,1} \mapsto x_{p,1},\ x_{c,2} \mapsto x_{p,1})$, $(x_{c,1} \mapsto x_{p,2},\ x_{c,2} \mapsto x_{p,2})$, $(x_{c,1} \mapsto x_{p,1},\ x_{c,2} \mapsto x_{p,2})$ or $(x_{c,1} \mapsto x_{p,2},\ x_{c,2} \mapsto x_{p,1})$.

The first two pairs of mappings would indicate that the scene depicts an object which tangents the rays from either $x_{p,1}$ or $x_{p,2}$ respectively, and are trivial. The third mapping corresponds to an object having the surface of l_2, which is perfectly possible from a physical perspective. For the last mapping, it would require the surface of l_1 to be causing the reflection. As the model shows, the situation depicted is not a reflection at all, as l_1 blocks the light rays from entering the camera. The situation is contradicting, and not physically possible.

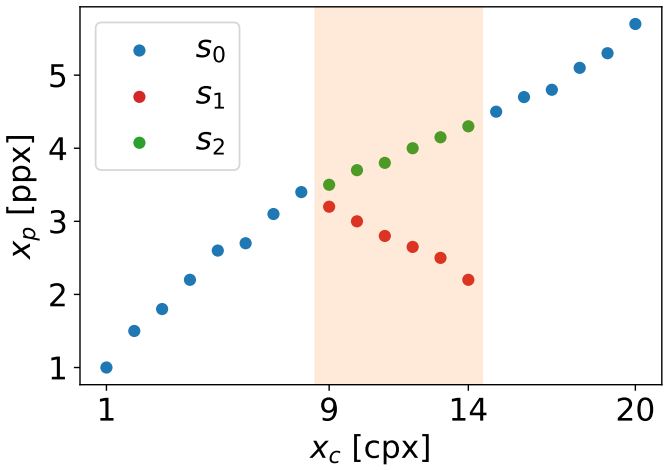


Figure 16: Projector column mapping with valid (s_2) and invalid (s_1) solution.

From a mathematical perspective, let the mapping be depicted as projector columns being a function of camera columns. Then the pair ($x_{c,1} \mapsto x_{p,1}$, $x_{c,2} \mapsto x_{p,2}$) corresponds to the function having a positive gradient and the pair ($x_{c,1} \mapsto x_{p,2}$, $x_{c,2} \mapsto x_{p,1}$) would correspond to a negative gradient. Thus, a negative gradient is invalid due to geometric considerations.

Next, consider the camera column to projector column mappings depicted in Figure 16. First, let s_0 and s_1 be the mappings obtained using a structured light system. The points s_1 have a negative gradient, and are physically impossible. Therefore, they should be considered invalid and removed in further processing. This is known as applying a gradient filter. Typically such negative gradients will originate from interreflections, as these will invert the order in which projector columns appear in a capture. For some pattern codification strategies, the decoding stage might give multiple possible solutions s_1 and s_2. In that case, s_2 would be considered the correct solution.

Although a gradient filter appears simple through its derivation, its practical application is significantly more difficult. Detecting such negative gradients requires application-specific tuning, and using the technique will be out of the scope of this thesis. It is included merely to substantiate the usefulness of pattern codification strategies which give multiple solutions.

4 Materials and Method

Much of the work in this thesis revolves around the analysis of a structured light system, which is inherently nonlinear. Later it will also be shown that some of the patterns themselves are nonlinear. For these reasons, direct analytical approaches are not feasible in most cases. Instead, experiments are developed to give insight into performance. These will be performed using rendering software.

This chapter serves as an overview of the materials and method applied in the work of the thesis. First, the experimental setup will be presented with an introduction to how the rendering software works. Next, the test scenes used to render the patterns are shown. The latter part defines metrics and benchmarks used to quantify the performance of the test results.

4.1 Experimental setup

There are mainly two ways of performing experiments with structured light systems. Either a physical system can be used or the system can be simulated using rendering software. The main advantage of using a real-life camera would be that the captures are true-to-life and therefore accurately reproduce all the distortions that are present in a typical scene. Consequently, a simulator should be unbiased for this to not be a key differentiator.

Nevertheless, this strength also turns out to be one of its biggest weaknesses. Objects may move slightly between experiments due to unintended human intervention. In turn, this might lead to the objects now being placed in such a way that reflections cause new distortions, making comparisons with previous experiments erroneous. The objects of importance here are typically slippery metal parts, and thus such displacements can easily occur without being easily noticeable.

Perhaps an even larger source of error is the camera positioning. Throughout the thesis, experiments will be performed by taking captures using a range of distances. If a physical system was to be used, this would require measuring and repositioning the system for each capture. An incorrect positioning can easily occur in both the distance and pose. This is mitigated by using a simulator because the distance and pose are set programatically.

In addition, comparisons with the ground truth will be used extensively to find out where pattern codification strategies give incorrect results. A physical camera has for obvious reasons no way of obtaining the ground truth, which makes it impossible to use for such comparisons. Fortunately, there are certain ways to find the ground truth using the simulator, which will be explained later.

For these reasons, a simulator is used in the experiments. The experimental setup revolves around a rendering engine configured to approximate the Zivid Two camera. A programming framework has been developed on top of this software to automate the process of rendering different code matrices for a range of configurations.

4.1.1 Software overview

OctaneRender® by OTOY Inc. was chosen as the rendering engine for the experiments. It is the world's fastest GPU-accelerated renderer (OTOY 2022), meaning that few compromises have to be made in the simulations in order to optimize for fast render times. Even more importantly, it is unbiased and physically correct. This means that physical phenomena such as specular reflections can be trusted to resemble reality.

The rendering engine has a feature rich application programming interface (API), which makes it possible to programatically modifyi attributes such as the camera distance Z_C and the patterns to be used. By using its command-line interface (CLI), rendering can be started from the command line without having to use a GUI. Together, these features make it possible to fully automate the experiments used in the thesis. The programming framework that performs the automation has been developed in Python. The language was chosen for its ease of use and wide variety of libraries. In particular, the library called `OpenCV2` proved to be useful in converting the renderings to a more programming-friendly format.

The rendering engine has been configured to use the camera and projector matrices as defined in (14) and (15) respectively. Furthermore, the combined translation and rotation matrix $\mathbf{T_{pc}}$ defined in (16) is used. These are the values that apply to the Zivid Two camera (*Zivid Two Datasheet* 2021). To keep things simple, the camera was configured with an aperture of $f/5.6$. According to Zivid AS, this is a value that is typical for the camera.

$$\mathbf{K_c} = \begin{bmatrix} 1.728 \times 10^3 & 0 & 9.715 \times 10^2 & 0 \\ 0 & 1.728 \times 10^3 & 7.355 \times 10^2 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{14}$$

$$\mathbf{K_p} = \begin{bmatrix} 1.153 \times 10^3 & 0 & 6.395 \times 10^2 & 0 \\ 0 & 1.035 \times 10^3 & 3.595 \times 10^2 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{15}$$

$$\mathbf{T_{pc}} = \begin{bmatrix} 9.925 \times 10^{-1} & -5.750 \times 10^{-7} & -1.218 \times 10^{-1} & 1.109 \times 10^2 \\ 8.071 \times 10^{-7} & 1 & 1.855 \times 10^{-6} & 2.965 \times 10^{-3} \\ 1.218 \times 10^{-7} & -1.939 \times 10^{-6} & 9.925 \times 10^{-1} & 1.899 \times 10^{-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{16}$$

4.1.2 Workflow and pre-processing

The programming framework has been developed with a focus on being easy to use, while still providing the flexibility to modify key attributes of the test setup. These advantages are achieved by making the framework fully declarative. In other words, the framework provides only a function called `render`, and all modifications must be made by changing parameters in a configuration file `render_config.yaml`. An example of such a configuration file has been provided in Source Code 1. The framework also takes the code matrix `code_matrix.mat` and a scene `scene_template.orbx`

as parameters. Together, these files are everything necessary to produce the render. The entire workflow is visualized in Figure 17. Here, white boxes with dashed blue border illustrate a file which is either provided by the user or produced by the framework. A box with a blue background represents a processing step. The whole workflow will be explained below.

Source Code 1: Default `render_config.yaml`.

```yaml
simulator:
    octane_path: "/localhome/studenter/leikli/OctaneRender/octane"
    projectors_count: 8 # [1]

scene:
    distance: 800 # [mm]
    rotation:
        vertical: 0.0 # [rad]
        horizontal: 0.0 # [rad]

camera:
  f_number: 5.6 # [1]
  focusing_distance: 800 # [mm]
  dimensions:
    x: 1944 # [cpx]
    y: 1472 # [cpx]

projector:
  power: 1.0 # [1]
  dimensions:
    x: 1280 # [ppx]
    y: 720 # [ppx]
```

4.1.2.1 Pattern generation

The rendering engine has no concept of a code matrix. Instead, the projector can be configured with a single pattern and render its light projected onto the scene as seen from the camera. Consequently, the first step in the workflow is to convert the code matrix into its corresponding patterns. The dimensions $Y_P \times X_P$ of the patterns are specified in `render_config.yaml` through `projector.dimensions.y` and `projector.dimensions.x` respectively. Recall that the code matrix can span any range of $\mathbb{R}$, but a projector can only output intensities $I \in [0, 1] \subset \mathbb{R}$. Therefore, the code matrix $\mathbf{C}$ should be mapped onto this range. This is done according to (17).

$$\mathbf{C}' = \frac{\mathbf{C} - \min\{\mathbf{C}\}}{\max\{\mathbf{C}\} - \min\{\mathbf{C}\}} \tag{17}$$

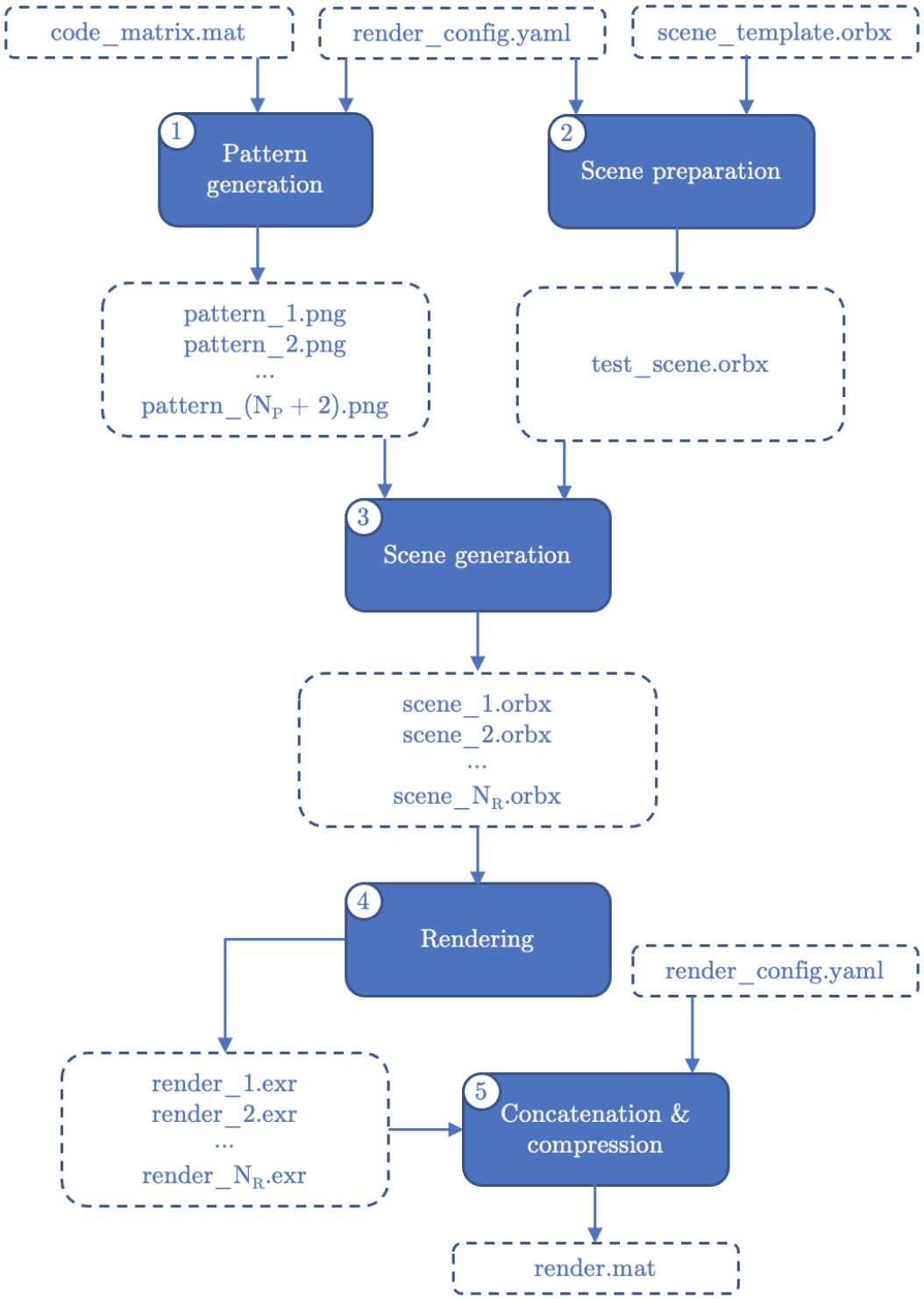


Figure 17: Flow chart of the rendering workflow.

Each row in the code matrix $\mathbf{C}'$ is then repeated Y_P times to make a pattern as seen in Section 2.2.1. In addition, a fully black and a fully white pattern is made to accommodate for the normalization of the captures according to Section 3.1.2.1. The patterns are saved as PNG-files, as the file format utilizes lossless compression (Salomon et al. 2006).

4.1.2.2 Scene preparation

Parallel to the pattern generation, the test scene `scene_template.orbx` needs to be modified to resemble the parameters specified in `render_config.yaml`. The modification is done by running the CLI of the renderer on `scene_template.orbx`. The CLI exposes its API through the `--script` parameter. A Lua-script is provided as parameter, which reads the YAML-file and modifies the parameters of the ORBX-files by utilizing the API. The resulting configured file is stored as `test_scene.orbx`. The programming framework allows setting the f-number, focusing distance X_C and Y_C of the camera. The projector can be configured through its power (maximum intensity), X_P and Y_P. For the experiments in the thesis, it is only the distance Z_C that is modified, and it is available through `scene.distance`. The pose of the camera can be set through `scene.rotation`.

4.1.2.3 Scene generation

Next, the ORBX-file needs to be configured to project the patterns through the projector. The renderer has the ability to record eight render passes, meaning that each render can output the capture of eight different patterns. This is exploited by having eight projectors stacked on top of each other in the rendering engine, and letting them output to each their own render pass. The total amount of renders would therefore be $N_R = \left\lceil \frac{N_P+2}{8} \right\rceil$, accounting for the additional white and black patterns.

For each render, an ORBX-file has to be created, configured with its corresponding eight patterns. This is done by running a different Lua-script which modifies the pattern input for each of the projectors. The Lua-script is executed on `test_scene.orbx` by applying the same technique as was done in the scene preparation.

4.1.2.4 Rendering

The fully configured scenes have to be rendered in order to generate the captures. The rendering is done by executing the render target through the command line. Each render generates a EXR-files, which is a slightly compressed raw image output containing the render passes. The rendering engine outputs both the direct reflections and the combined reflections (combination of the direct reflections and the interreflections), and these are stored in separate tensors.

4.1.2.5 Concatenation and compression

A code matrix consists in most cases of more than six patterns excluding the black and white ones, meaning that multiple renders are necessary. The renders are combined together to one single file in order to keep the files organized. They are stored in MAT-files, as these are easily read in python by using `scipy.io.loadmat`. For the purpose of debugging, the outputted `render.mat` also contains the configuration provided through `render_config.yaml`.

4.1.3 Limitations

Although the simulator is unbiased and fast, it also has its shortcomings. The most significant of these fall into the category of that the simulator performs *too good*. As previously mentioned, it does not suffer from any radial distortions. In addition, there are no calibration errors with regard to the relative positioning of the camera and projector. For physical systems, one would typically experience noise in the pixels (Jin and Hirakawa 2013), and this is also not present in the simulator. Such sources of distortions would degrade the performance of the system, meaning that the simulator gives a better performance than one could expect from a real-world structured light system. However, this difference is not expected to be significant. Moreover, the novel patterns will be compared to the state-of-the-art, which will be rendered using the same conditions. Therefore, the *relative* performance between these two should be a good metric of how well they compare with each other. In other words, if a new type of patterns performs better than the state-of-the-art in simulations, it should also perform better in the real world.

4.2 Test scenes

Two test scenes have been used for the experiments. They are both developed and provided by Zivid AS. When a structured light system is used in a real-world application, environmental lighting is almost always present. This could be artificial lighting such as the sun, lamps, etc. In the test scenes provided below, there is no ambient lighting. The exclusion was done so that only interreflections between different parts of the patterns themselves can affect the performance.

For some of the experiments, renders are performed on a range of distances called *distance sweeps*. Depending on the complexity of the calculations done on the renderings, two distance sweeps can be used. They are listed below.

$$\mathbf{z_{rough}}[\text{mm}] = \{300, 400, 500, 600, 650, 700, 750, 800, 1000, 1200, 1400, 1500\}$$
$$\mathbf{z_{detailed}}[\text{mm}] = \{300, 350, \cdots, 1450, 1500\}$$

4.2.1 Diffuse plane

A scene called *Diffuse plane* has been developed to be as simple as possible. As the name suggests, it is simply an infinitely-sized plane coated in a 100% diffuse material. The plane is perpendicular to the Z_C-axis of the camera in the structured light system, and the whole plane has the gray-color `rgb(200, 200, 200)`. The whole scene is planar, which in turn means that the projector pixels are not stretched or shrunk due to the terrain of the scene. Also, no occlusions are present of the same reasons. The absence of specular reflections further allows the analysis of pattern codification strategies *not* subject to interreflections.

4.2.2 Objects in bin

A more challenging scene called *Objects in bin* scene is used for the testing and evaluation of pattern codification strategies. It consists of a picking bin containing twelve cylinders and twelve truncated icosahedrons. The scene has been made in Cinema4D by simulating throwing the objects from a height of 1 m above the bin until they reside at their final positions. Therefore, the positioning of the objects should be similar to what is observed in a typical bin-picking scenario at a pick-and-place station. The resulting scene is shown in Figure 18. It has been rendered using the Zivid Two simulator by projecting a fully illuminated white pattern at distance $Z_C = 800\,\text{mm}$.

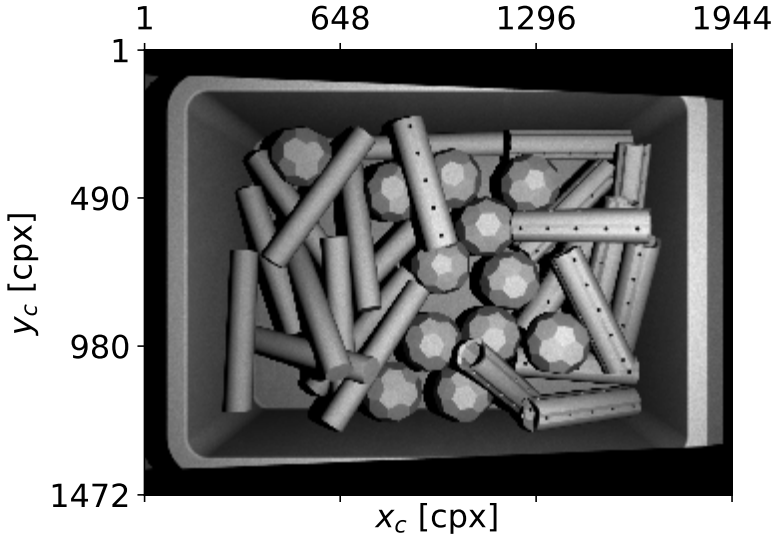


Figure 18: Render of the *Objects in bin* scene at distance $Z_C = 800\,\text{mm}$.

Notice from Figure 18 that there are several areas where little light from the projector reaches. This is known as occlusion, and means that no signal from the pattern codification strategies will be present. When the performance of these strategies later will be evaluated and compared against each other, it is important that the occluded areas are not considered in the analysis. Otherwise, the performance will be negatively biased. In order to disregard these areas, occlusion exclusion masks have been developed for the distances $Z_C \in \{550\,\text{mm}, 800\,\text{mm}, 1400\,\text{mm}\}$. The exclusion

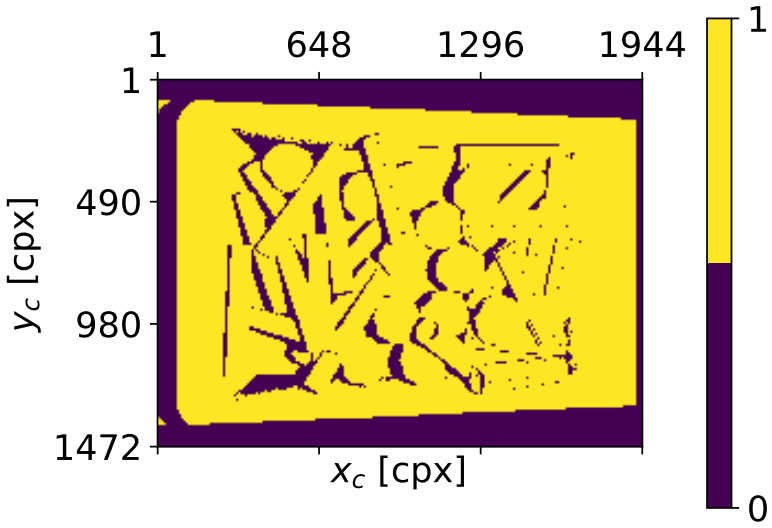


Figure 19: The occlusion exclusion mask for $Z_C = 800\,\text{mm}$.

masks are matrices which are zero in occluded areas and one otherwise. They are derived from fully illuminated captures by finding a threshold in light intensity for which brighter areas are not considered occluded. These matrices will be referred to as occlusion exclusion masks. Figure 19 shows the occlusion exclusion mask for distance $Z_C = 800\,\text{mm}$, and the other exclusion masks are available in Appendix A.

The picking bin and objects can be coated in one of two materials, which have optical properties resembling metals. The least challenging is the material **metal-50**. It gives off 50% diffuse reflections and 50% specular reflections and is close to the average metal part encountered in pick-and-place operations. A more challenging material is **metal-80**. This material gives off 80% specular reflections and 20% diffuse reflections, and is optically close to the most challenging metal parts.

4.3 Benchmarks

The results of the decoded captures using a particular pattern codification strategy only give the mapping $(x_c, y_c) \mapsto x_p$. This alone gives little information on how well the codification strategy performs. In order to give meaning to the results, there are two other results needed. First, the ground truth should be obtained. This data is the *correct* mapping $(x_c, y_c) \mapsto x_c$ that results from direct reflections only. By comparing the ground truth to the pattern codification strategy, it is possible to determine where this codification strategy decodes erroneously. Secondly, the results should be compared to the ones obtained using the state of the art; if the novel codification strategy does not improve compared to the de facto standard, then it is of no use.

4.3.1 Ground truth

The simulator will not give $(x_c, y_c) \mapsto x_p$ by itself, so other methods must be used. There are two ways of finding the ground truth, and the correct way of doing it will depend on the scene itself. For the *Diffuse plane* scene, the whole scene will have the same distance Z_C from the camera. Therefore, the equation (22) which will be derived later in Chapter 5 can be used to analytically find the correct mapping for this scene.

It is a bit more complex for the *Objects in bin* scene. Here, the mapping must be constructed using a pattern codification strategy. It has already been established that GCPS gives an accurate mapping, but only when interreflections are not present. These interreflections are easily eliminated by decoding the direct reflections only, available in the render output as a separate array.

4.3.2 State of the art - GCPS

In addition to the ground truth, GCPS will serve as a comparison to the state of the art. The pattern codification strategy is subject to the same scene covered with the same materials in order to give a fair comparison. In addition, the parameters of the codification strategy will be chosen such that it is similar to the codification strategy in subject.

4.4 Metrics

A pattern codification strategy works on a MIMO system, with a large output matrix of dimensions $Y_C \times X_C$. In order to give meaning to the results, some metrics should be defined.

4.4.1 Residual matrix

It has previously been established that both the ground truth and the estimated mapping $(x_c, y_c) \mapsto x_p$ from a particular pattern codification strategy can be obtained using the simulator. The difference between the ground truth and the estimated mapping will then indicate the decoding error in each camera pixel. This error will be known as the residual. Let $\mathbf{Q}$ be a $Y_C \times X_C$ matrix such that $(\mathbf{Q})_{y_c x_c}$ contains the estimated originating projector column for camera pixel (x_c, y_c). Also, let $\mathbf{Q'}$ be a similar matrix containing the ground truth projector columns. The residual matrix $\mathbf{E}$ is then defined as

$$\mathbf{E} = \mathbf{Q} - \mathbf{Q'} \tag{18}$$

These matrices will be visualized as heatmaps. One example of such a matrix is shown in Figure 20. Here, blue denotes that the camera pixel decodes a lower

projector column value than the ground truth, and vice versa for the red color. Note that the heatmap only spans the range $[-2, 2]$. As seen in Section 3.1, GCPS either gives residuals in integer multiples of W_F or it gives really small residuals. For the large residuals, it is not interesting to see whether it is W_F, $2 \cdot W_F$ etc as it is too large to be acceptable. Therefore, the residuals are clipped to the range $[-2, 2]$ so that the smaller, sub-pixels errors are better visualized. This would mean that large residuals would appear as ± 2 ppx.

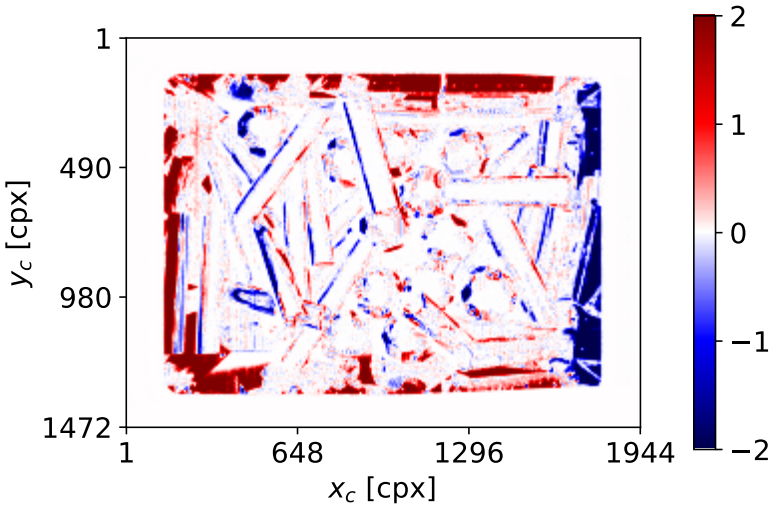


Figure 20: Example of a residual matrix heatmap.

4.4.2 Empirical CDF plot

While the residual matrix heat-map is good at showing the *spatial* distribution of residuals, it lacks in visualizing their *numerical* distribution. This is what an empirical CDF plot is useful for. This plot shows the cumulative distribution of the *absolute value* of the residuals. An example is included in Figure 21. For a given residual value along the x-axis, the corresponding y-value shows the fraction of the residuals which are equal to or lower than that particular residual. For instance, the example plot reveals that 60% of the residuals are less than 0.2 ppx, and that 80% are less than one projector pixel. Some details are however left out of the plot, as it has the range restricted to $[0, 1\,\text{ppx}]$. This choice of range was made so that the distribution of sub-pixel residuals are visualized the best. The rest of the empirical CDF will be summarized in an accompanying table.

4.4.3 Histogram of residuals

Histograms of the residuals will also be included for some of the results. These show the numerical distribution of the residuals and will, for instance, be useful to observe whether they follow a Gaussian distribution.

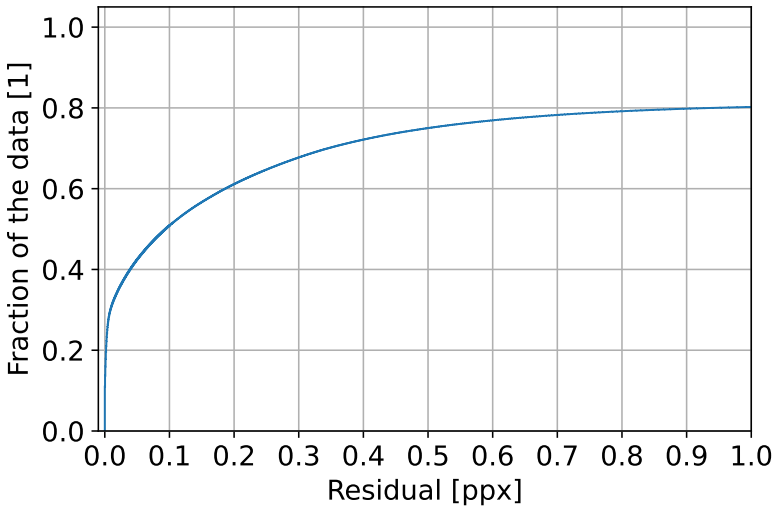


Figure 21: Example of an empirical CDF plot.

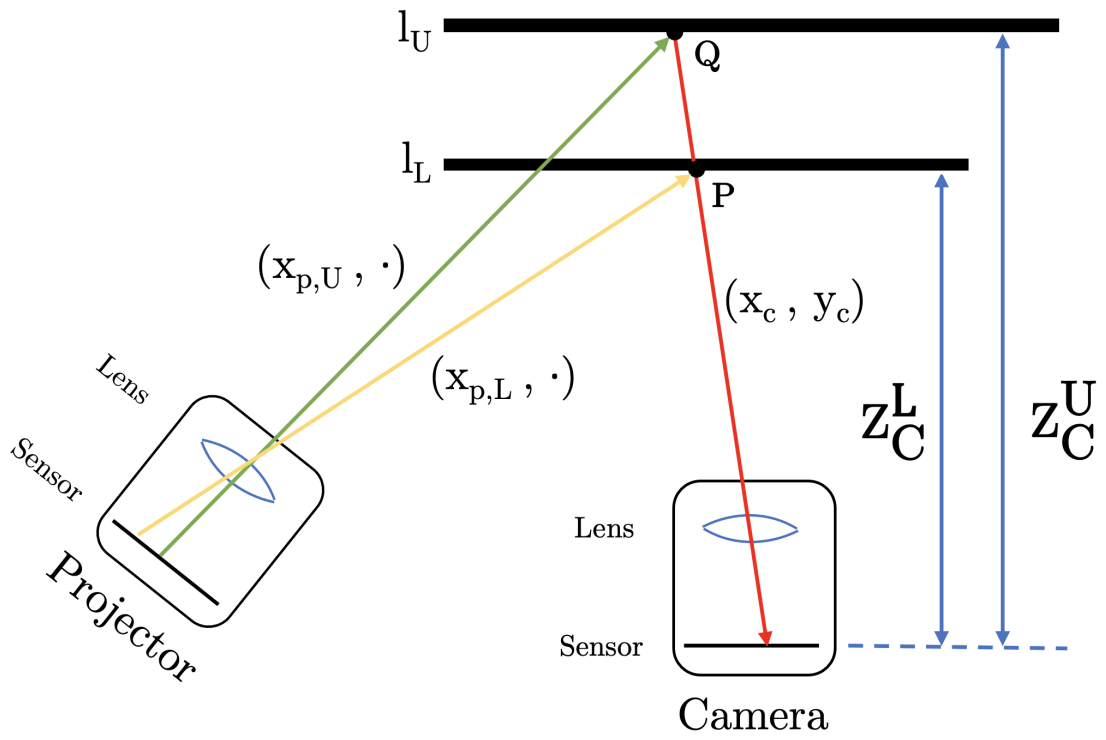


Figure 22: Simplified model of the working principles behind the projector column distance constraint.

5 Projector Column Distance Constraint

This chapter introduces a coding constraint for structured light systems. Due to geometric considerations, only a certain range of all projector columns can be seen by a particular camera pixel within a distance range $Z_C^L \leq Z_C \leq Z_C^U$. The constraint is first explained from a purely geometric perspective. Following is the development of the corresponding algorithm using the algebraic framework provided in Section 2.3. Lastly, the constraint is viewed from a practical perspective in terms of how it can be used to improve on the performance of pattern codification strategies. This constraint was first discovered in the project thesis (Lima-Eriksen 2022). The algorithm is reproduced here with some additions and corrections.

5.1 Working principles

A simplified model of a structured light system has been provided in Figure 22. Here, the camera and projector are viewed from a top-down perspective. The two large horizontal bars l_L and l_U represent possible objects causing reflections at their corresponding camera distances Z_C^L and Z_C^U. Recall from Section 2.2 that there exists a single ray for each camera pixel along which light may be incident onto that particular pixel. For the camera pixel (x_c, y_c), this ray corresponds to the red line in Figure 22. Introduce first the horizontal bar l_L at camera distance Z_C^L. Then the light that is captured by (x_c, y_c) would have to originate from a reflection at the

intersection between the red line and l_L at the point $\mathbf{P}$.

It has also been established in Section 2.2 that light originating from a projector column $(x_{p,L}, \cdot)$ forms a vertical plane in space. Let $(x_{p,L}, \cdot)$ denote the projector column that is incident on $\mathbf{P}$, indicated by the yellow line in Figure 22. The same considerations can be made for the object l_U at distance Z_C^U. The light incident on the same camera pixel (x_c, y_c) would now instead have to originate from a reflection in the point $\mathbf{Q}$. As seen in the figure, the projector column $(x_{p,U}, \cdot)$ indicated by the green line is the one seen in the camera pixel at that particular distance.

Note that this green line would have to originate from a projector column $x_{p,U} > x_{p,L}$. For a given range of camera distances $Z_C^L \leq Z_C \leq Z_C^U$, there should by induction exist a range of valid projector columns $[x_{p,L}, x_{p,U}]$ that are observable for a particular camera pixel (x_c, y_c). This is known as the projector column distance constraint. Its algorithmic derivation is provided in the following.

5.2 Algorithm

Section 2.3 introduced theory for mapping 3D world coordinates to 2D images and vice versa for the camera and projector. In addition, it gave the mapping between the camera world coordinates and projector world coordinates. Given a distance from the camera Z_C, the equations from Section 2.3 can be combined to find out which projector pixel (x_p, y_p) is visible from each camera pixel (x_c, y_c). More specifically, it makes it possible to construct the mapping $(x_c, y_c) \mapsto x_p$ for a given Z_C. Let $\mathbf{x}_c = \begin{bmatrix} x_c & y_c & Z_c \end{bmatrix}^T$. By reordering the camera version of (1), one gets

$$\mathbf{X_c} = \mathbf{K_c^{-1}} \mathbf{x_c} \tag{19}$$

The expression for $\mathbf{X_c}$ from (3) is inserted into (19), yielding

$$\begin{aligned} \mathbf{T}_{pc}\mathbf{X}_p &= \mathbf{K}_c^{-1}\mathbf{x}_c \\ \mathbf{X_p} &= \mathbf{T}_{pc}^{-1}\mathbf{K}_c^{-1}\mathbf{x}_c \end{aligned} \tag{20}$$

Lastly, the projector version of (1) is inserted into (20) to get the mapping from camera pixels to projector pixels at camera distance Z_C:

$$\begin{aligned} \mathbf{K_p^{-1}} \mathbf{x_p} &= \mathbf{T_{pc}^{-1}} \mathbf{K_c^{-1}} \mathbf{x_c} \\ \mathbf{x_p} &= \mathbf{K_p} \mathbf{T_{pc}^{-1}} \mathbf{K_c^{-1}} \mathbf{x_c} \end{aligned} \tag{21}$$

Define two camera distances Z_C^U and Z_C^L such that $Z_C^U > Z_C^L$. Consider now the tensors $\mathbf{C^U}$ and $\mathbf{C^L}$ of dimensions $X_p \times Y_p \times 3$ where

$$\left(\mathbf{C^{U|L}}\right)_{x_c y_c} = \begin{bmatrix} x_c & y_c & Z_C^{U|L} \end{bmatrix}^T \quad \begin{matrix} \forall & x_c \in \mathbb{N} \cap [1, X_C] \\ \forall & y_c \in \mathbb{N} \cap [1, Y_C] \end{matrix}$$

Then the tensors $\mathbf{P^U}$ and $\mathbf{P^L}$ of the same dimensions $X_p \times Y_p \times 3$ can be constructed using (21) such that

$$\left(\mathbf{P^{U|L}}\right)_{x_c y_c} = \mathbf{K}_p \mathbf{T}_{pc}^{-1} \mathbf{K}_c^{-1} \left(\mathbf{C^{U|L}}\right)_{x_c y_c} \tag{22}$$

The element $\left(\mathbf{P^{U|L}}\right)_{x_c y_c}$ would then contain the homogeneous projector pinhole coordinate belonging to camera pixel (x_c, y_c) at the camera distance $Z_C^{U|L}$. Therefore, $\left(\mathbf{P^U}\right)_{x_c y_c 2}$ and $\left(\mathbf{P^L}\right)_{x_c y_c 2}$ now contain the largest and smallest values for x_p respectively that a particular camera pixel (x_c, y_c) can capture given the constraint that all distances the camera can capture are in the range $Z_c \in \left[Z_C^L, Z_C^U\right]$. Consider the vectors $\mathbf{p^U}$ and $\mathbf{p^L}$ each of dimensions $X_C \times 1$ where

$$\left(\mathbf{p^{U|L}}\right)_{x_c} = \max_{i \in [1, Y_C] \subset \mathbb{N}} \left(\mathbf{P^{U|L}}\right)_{x_c i 2} \tag{23}$$

Now $\left(\mathbf{p^U}\right)_{x_c}$ and $\left(\mathbf{p^L}\right)_{x_c}$ contain the largest and smallest observable projector column respectively for a given x_c. Then a vector $\mathbf{d}$ of dimensions $X_C \times 1$ can be defined such that

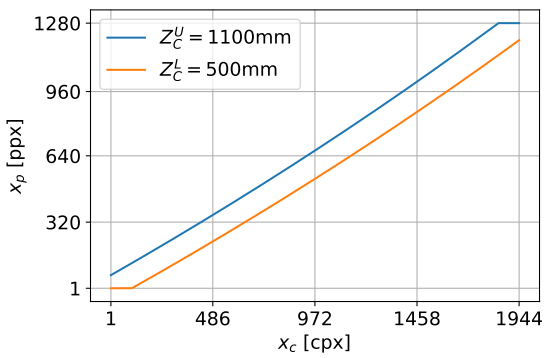
$$\left(\mathbf{d}\right)_{x_c} = \frac{\left(\mathbf{p^U}\right)_{x_c} - \left(\mathbf{p^L}\right)_{x_c}}{X_P} \tag{24}$$

Each element $\left(\mathbf{d}\right)_{x_c} \in (0, 1]$ now specifies the ratio of projector columns that can be seen in a particular camera column x_c.

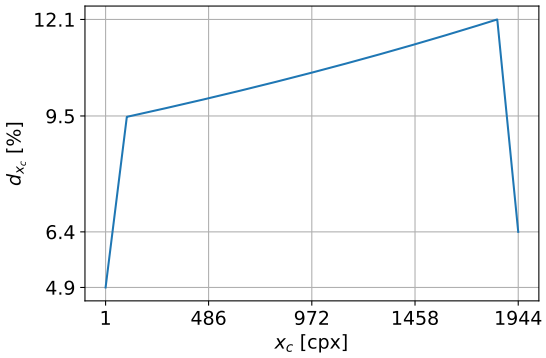
5.3 Usage

Consider the optimal working distance of the camera (*Zivid Two Datasheet* 2021) such that $Z_C^L = 500\,\text{mm}$ and $Z_C^U = 1100\,\text{mm}$, and define $\mathbf{K_c}$, $\mathbf{K_p}$ and $\mathbf{T_{pc}}$ according to (*Zivid Two Datasheet* 2021) as done in equations (14), (15) and (16) respectively. Then $\mathbf{p^L}$ and $\mathbf{p^U}$ become piecewise straight lines as depicted in Figure 23a. Notice how the lines are quite close to each other, meaning that each camera column x_c can only see a small range of values for x_p. This is further visualized in Figure 23b, where it is apparent that at most 12% of the projector columns x_p is visible for any x_c.

These calculations can be extended to a range of different pairs of distances $\left(Z_{C,i}^L, Z_{C,i}^U\right)$. By calculating $\max\left(\mathbf{d}\right)$ for the vector $\mathbf{d}$ corresponding to each pair of $\left(Z_{C,i}^L, Z_{C,i}^U\right)$, a matrix can be made. One such matrix that uses the attributes for the Zivid Two



(a) $\left(\mathbf{p^U}\right)_{x_c}$ (orange) and $\left(\mathbf{p^L}\right)_{x_c}$ (blue) plotted for all $x_c \in [1, X_c] \subset \mathbb{N}$.



(b) $(\mathbf{d})_{x_c}$ plotted for all $x_c \in [1, X_C] \subset \mathbb{N}$.

Figure 23: Distance constraint for Zivid Two when $Z_C^L = 500\,\text{mm}$ and $Z_C^U = 1100\,\text{mm}$.

camera has been illustrated in Figure 24. Note that for most of the distance ranges less than 20% of the projector columns can be visible for a particular camera pixel.

As seen in the above-mentioned plots, the distance constraint significantly restricts the possible projector columns in each of the camera pixels. One way to use the constraint is by simplifying the codes in a pattern codification strategy. For instance, the gray codes in GCPS require the unique identification of all $N_F = \frac{X_P}{W_F}$ fringes. However, this constraint means that at most $\lceil \max(\mathbf{d}) \cdot N_F \rceil$ fringes can be seen for any camera pixel. Therefore, the length of the gray codes can be reduced accordingly.

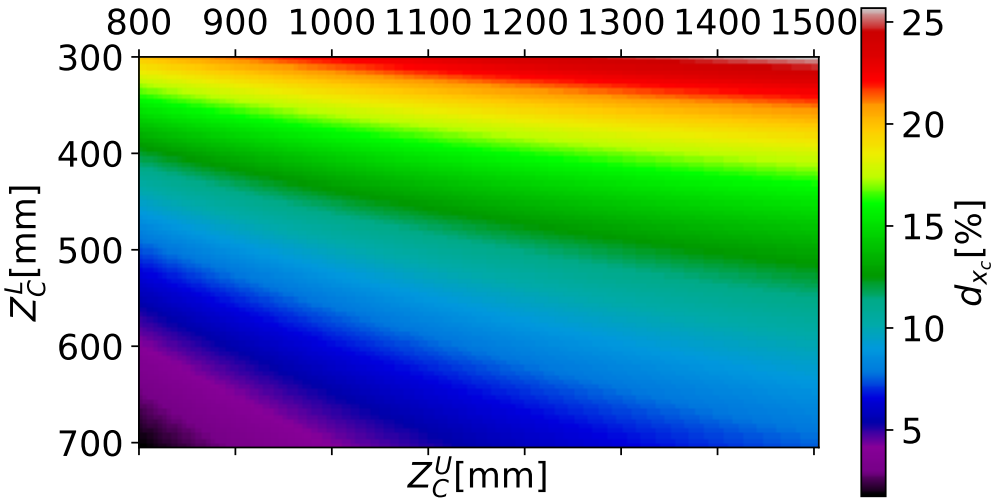


Figure 24: Maximum ratio of projector columns visible per camera pixel for select pairs of camera distances (Z_C^L, Z_C^U).

Another perhaps more relevant use case for the work in this thesis is to filter out invalid decoded projector columns. If a camera pixel decodes a projector column that is too large or small to fit within the distance constraint, it can be considered invalid. This will be used later in the development of a novel pattern codification strategy.

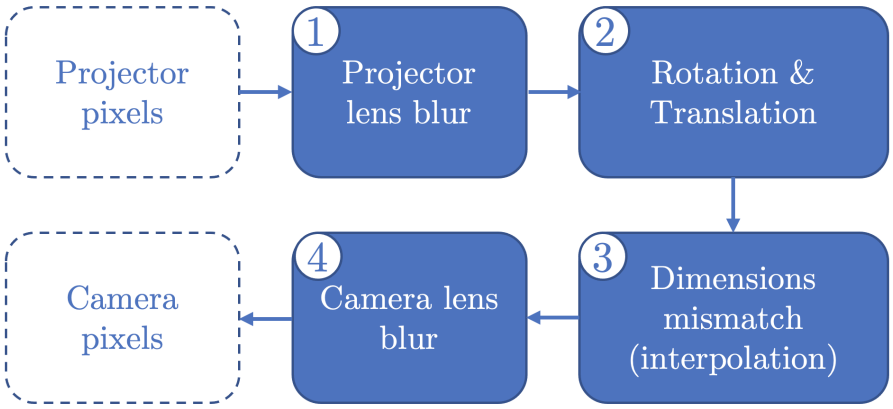


Figure 25: Flow chart of how a projector pixel is modified before captured by a camera pixel.

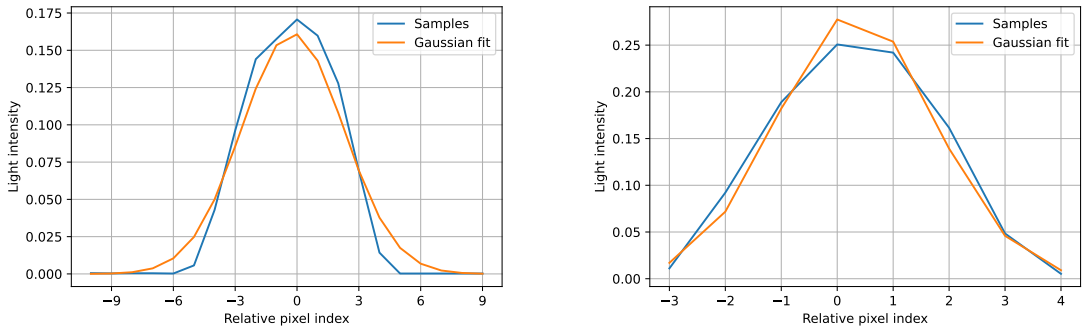
6 System Identification

This chapter serves as an overview of the attributes of the system which are of relevance in the domain of signal processing. Since the system is highly non-linear, the attributes will be estimated in an empirical fashion using the experimental setup as provided in Section 4.1. First, the empirical point-spread function (PSF) is estimated. This will prove to be useful in estimating the amount of blur that should be added in certain numerical approximations. In the latter half, the frequency response of the system will be estimated and analyzed. The frequency response is later used as a base for choosing parameters in patterns which have known spatial frequencies.

6.1 Point-spread function

The point-spread function is a measure of how an imaging system responds to an excitation caused by a point source (Rottenfusser et al. 2022). In the context of structured light systems, this corresponds to how the illumination of a single projector pixel spreads out in the capture made by the camera. Under ideal circumstances, the illumination of a single projector pixel should illuminate a single camera pixel. In that case, the PSF would be equal to the Dirac delta function. Several optical and geometric considerations prevent this from happening. An overview of the steps involved between the illumination of a projector pixel to the capture of a camera pixel is presented in Figure 25.

The projector pixel first travels through the projector lens, and hits a surface. Due to lens defocus, the pixel will be blurred out as seen in Section 2.4. This effect can be approximated as the convolution with a Gaussian kernel where the SD is dependent on the distance to the projector Z_p. The camera and projector are translated and rotated relative to each other as seen in Section 2.3.2. Therefore, the blurred projector pixel will appear skewed in the camera, further adding to the PSF. For structured light systems with low baseline such as Zivid Two, this effect



(a) $Z_C = 500\text{mm}$ (b) $Z_C = 1300\text{mm}$

Figure 26: Examples of how Gaussian fits match the empirical PSF for various distances.

will typically be insignificant. There is also a mismatch between the spatial size of the projector and camera pixels. This effect depends on the field of view and the number of pixels in the x and y directions for the projector and camera. In the case of a Zivid Two camera, $S_X = 1.7\,\text{cpx/ppx}$ from the camera will cover a single projector pixel in the x-dimension. From a signal processing perspective, the effect of this will be a linear interpolation of the blurred and skewed projector pixel, which further modifies the PSF. Lastly, the signal is subject to the blur caused by the camera lens. This effect is also approximated by a Gaussian blur with a standard deviation that depends on the distance to the camera Z_C.

As seen above, there are multiple distorting factors that contribute to the PSF. Therefore, an analytical derivation of the PSF is not feasible and an empirical approach is desired. Recall from Section 3.1 that the patterns only vary along the x-axis. This means that the PSF will distort the patterns in this direction only, and an estimate of the PSF in the x-direction should be sufficient to describe the distortions caused by it.

A pattern $\mathbf{P_{PSF}}$ consisting of vertical lines of width 1 ppx with a spacing of 100 px between each line will be used to obtain the PSF estimate. The pattern will be rendered using the *diffuse plane* scene, as it has a constant distance Z_C for all camera pixels. Certainly, the PSF will vary with the camera distance Z_C. By that reason, the estimate should be made for a range of distances. The calculations are not computationally demanding, which means that the $\mathbf{z_{detailed}}$ distance sweep can be used here.

The empirical PSFs are plotted for the distances $Z_C \in \{500\,\text{mm}, 1100\,\text{mm}\}$ in Figure 26 as the blue curves. Gaussian fits have been made for them both, and they are illustrated by the orange curves. Notice how the Gaussian fits either match the samples well or slightly overestimates the standard deviation σ_{PSF}. By that reason, the standard deviation seems to be a reasonable measure of the PSF.

With the captures obtained of the $\mathbf{P_{PSF}}$ using the $\mathbf{z_{detailed}}$ distance sweep, the resulting blurred vertical lines must be identified in each of the captures. This is done using a simple peak-finding algorithm. For this case, the `find_peaks` from the

`scipy.signal` python package is used. First, the center row of each capture m is extracted. Next, the `find_peaks` function is applied to this vector with parameters `height=100`, `distance=30`, and `prominence=5`. The function returns a vector containing the center indices of the peaks in capture m obtained at the distance $(\mathbf{z_{detailed}})_m$.

From each of these indices, new vectors $\mathbf{s_n^m}$ can be made by considering the $N_S = 71$ samples centered around the sample corresponding to each of the indices in capture m. As an example, the vectors $\mathbf{s_1^3}$ and $\mathbf{s_2^3}$ will then contain the samples centered around the first and second peak in capture $m = 3$, respectively. These vectors will then contain the samples of each of the blurred lines. Let $\mathbf{x} = \begin{bmatrix} -30 & 29 & \cdots & 30 \end{bmatrix}^T$, and define the normalized samples vectors

$$\mathbf{u_n^m} = \frac{\mathbf{s_n^m}}{\sum_i \left(\mathbf{s_n^m}\right)_i}$$

An unbiased estimator for the standard deviations of each of the lines n in every capture m is found through the empirical standard deviation defined below:

$$\left(\widehat{\boldsymbol{\sigma}_{\mathbf{PSF}}^{\mathbf{m}}}\right)_n = \frac{1}{S_X} \cdot \sqrt{\sum_i (\mathbf{x})_i^2 \left(\mathbf{u_n^m}\right)_i - \widehat{\mu_n}^2} \quad \text{, where} \quad \widehat{\mu_n} = \mathbf{x^T} \cdot \mathbf{u_n^m} \tag{25}$$

Now $\widehat{\boldsymbol{\sigma}_{\mathbf{PSF}}^{\mathbf{m}}}$ is a vector containing all estimates of the σ_{PSF} for a particular capture m. The term $\frac{1}{S_X}$ comes from the fact that the standard deviation is measured in camera pixels but should be specified in projector pixels. These estimates should be aggregated to a single estimate for each capture m. As the PSF will be used as a measure for the worst possible distortions, the following estimator will be used for a given distance:

$$\widehat{\sigma_{\text{PSF}}^m} = \max_n \left(\widehat{\boldsymbol{\sigma}_{\mathbf{PSF}}^{\mathbf{m}}}\right)_n \tag{26}$$

The calculations have been done with the result is plotted in Figure 27. Notice that $\sigma_{\text{PSF}} \leq 0.9\,\text{ppx} \,\forall\, Z_C \in [550\,\text{mm}, 1500\,\text{mm}]$. Also, it grows fast for $Z_C < 550\,\text{mm}$. This suggests that the range $550\,\text{mm} \leq Z_C \leq 1500\,\text{mm}$ might be beneficial for pattern codification strategies requiring a low standard deviation.

6.2 Frequency response

Before estimating the frequency response of the system, it is important to have a clear definition of what the response measures in this particular sense. Normally in signal processing, the frequency response defines the quantitative measure of amplitude and phase changes of cosines propagating through a system from an input to the output (Smith 1997). Structured light is a system identification problem itself, where the task is to find the mapping from the input (projector columns)

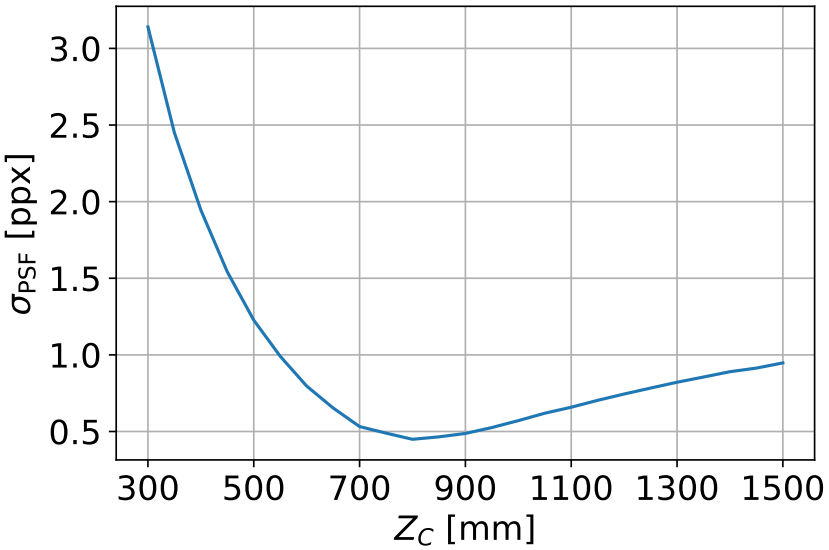


Figure 27: Standard deviation estimates of the empirical PSF for the system.

to the output (camera pixels). Since this mapping is not known beforehand, one cannot think of the frequency response in the same way for structured light. To complicate things further, the system is highly susceptible to cross-talk between inputs. In fact, cross-talk is what happens when blur is present and causes signal to spread into neighboring camera pixels. Another major difference is in which domain the frequency response is to be calculated. For an analog LTI circuit, one would calculate the response in the temporal domain, meaning that the system responds differently to varying frequencies in time. This temporal variability does not occur in structured light systems. Instead, it is the *spatially* varying frequencies which are subject to amplitude change through the system. The same phenomena as shown in Figure 25 are also responsible for these effects.

As with LTI systems, there are two ways of estimating the frequency response of a structured light system. The impulse response can be found, and from this the frequency response can be obtained through calculating the discrete fourier transform of it (Smith 1997). An impulse response does not exist for structured light systems, but the PSF has the same usage for such systems. The PSF found in the preceding section is an approximation to the actual PSF. Nevertheless, it is a worst-case estimate which relies of several simplifications, and it was made for other use cases. Therefore, it is not considered usable for obtaining the frequency response.

Another way of calculating the frequency response is by projecting cosines of the desired range of frequencies, and observing how much their amplitudes are dampened. As mentioned previously, these cosines should vary in the spatial domain, and thus their periods are denoted in projector pixels [ppx]. The cosines will vary spatially along the projector x-axis, as this also is the direction that conventional patterns will vary.

Normally for standard LTI systems, the cosine is sampled along the same dimension as the cosine propagates; the cosine would in the LTI case both propagate in the temporal dimension and be sampled in the temporal dimension. If the same prin-

ciple was to be applied to the structured light case, it would mean that one should consider multiple samples along the x-direction in the capture of a spatially varying cosine, and use those samples to estimate the amplitude. However, it is already established that each camera pixel is a separate output in the captures. Using multiple camera pixels to make these calculations would therefore not correspond to the true frequency response. Instead, one needs to obtain multiple samples of each cosine using the same input (projector pixel) and output (camera pixel).

One way of accomplishing this can be found by considering how the phase shifts are used in GCPS as seen in Section 3.1. Here, a cosine with a particular spatial frequency is phase shifted in order to obtain multiple samples of it using the same input and output. This technique can be modified for the estimation of the frequency response. The phase shifting will be performed four times as in GCPS, but for multiple spatial frequencies within the range of interest. Instead of estimating the phase of the cosines, the amplitude is what is desired. By calculating the DFT of all of the samples in the temporal domain for a particular camera pixel using a particular spatial frequency, the amplitude can be obtained. This is done by calculating the absolute value of the DFT, and summing the components corresponding to temporal frequencies of $f_t = \pm\frac{1}{2}$Hz. The possible intensity range is found by projecting an all black and all white pattern. This obtained intensity range can then be used in order to normalize the amplitudes to a ratio between 0 and 1, where 1 is no amplitude dampening. Estimates will then be obtained for each camera pixel for each chosen spatial frequency, whereas a single estimate for each pair of spatial frequency and distance is desired. This is done by calculating the mean of all samples for each such pair. These renders should be performed using the *diffuse plane* scenes, for the same reasons as for the PSF estimates. Also, it should be done for a range of distances $\mathbf{z_{detailed}}$.

There are two free variables in the estimates — the spatial frequency f_s and the camera distance Z_C. This leads to two ways of visualizing the results. By letting the x-axis represent the spatial frequency, one will obtain the typical frequency response, and each curve will correspond to a certain distance. The frequency response is visualized in Figure 28. There are multiple insights to be made from this plot. First, it is apparent that all frequencies are eliminated for $Z_C = 300\,\text{mm}$. This is particularly unexpected, as the working distance of the Zivid Two spans from this distance and can be used with GCPS. It is unclear what causes this phenomenon, but it could be due to the fact that the simulator is not a fully realistic model of the physical system. This plot also makes it apparent that the system in fact acts as a low-pass filter; all of the curves resemble the characteristic shape of a frequency response belonging to such a filter. This is also expected because the PSF can be approximated by the convolution with a Gaussian kernel as seen previously.

Perhaps an even more useful representation is obtained by assigning the camera distance Z_C to the x axis and letting each curve correspond to a particular spatial frequency. This type of visualization is made in Figure 29, and will from now be known as the *distance response*. Notice in particular how all of the curves follow the same shape; they are quite low for small distances, and quickly grow towards their maximum attained at around $Z_C = 800\,\text{mm}$. From there on, they fall towards further growing distances. Nevertheless, these are the only similarities between the

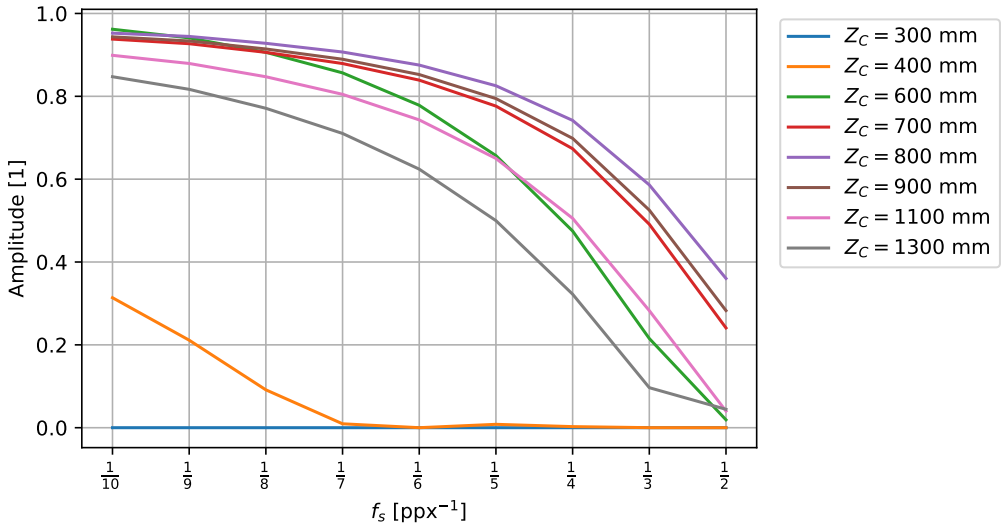


Figure 28: Empirical frequency response of the system.

curves. Observe that a decreasing spatial frequency leads to a distance response that rises faster for small distances, and falls slower for the large distances. Also, the distance response reaches a higher maximum value for lower frequencies. In general, this concludes the fact that lower spatial frequencies function better in structured light systems. The plot also reveals that subsequent decreases in spatial frequency lead to less improvement in response, converging to a maximum. The differences are mainly seen in the tails of the curves for particularly large distances. For example, there are few differences between the distance responses comparing $f_s = \frac{1}{9}\text{ppx}^{-1}$ to $f_s = \frac{1}{10}\text{ppx}^{-1}$. Depending on the usage of the system, this plot suggests that patterns containing spatial frequencies below $f_s = \frac{1}{8}\text{ppx}^{-1}$ should function well.

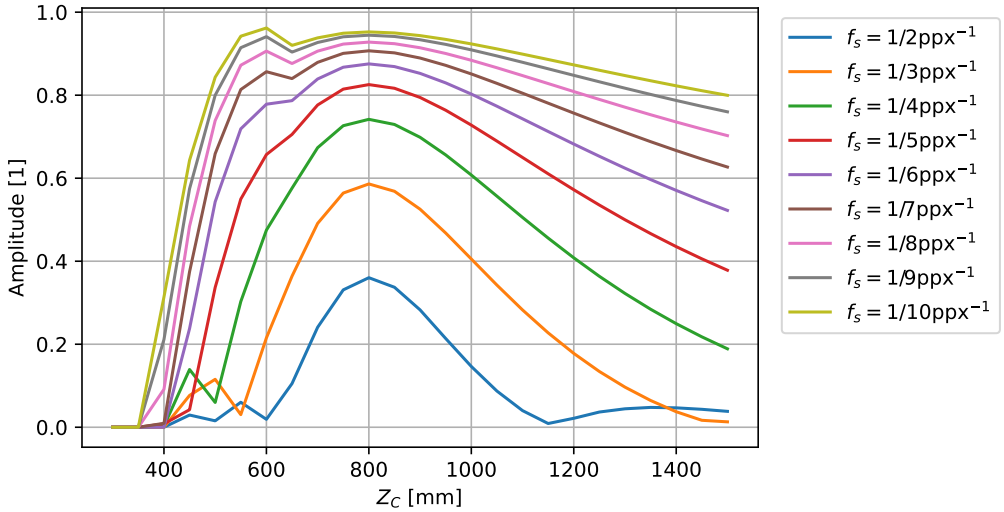


Figure 29: Empirical distance response of the system.

7 Distortion-Resilient Patterns

This chapter introduces two novel types of patterns which are resilient to the defocus and interreflection distortions as presented in Section 2.4 and Section 2.5 respectively. The patterns will later be combined to form a complete pattern codification strategy in Chapter 8. The first type of pattern is based on the principle of separating signals through correlation. It allows for the unique identification of fringes similar to the gray codes as presented in Section 3.1. This type of pattern is an improvement of the correlation-based pattern codification strategies developed in the project thesis (Lima-Eriksen 2022), adding the possibility of code words that occupy fringes of width W_F instead of being only one projector pixel wide. The latter type of patterns presented in this chapter is a modification of the phase shifts from GCPS which makes it more resilient to interreflections. It allows for sub-pixel decoding of projector columns.

7.1 Correlation-identified fringes

A major issue with the gray codes used in GCPS is that a single bit error makes the decoding erroneous. This can often be the case when there are interreflections. Patterns which tolerate bit errors are needed to allow the application of structured light in such demanding environments. These attributes are found in bipolar sequences with good correlation properties, typically known from the field of telecommunication. Therefore, patterns using these sequences should be a good fit.

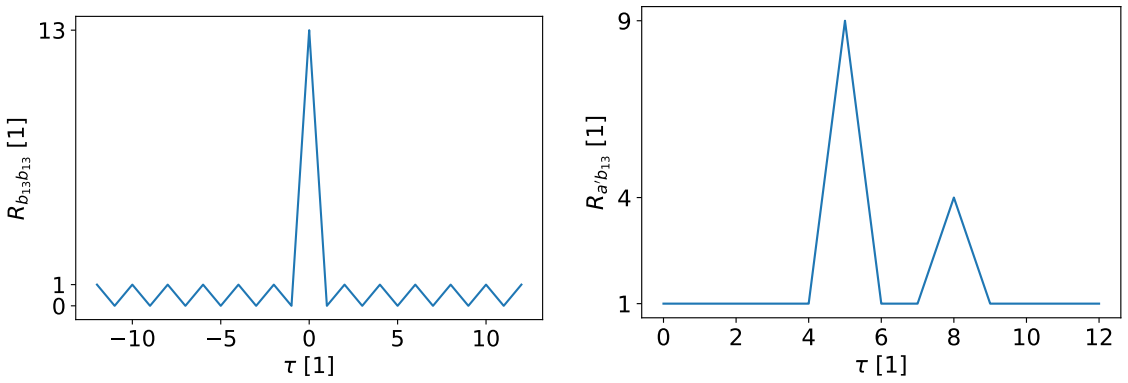
The project thesis (Lima-Eriksen 2022) introduced such correlation-based patterns with a fringe width $W_F = 1\,\text{ppx}$. As mentioned above, it was discovered that the patterns worked well within focus, but failed outside. The distance response in Section 6.2 indicated that patterns containing low spatial frequencies should work better at distance outside of focus.. An increase in W_F would lead to such a lowering of spatial frequencies, as more neighboring pixels are similar. The correlation-identified fringes from the project thesis will therefore be extended to allow for an arbitrary fringe width W_F. Succeeding this, insights into how the choice of W_F affects the performance will be given.

7.1.1 Working principles

Consider first binary bipolar sequences with *ideal* correlation properties. There are in total seven known sequences with these requirements, and they are known as the Barker sequences (Barker 1953). The longest of them all is the 13-digit Barker sequence $b_{13}[n]$ as defined in (27). Its auto-correlation function has been plotted in Figure 30a, showing a large peak at lag $\tau = 0$ and low values otherwise.

$$b_{13}[n] = \{1, 1, 1, 1, 1, -1, -1, 1, 1, -1, 1, -1, 1\} \tag{27}$$

Define the $n \times n$ circular right-shift matrix $\mathbf{R_n}$ as



(a) The auto-correlation function $R_{b_{13}b_{13}}(\tau)$ of $b_{13}[n]$.

(b) The cross-correlation function $R_{a'b_{13}}$of $a'[n]$ and $b_{13}[n]$.

Figure 30: The 13-digit Barker code $b_{13}[n]$. Courtesy of Lima-Eriksen 2022.

$$\mathbf{R_n} = \begin{bmatrix} 0 & & & \\ \vdots & & \mathbf{I_{n-1}} & \\ 0 & & & \\ 1 & 0 & \cdots & 0 \end{bmatrix}$$

where $\mathbf{I_{n-1}}$ is the $(n-1)\times(n-1)$ identity matrix. Now, all the 13 circularly shifted permutations of $b_{13}[n]$ can be derived from (27) such that

$$\mathbf{s_m} = \mathbf{R_{13}^m b_{13}} \quad \forall \quad m \in \mathbb{N} \cap [0, 12] \tag{28}$$

Consider the case in which a sequence $a[n]$ is made by superpositioning two scaled $A \cdot s_m[n]$ and $B \cdot s_o[n]$ such that $m \neq o$. Then

$$a[n] = A \cdot s_m[n] + B \cdot s_o[n] \tag{29}$$

Correlation is a linear operator, and so

$$\begin{aligned} R_{ab_{13}}(\tau) &= (a * b_{13})(\tau) \\ &= ((A \cdot s_m + B \cdot s_o) * b_{13})(\tau) \\ &= (A \cdot s_m * b_{13})(\tau) + (B \cdot s_o * b_{13})(\tau) \\ &= A \cdot R_{s_m b_{13}}(\tau) + B \cdot R_{s_o b_{13}}(\tau) \end{aligned} \tag{30}$$

Since $b_{13}[n]$ has an ideal autocorrelation function, (30) implies that $R_{ab_{13}}(\tau)$ will have two distinct peaks at $\tau = m$ and $\tau = o$, where the largest peak corresponds to the sequence having the largest scaling factor. This is shown visually in Figure 30b considering the realization $a'[n]$ of $a[n]$ as defined in (31).

$$a'[n] = 0.7 \cdot s_5[n] + 0.3 \cdot s_8[n] \tag{31}$$

Consider the example of reflections as depicted in Figure 11b. If each projector column $x_{p,m}$ encoded the pattern $\mathbf{s_m}$ in the temporal domain, then $a'[n]$ would be the signal captured in (x_c, y_c), and the direct reflection and interreflection would be separable by calculating the cross-correlation as depicted in Figure 30b.

For practical applications, the Barker sequences are too short. Having at most 13 distinct permutations, they can only uniquely identify 13 columns. Therefore, the requirement of having *ideal* correlation properties should be relaxed to having *good* correlation properties.

Several families of sequences meet this requirement, including the Kasami and Gold sequences (Spinsante et al. 2011). The Gold sequences will be used throughout the thesis due to its favorable lengths, but the algorithms apply equally to other families as well.

First discovered by Robert Gold in 1967, Gold codes are a type of bipolar sequences typically used in CDMA and GPS (Gold 1967). One set has $2^m + 1$ Gold codes, each having a period of $2^m - 1$. The autocorrelation function is two-valued and is defined as

$$R_{g_m}(\tau) = \begin{cases} \pm 2^m - 1 & , \quad \tau = 0 \\ \mp 1 & , \quad \tau \neq 0 \end{cases} \tag{32}$$

All except one of the Gold codes will have zero cross-correlation for lag $\tau = 0$. There exists no closed-form expression for the cross correlation of any two gold codes $g_{m,i}$ and $g_{m,j}$, but its absolute value is upper bounded by

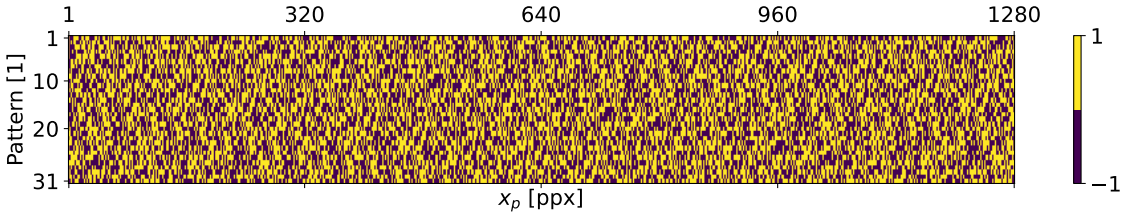
$$\left| R_{g_{m,i} g_{m,j}, max} \right| = \begin{cases} 2^{(m+2)/2} + 1\,, & m \text{ even} \\ 2^{(m+1)/2} + 1\,, & m \text{ odd} \end{cases} \tag{33}$$

7.1.2 Algorithm

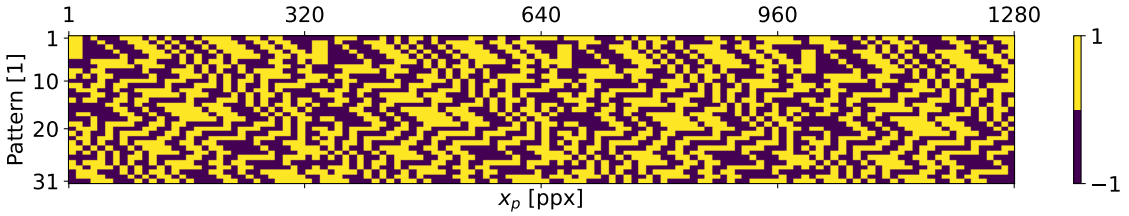
For the codes to be used as patterns, both an encoding and a decoding algorithm should be constructed. This section serves as an explanation on how the codes are used to construct patterns, and how the sequence of patterns captured can be decoded to reconstruct the originating fringe.

7.1.2.1 Encoding

A set of Gold codes can be generated as follows: A preferred pair of m-codes having length $2^m - 1$ is first generated from two irreducible polynomials. A table of such



(a) The code matrix $\mathbf{C_G}$.



(b) The code matrix $\mathbf{C_{G(10)}}$.

Figure 31: Code matrices using Gold codes.

irreducible polynomials is available from (Peterson 1970). The preferred pair will be known as the first two Gold codes $\mathbf{g_1}$ and $\mathbf{g_2}$ in this particular set. Then the remaining $2^m - 1$ Gold codes can be derived from these through

$$\mathbf{g_n} = (\mathbf{R}_{\mathbf{2^m-1}}^{\mathbf{n-2}} \cdot \mathbf{g_1}) \oplus \mathbf{g_2} \quad \forall \quad n \in \mathbb{N} \cap [3, 2^m + 1] \tag{34}$$

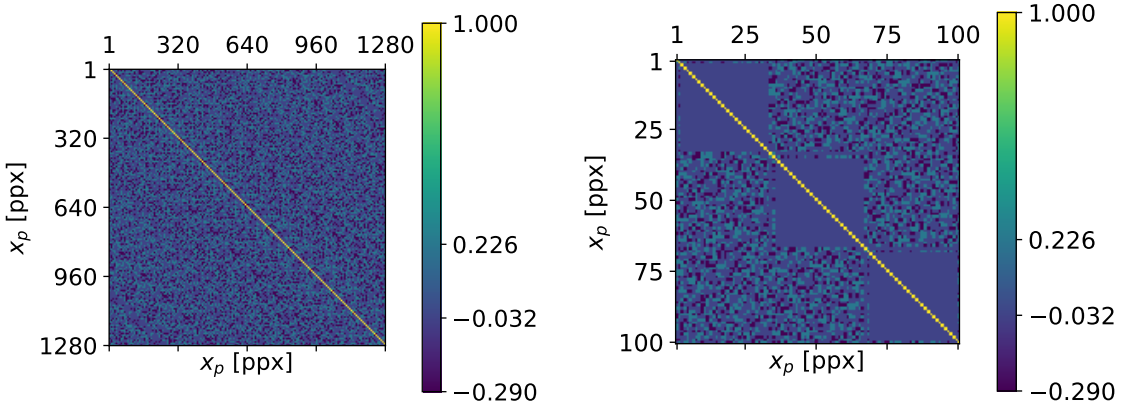
Let $\mathbf{G} = \begin{bmatrix} \mathbf{g_1} & \mathbf{g_2} & \cdots & \mathbf{g_{2^m+1}} \end{bmatrix}$ be the matrix containing the set of Gold codes originating from a preferred pair of length $2^m - 1$. Since a set of Gold codes has good cross-correlations for any lag τ, an extended code matrix can be formed containing all the circularly right shifted permutations of $\mathbf{G}$ according to

$$\mathbf{G_+} = \begin{bmatrix} \mathbf{R}_{\mathbf{2^m-1}}^{\mathbf{0}}\mathbf{G} & \mathbf{R}_{\mathbf{2^m-1}}^{\mathbf{1}}\mathbf{G} & \cdots & \mathbf{R}_{\mathbf{2^m-1}}^{\mathbf{2^m-2}}\mathbf{G} \end{bmatrix} \tag{35}$$

For $m = 5$, one can construct $2^5 + 1 = 33$ Gold codes each of length $2^5 - 1 = 31$. Then $\mathbf{G_+}$ will contain $33 \cdot 31 = 1023$ codes in total. In the case of the Zivid Two camera, this is less than the number of projector columns $X_P = 1280\,\text{ppx}$, and insufficient for the use in this camera. Of course, one could use $m = 6$ instead and obtain codes of length $2^6 - 1 = 63$. But capturing 63 images takes a lot of time, and is therefore not considered any further. Another way of solving this issue would be to simply repeat some of the codes twice and filter them out using the projector column distance constraint from Chapter 5. This is exactly what is done in (36).

$$\mathbf{C_G} = \left(\begin{bmatrix} \mathbf{G_+} & \mathbf{G_+} \end{bmatrix}_{ix_p} \right)_{\substack{1 \le i \le 2^m - 1 \\ 1 \le x_p \le X_P}} \quad \text{constrained by} \quad X_p \le 2 \cdot (2^m + 1) \cdot (2^m - 1) \tag{36}$$

The code matrix for these patterns using $m = 5$ is visualized in Figure 31a. The normalized covariance matrix for $\mathbf{C_G}$ is available through



(a) The full covariance matrix.

(b) Zoomed covariance matrix for the first 100×100 elements.

Figure 32: Normalized covariance matrix for the code matrix $\mathbf{C_G}$.

$$\mathrm{Cov}[\mathbf{C_G}, \mathbf{C_G}] = \frac{1}{\max\left(\mathbf{C_G^T C_G}\right)} \cdot \mathbf{C_G^T C_G} \tag{37}$$

and is provided in Figure 32a. Notice from the zoomed covariance matrix in Figure 32b that the matrix has block diagonals of nearly uncorrelated neighbors. This comes from the fact that the codes are arranged in such a way that nearby codes have the same lag and thus low covariance. The distribution of the values of the normalized covariance matrix is shown in Table 1.

Covariance value	**Occurrence**
1.0	0.10%
0.23	30.2%
-0.032	51.6%
-0.29	18.1%

Table 1: Distribution of normalized covariance values for $\mathbf{C_G}$.

Each code occupies a single column in $\mathbf{C_G}$, which corresponds to having a fringe width $W_F = 1\,\mathrm{ppx}$. This code matrix can be used to develop new code matrices $\mathbf{C_{G(W_F)}}$ of same dimensions $(2^m - 1) \times X_P$ with greater fringe widths $W_F > 1\,\mathrm{ppx}$. Each entry in such matrices can be derived using (38).

$$\left(\mathbf{C_{G(W_F)}}\right)_{ix_p} = (\mathbf{C_G})_{i\lfloor \frac{x_p}{W_F} \rfloor} \quad \forall \quad W_F > 1\,\mathrm{ppx} \tag{38}$$

An example of this is illustrated with $W_F = 10\,\mathrm{ppx}$ in Figure 31b. It is apparent from the figure that each code now occupies 10 ppx instead of 1 ppx.

7.1.2.2 Decoding

The decoding algorithm will be utilizing the projector column distance constraint as derived in Chapter 5. This will be used to filter out which codes from $\mathbf{C_{G(W_F)}}$ are valid for each particular camera pixel (x_c, y_c). Since only a small fraction of all codes are valid, the decoding algorithm will gain a significant reduction in computational complexity, reducing the decoding time accordingly. Also, the constraint will mean that reflections coming from outside of the valid projector columns will not be considered in the decoding, making it more resilient to interreflections. The correlation between the sequence captured for a particular camera pixel and all the valid codes are first calculated. Then the two codes with the highest correlation will be considered the possible solutions. It is beyond the scope of this thesis to figure out the single correct solution, as this would require the application of a properly tuned gradient filter according to Section 3.2.

First, the upper and lower bounds Z_C^U and Z_C^L of the camera distance have to be defined according to the range of distances to the objects present in the scene. The equations from Chapter 5 should be used to calculate $\mathbf{p^U}$, $\mathbf{p^L}$ and $\mathbf{d}$ for these distance constraints. There are in total $\lceil \frac{X_P}{W_F} \rceil$ distinct codes in the code matrix. The vector $\mathbf{d}$ stores the ratio of valid codes for each camera column x_c, which means that $\max(\mathbf{d})$ should be the highest ratio of valid codes for any camera column. Therefore, at most $N_C = \left\lceil \frac{X_p}{W_F} \cdot \max(\mathbf{d}) \right\rceil$ codes can be visible for any camera pixel.

All camera pixels are decoded independently, which means that the decoding algorithm is massively parallelizable. In order to take full advantage of this, all of the calculations should be done using tensors and matrices. The first step is to create a tensor $\mathbf{A}$ which stores all valid codes for each camera column x_c. It should be constructed in such a way that $(\mathbf{A})_{x_c}$ is a code matrix containing all the valid codes for that particular x_c. In the case that $W_F > 1\,\text{ppx}$, multiple columns in the originating code matrix $\mathbf{C_{G(W_F)}}$ will contain the same code. As $\mathbf{A}$ later will be used directly in computations, it is desirable that each entry $(\mathbf{A})_{x_c}$ contains *unique* codes only to decrease computational complexity. Therefore, duplicate codes should not be stored in $(\mathbf{A})_{x_c}$. Also, the number of valid codes varies depending on x_c. By that reason, the tensor $\mathbf{A}$ should be of such dimensions that it can store the highest amount of possible valid codes N_C. Initialize an all-zero tensor $\mathbf{A}$ of dimensions $X_C \times N_P \times N_C$, where $N_P = 2^m - 1$ specifies the number of patterns in the code matrix. Then, assign

$$(\mathbf{A})_{x_c} = \left\{ \left(\mathbf{C_{G(W_F)}} \right)_{i x_p} \right\}_{\substack{1 \le i \le N_P \\ (\mathbf{p^L})_{x_c} \le x_p \le (\mathbf{p^U})_{x_c}}} \tag{39}$$

Now $(\mathbf{A})_{x_c}$ contains all the unique codes which are valid for each camera column x_c. An example of $(\mathbf{A})_{x_c}$ for $x_c = 400\,\text{cpx}$ using $\mathbf{C_G}$ is provided in Figure 33.

Consider the sequence of captured images using the code matrix $\mathbf{C_{G(W_F)}}$ in which the intensities have been normalized according to (7). These can be stored in a tensor $\mathbf{M_{G(W_F)}}$ of dimensions $Y_C \times X_C \times N_P$. Then the intensity captured for the pattern $(\mathbf{C_{G(W_F)}})_i$ in the camera pixel (x_c, y_c) is stored in $(\mathbf{M_{G(W_F)}})_{y_c x_c i}$. As the

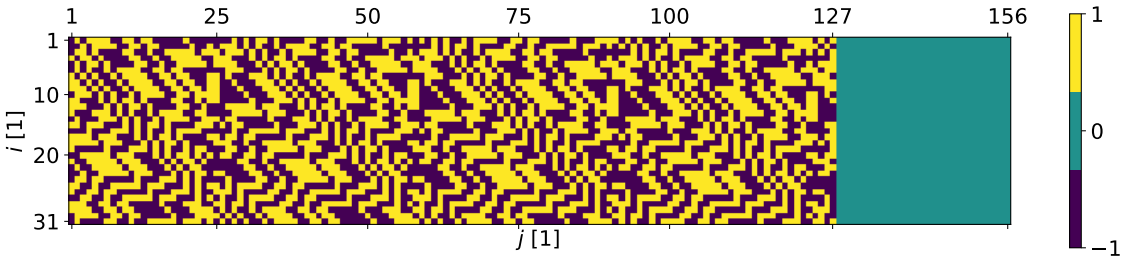


Figure 33: Example of $(\mathbf{A})_{x_c}$ for $x_c = 400\,\text{cpx}$ using $\mathbf{C_G}$.

intensity ranges from zero to one and the codes are bipolar, the intensities should first be mapped to the range from -1 to 1 by subtracting $1/2$ and then multiplying by 2. All valid codes for the camera column x_c are stored in $(\mathbf{A})_{x_c}$, so the correlation becomes the matrix product between the bipolar version of $\left(\mathbf{M}_{G(W_F)}\right)_{y_c x_c}$ and $(\mathbf{A})_{x_c}$. All this can be written in Einstein notation, as shown in (40).

$$(\mathbf{T})_{y_c x_c} = 2 \cdot \left[\left(\mathbf{M_{G(W_F)}}\right)_{y_c x_c} - \frac{1}{2}\right] \cdot (\mathbf{A})_{x_c} \tag{40}$$

Now, $\mathbf{T}$ is a tensor of dimensions $Y_C \times X_C \times N_C$ in which the entry $(\mathbf{T})_{y_c x_C}$ is a vector storing the correlation values for all the valid codes in camera pixel (x_c, y_c).

Lastly one needs to find the two *highest* correlation values for each camera pixel. As seen below, $(\mathbf{B})_{y_c x_c}$ contains the fringe number of the valid solutions for camera pixel (x_c, y_c) relative to the lower bound of the projector column distance constraint $(\mathbf{p^L})_{x_c}$. Equation (41) maps these solutions to absolute projector column values by first multiplying with W_F to convert relative fringe numbers to relative pixels, and then adding the lower bound of the distance constraint.

$$\begin{aligned}
(\mathbf{B})_{y_c x_c 1} &= \underset{i \in [1, N_P] \subset \mathbb{N}}{\arg\max} \; (\mathbf{T})_{y_c x_c i} \\
(\mathbf{B})_{y_c x_c 2} &= \underset{i \in [1, N_P] \subset \mathbb{N} \setminus (\mathbf{B})_{y_c x_c 1}}{\arg\max} \; (\mathbf{T})_{y_c x_c i} \\
(\mathbf{D})_{y_c x_c} &= W_F \cdot (\mathbf{B})_{y_c x_c} + \left(\mathbf{p^L}\right)_{x_c} - \left(\left(\mathbf{p^L}\right)_{x_c} \bmod W_F\right)
\end{aligned} \tag{41}$$

7.1.3 Limitations

While the patterns appear to handle interreflections well by using correlation properties, they also have their limitations. The first of its shortcomings is related to its resilience to blur. As seen in Figure 31, the neighboring codes are quite dissimilar. This in turn will mean that the patterns contain mostly spatially high frequency content. According to the distance response depicted in Figure 29, such high spatial frequencies are significantly dampened when the lenses of the system are not in focus. Therefore, it is expected that having a low fringe width W_F would make the patterns fail when out of focus. Also, its discrete nature limits its resolution.

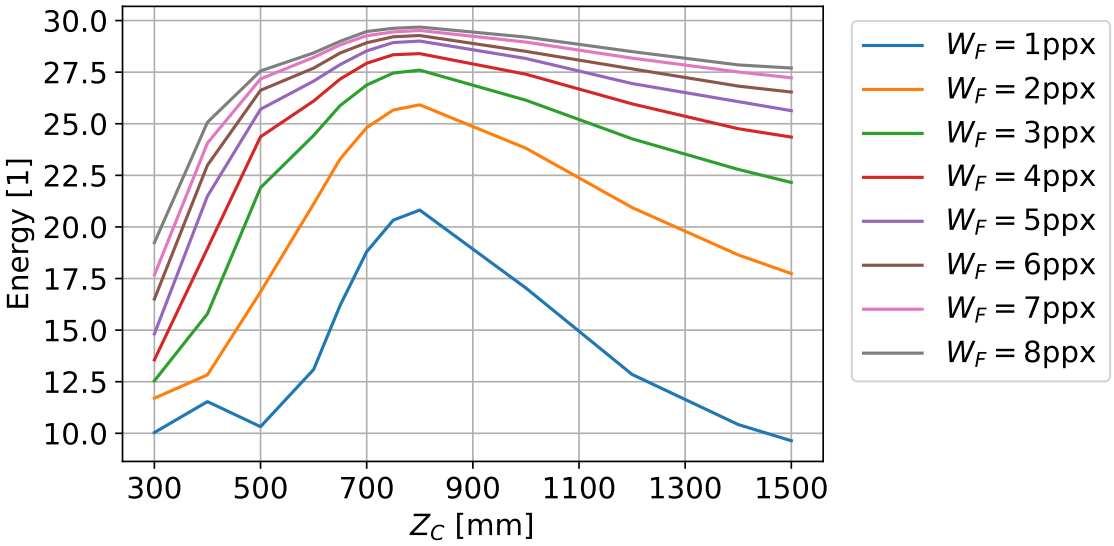


Figure 34: Energy plot of $\mathbf{C_G^{5,W_F}}$ for a range of W_F.

Only integer projector columns can be identified in the decoded signal, and this is significantly worse than the sub-pixel accuracy of GCPS.

The increase in the width of the fringe W_F is expected to lead to higher defocus resilience. Making each code word occupy more pixels will in turn make the patterns contain lower spatial frequencies. According to the distance response shown in Figure 29, these frequencies are less dampened due to lens defocus. The increase in fringe width does unfortunately come at the cost of worsening the resolution even further, as the projector columns are identified only up to an integer multiple of the fringe width. Having a fringe width $W_F = 10\,\text{ppx}$ would, for instance, mean that projector columns can be identified only up to a 10 ppx accuracy.

7.1.4 Tuning

When a certain code family (such as the Gold codes) has been chosen, there are only a few parameters which can be adjusted. The first is the order in which the codes appear code matrix. Referring to the project thesis (Lima-Eriksen 2022), there are particularly two useful ways of ordering the codes. They could be grouped either by lag or they could be grouped by similar neighboring code values. The former is what has been done in this thesis, and is known as *Grouped Gold* from the project thesis. Here, the 31 distinct codes are placed next to each other, and are repeated with incremental lags of 1. As the codes are made in such a way that they are almost orthogonal to each other, this leads to low covariance values within each lag group. The phenomenon is visualized by the blocks of low values along the diagonal of the covariance matrix, as seen previously in Figure 32. An alternative ordering is to place codes in such a way that the number of similar neighbors are maximized. This is what was done for the *Low-pass Gold* in the project thesis. Ordering outperformed *Grouped Gold* in all metrics, and this could probably be attributed to the fact that the patterns contained lower frequency content. However, this ordering comes with a

disadvantage. When similar codes are placed adjacent to each other, the covariance values of the spatially close codes become quite high. Interreflections often occur between projector columns that are spatially close, as they typically originate from reflections caused by neighboring objects. Therefore, it is assumed that it is best to organize the codes in such a way that spatially close codes have low covariance values as seen in the *Grouped Gold* arrangement.

Defocus issues should instead be handled by increasing the fringe width W_F, which decreases spatial frequencies without affecting the covariance values for spatially close codes. Patterns with fringe widths $W_F \in \{1\,\text{ppx}, 2\,\text{ppx}, \cdots, 8\,\text{ppx}\}$ can be made according to the algorithm in Section 7.1.2.1. These can then be rendered using the *diffuse plane* scene using the distance sweep $\mathbf{z_{rough}}$. For each of these renders, the performance of the patterns is measured through its average energy. The results of the renderings are summarized in Figure 34. Each curve represents the average energy of the signal over a range of distances along the x-axis. Increasing W_F should lead to lower spatial frequencies, and this is confirmed by comparing the overall shape of these curves with those of the distance response shown in Figure 29. Notice from Figure 34 that the average energy with $W_F = 1\,\text{ppx}$ is at most 20.0 when the distance Z_C is close to the focus distance (800 mm), but it drops sharply outside of the focus. From the project thesis, it was revealed that the correlation-based patterns with $W_F = 1\,\text{ppx}$ work within focus, but fails outside. By increasing the fringe width to $W_F = 3\,\text{ppx}$ the average energy is greater than 20.0 for all distances $Z_C \in [500\,\text{mm}, 1500\,\text{mm}]$. As the patterns worked well when the average energy was above 20.0, these observations together suggest that the correlation-based patterns should work well as long as the fringe width $W_F \geq 3\,\text{ppx}$.

7.2 Permuted phase shifts

The correlation-identified fringes cannot decode the projector columns to a high enough accuracy due to its energy loss caused by lens defocus. Increasing the fringe width increases the energy, but at the expense of the accuracy. Additional patterns having high accuracy are needed.

The correlation-identified fringes can in some ways be compared with the gray codes in GCPS. They are both binary and discrete, meaning that they can never achieve subpixel accuracy. Also, they only work sufficiently well when having a fringe width $W_F > 1\,\text{ppx}$. As seen in Section 3.1, phase shifts handle all distortions but interreflections really well. In addition, they have the desired sub-pixel accuracy. This begs the question whether phase shifts can be modified in such a way that it also behaves nicely in the presence of interreflections. This will be further investigated in the following

7.2.1 Working principles

Recall from Section 3.1 that the major issue with using phase shifts in structured light is the inseparability of superpositioned cosines. Consider two sampled cosines

$s[n] = A\cos(2\pi f_t \cdot n - \alpha)$ and $w[n] = B\cos(2\pi f_t \cdot n - \beta)$ with the same frequency f_t, but different phase shifts $\alpha \neq \beta$. The superpositioning $i[n] = s[n] + w[n]$ of these two signals becomes a new cosine with the same frequency f_t, but different phase shifts:

$$\begin{aligned} i[n] &= s[n] + w[n] \\ &= A\cos(2\pi f_t \cdot n - \alpha) + B\cos(2\pi f_t \cdot n - \beta) \\ &= C\cos\left(2\pi f_t \cdot n + \zeta\right), \quad \begin{cases} C = \sqrt{[A\cos(\alpha) + B\cos(\beta)]^2 + [A\sin(\alpha) + B\sin(\beta)]^2} \\ \zeta = \arctan\left[\frac{A\sin\alpha + B\sin\beta}{A\cos\alpha + B\cos\beta}\right] \end{cases} \end{aligned}$$

Let now the vectors

$$\begin{aligned} \mathbf{s} &= \begin{bmatrix} s[0] & s[1] & \cdots & s[N_P - 1] \end{bmatrix}^T \\ \mathbf{w} &= \begin{bmatrix} s[0] & s[1] & \cdots & s[N_P - 1] \end{bmatrix}^T \end{aligned}$$

contain the first N_P samples of these two signals. It can be shown (Hung 2000) that the phase ζ of the sampled $\mathbf{i} = \mathbf{s} + \mathbf{w}$ is calculated through

$$\zeta = \text{arctan2}\left\{\sin\left(\mathbf{\Phi}_{\mathbf{N_P}}^{\mathbf{T}}\right) \cdot \mathbf{i}, \quad \cos\left(\mathbf{\Phi}_{\mathbf{N_P}}^{\mathbf{T}}\right) \cdot \mathbf{i}\right\}$$

where $\mathbf{\Phi}_{\mathbf{N_P}}$ is defined according to (8). Drawing the analogy to structured light, let $\mathbf{s}$ denote the temporal signal containing the direct reflection in a particular camera pixel and $\mathbf{w}$ denotes an interreflection. It is then apparent that an interreflection will cause a phase distortion.

There is, however, one exploitation that can be made here. Due to the phase-ambiguity problem, the phase shifts are typically used in a fringed setting, where the receiver has already identified the originating fringe for each camera pixel. Consider the case where it is known that the camera pixel (x_c, y_c) receives its direct reflection from fringe number 1. Let also the interreflection originate from fringe number 2, but this is unknown. Now, associate the permutation matrices $\mathbf{S_1}$ and $\mathbf{S_2}$ with fringe numbers 1 and 2, respectively. A permutation matrix $\mathbf{S}$ is a special type of matrix which changes the order of the entries of a vector when it is premultiplied to it. Instead of projecting the signals $\mathbf{s}$ and $\mathbf{w}$, these signals will first be pre-multiplied with their corresponding permutation matrices. This means that $\mathbf{s'} = \mathbf{S_1 s}$ and $\mathbf{w'} = \mathbf{S_2 w}$ are the projected signals from the direct reflection and the interreflection, respectively. The captured signal in camera pixel (x_c, y_c) will then be

$$\begin{aligned} \mathbf{i'} &= \mathbf{s'} + \mathbf{w'} \\ &= \mathbf{S_1 s} + \mathbf{S_2 w} \end{aligned}$$

It is already known that fringe number 1 causes the direct reflection, and so the captured signal should be pre-multiplied with the transpose of the corresponding permutation matrix to recover the signal:

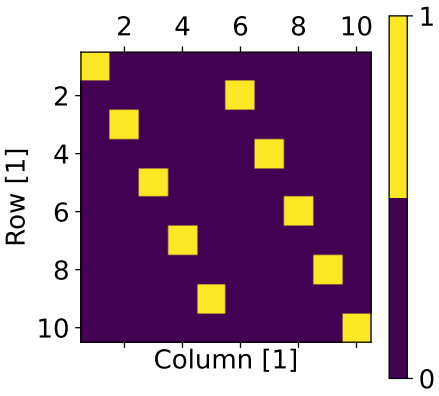
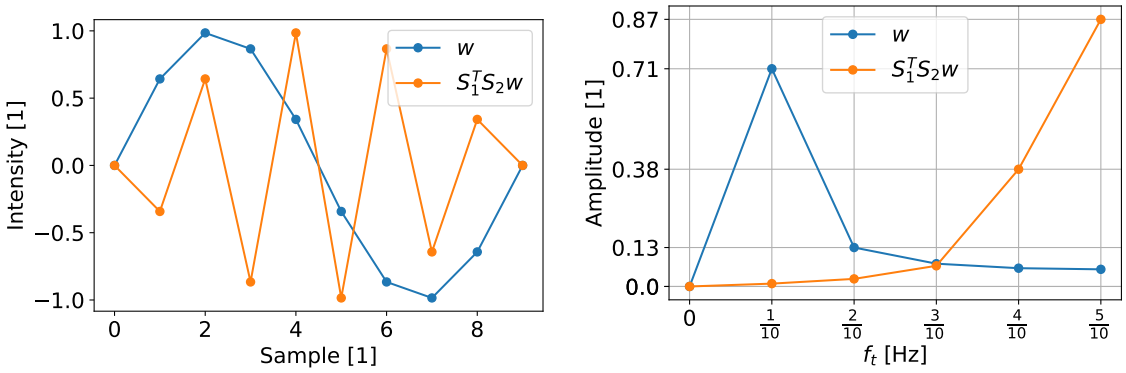


Figure 35: One ideal product of permutation matrices $\mathbf{S_1}^T\mathbf{S_2}$.



(a) The samples of $\mathbf{w}$ with and without the permutation.

(b) The DFT of the samples of $\mathbf{w}$ with and without the permutation

Figure 36: Example of how the permutation of a sequence of a pure cosine affects its frequency components.

$$\begin{aligned}\mathbf{i} &= \mathbf{S_1}^T \cdot (\mathbf{S_1 s} + \mathbf{S_2 w}) \\ &= \mathbf{s} + \mathbf{S_1^T S_2 w}\end{aligned} \tag{42}$$

Several observations can be made from (42). First, if no interreflection is present ($B = 0$), then the equation reduces to $\mathbf{i} = \mathbf{s}$. Consequently, the estimator is unbiased. Also, notice how the term $\mathbf{S_1^T S_2}$ appears only in front of the signal from the interreflection. By selectively choosing $\mathbf{S_1}$ and $\mathbf{S_2}$, the effects of $\mathbf{w}$ on the estimation of ζ can be minimized.

As an example, consider the case where $\mathbf{S_1}$ and $\mathbf{S_2}$ are chosen such that the product $\mathbf{S_1^T S_2}$ of the two are as depicted in Figure 35. Next, define the noise vector $\mathbf{w}$ as one period of a cosine with a period of 10 samples. This is depicted as the blue curve in Figure 36a. By pre-multiplying $\mathbf{w}$ with $\mathbf{S_1^T S_2}$, the samples obtained will be as indicated by the orange curve in Figure 36a. The DFT can be calculated for them both, and has been visualized in Figure 36b. From this plot, important observations can be made. Notice how $\mathbf{w}$ contains only the frequency $\frac{1}{10}$Hz. By pre-multiplying with the permutation, the frequency content is increased to mostly

containing the frequencies $\frac{4}{10}$Hz and $\frac{5}{10}$Hz. The permuted cosine now does not contain any components of temporal frequency $\frac{1}{10}$Hz. Recall that the recovered signal $\mathbf{i} = \mathbf{s} + \mathbf{S_1^T S_2 w}$. It has been shown that $\mathbf{s}$ and $\mathbf{S_1}^T\mathbf{S_2 w}$ contain different frequencies. This should in turn mean that phase estimation of $\mathbf{s}$ by using $\mathbf{i}$ can be done without being subject to distortions by the interreflection $\mathbf{w}$. These are the working principles behind permuted phase shifts (PPS).

7.2.2 Algorithm

Encoding and decoding algorithms for permuted phase shifts have to be developed. As seen in the previous section, the decoding algorithm requires that the originating fringe had been identified in all camera pixel. The algorithms will therefore assume that this is known.

7.2.2.1 Encoding

It has been shown that permuted phase shifts makes it possible to substantially reduce the effect of interreflections if the permutation matrices are chosen correctly. This then begs the question whether it is possible to analytically find the best permutation matrices for the fringes. At first glance, it might seem reasonable, as all of the above equations are matrix and vector algebra. However, recall that $\mathbf{s}$ and $\mathbf{w}$ were vector representations of *sequences*. For this reason, while the equations themselves are linear, the mathematics they represent is certainly not linear. An extensive literature search has been performed with the purpose of investigating whether or not there exists algebra related to this kind of permutation. No usable sources were found.

Instead of finding the permutation matrices analytically, random permutations will be used. Note that the samples from each projector column come from a cosine. Therefore, the samples must come from the arcsine distribution. Intuitively, it is expected that when the samples are shuffled at random, the resulting sequences will be whitened permutations of the original cosines. Increasing the number of temporal samples of the same cosine should lead to a broader frequency spectrum in which the signal can be whitened, with less energy for each of the frequencies. Recall that the direct reflection will be a cosine of a single frequency. For that reason, this algorithm is expected to converge to a lower error as the number of temporal samples increases.

The first step of the encoding algorithm is to make a non-permuted fringe. This is a matrix consisting of all projector pixel values for each of the patterns within each fringe, but in a nonpermuted order. Before constructing the matrix, three parameters should be chosen. The fringe width W_F specifies how many projector columns each fringe should occupy. Next is the spatial frequency f_s. Recall from the phase shifts in GCPS that the phase estimation is periodic up to 2π, so f_s should be chosen in such a way that one period occupies no less than one fringe width, that is, $\frac{1}{f_s} \geq W_F$. Lastly, the number of patterns N_P has to be chosen. A non-permuted

fringe $\mathbf{F}$ of dimensions $N_P \times W_F$ with spatial frequency f_s is made according to (43), and Figure 37a shows an example of one such matrix. Notice that both rows and columns of the matrix resemble cosines.

$$(\mathbf{F})_{ij} = \sin\left(2\pi f_s \cdot j - \frac{2\pi}{N_P} \cdot i\right) \tag{43}$$

The next step of the algorithm is to generate the permutation matrices. There are in total $N_F = \left\lceil \frac{X_P}{W_F} \right\rceil$ fringes, meaning that also N_F permutation matrices are needed. There are various ways of generating these. In Python, the `numpy` package has a function `numpy.random.permutation(length)`. By using it with the `length` parameter equal to N_P, the function returns a vector of integers that span the range $[0, N_P)$ in random order. This shuffle vector $\mathbf{v_n}$ can easily be converted into a permutation matrix by first initializing an all-zero matrix $\mathbf{S_n}$ of dimensions $N_P \times N_P$ and then assigning

$$(\mathbf{S_n})_{i(\mathbf{v_n})_i} = 1 \quad \forall \quad i \in [1, N_P] \subset \mathbb{N}$$

Consider now one particular permutation matrix with $N_P = 8$ as depicted in Figure 37a. By pre-multiplying this with the fringe in Figure 37b, the permuted fringe as shown in Figure 37c is obtained. Notice how the permutation matrix changes the order of the patterns so that the matrix no longer resembles cosines along the temporal dimension (columns), but conserves the shape of a cosine in the spatial dimension (rows). Therefore, the cosine should be well preserved within the fringe, as it is still a cosine here.

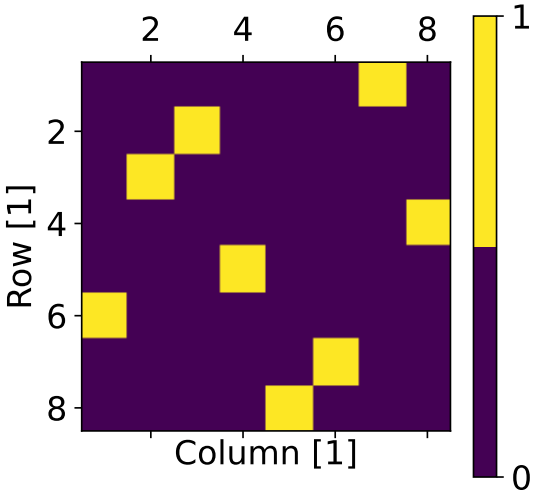
However, permutations should be applied to a whole code matrix, and not to a single fringe only. Let now additionally $X_P = 50\,\text{ppx}$, and assume that all N_F permutation matrices $\mathbf{S_n}$ have been created according to the algorithm mentioned above. Then the code matrix $\mathbf{C_{PPS}}$ consisting of the permuted phase shifts can be created according to

$$\mathbf{C_{PPS}} = \left(\begin{bmatrix}\mathbf{S_1F} & \mathbf{S_2F} & \cdots & \mathbf{S_{N_F}F}\end{bmatrix}_{ix_p}\right)_{1 \le x_p \le X_P} \tag{44}$$

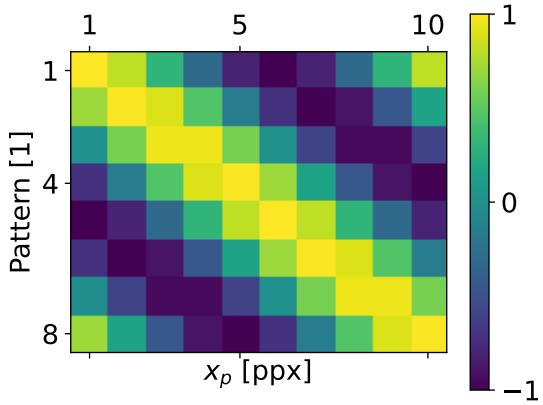
The code matrix for this particular case is illustrated in Figure 38. Notice that there is a discontinuity in the patterns in the boundary between two fringes. This comes from the fact that each fringe uses a different permutation matrix.

7.2.2.2 Decoding

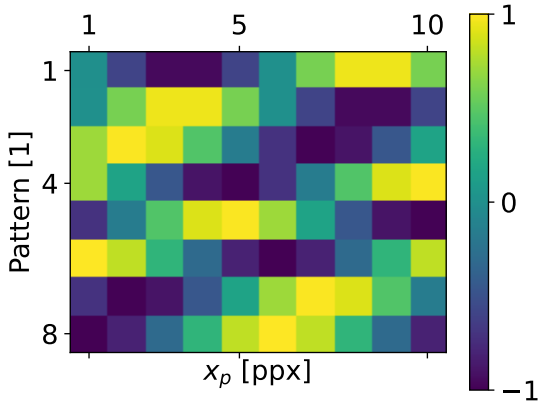
As mentioned initially, a requirement for the successful decoding is to already know which fringe is shown in each of the camera pixels. One way of accomplishing this will be explained later in Chapter 8. Assume a tensor $\mathbf{L}$ of dimensions $Y_C \times X_C \times N_P \times N_P$ such that $(\mathbf{L})_{y_c x_c}$ is the permutation matrix belonging to the fringe observed in the camera pixel (x_c, y_c). Furthermore, let $\mathbf{M_{PPS}}$ denote a tensor of



(a) Permutation matrix created by random shuffling.



(b) Non-permuted fringe matrix.



(c) Permuted fringe matrix.

Figure 37: Example of how a fringe is modified through the pre-multiplication of a permutation matrix. Constructed with parameters $f_s = \frac{1}{10}\text{ppx}^{-1}$, $W_F = 10\,\text{ppx}$ and $N_P = 8$.

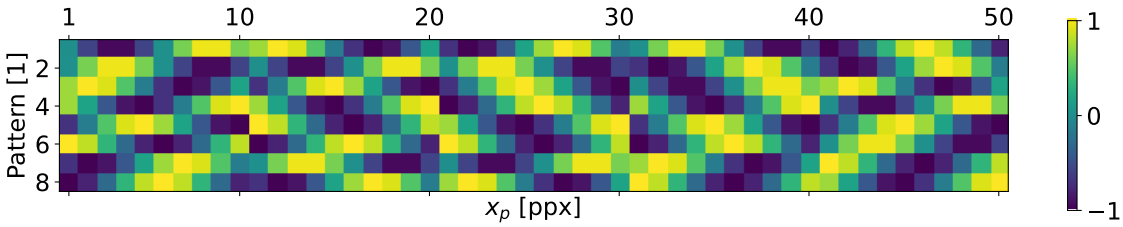
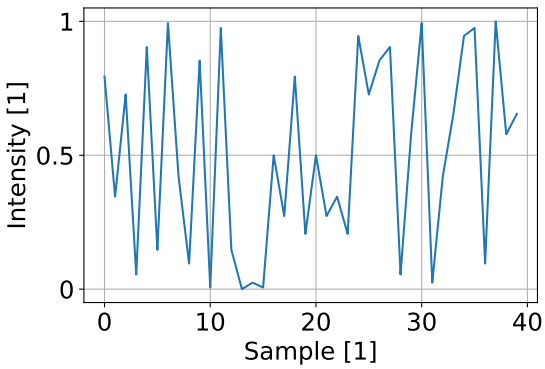
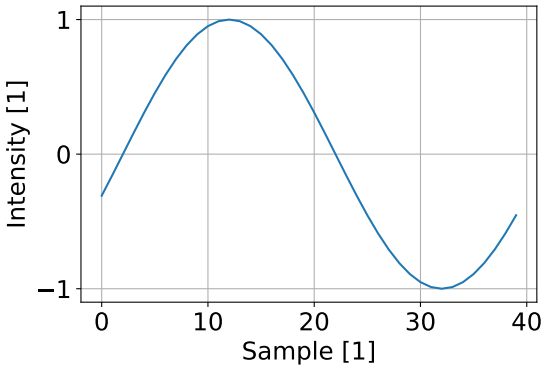


Figure 38: Code matrix for permuted phase shifts with $f_s = \frac{1}{10}\text{ppx}^{-1}$, $W_F = 10\,\text{ppx}$ and $N_P = 8$.



(a) The samples $(\mathbf{M_{PPS}})_{y_c x_c}$ of the permuted phase shift.



(b) Reconstructed phase shifts $(\mathbf{U})_{y_c x_c}$.

Figure 39: Samples of permuted phase shifts before and after reconstruction using $N_P = 40$ for a particular camera pixel (x_c, y_c).

dimensions $Y_C \times X_C \times N_P$ that stores the normalized captures of camera pixel (x_c, y_c) in $(\mathbf{M_{PPS}})_{y_c x_c}$. The vector $(\mathbf{M_{PPS}})_{y_c x_c}$ stores the permuted phase shift samples of (x_c, y_c), and one example of such a vector is visualized in Figure 39a. Notice that the curve does not resemble a cosine. To reconstruct the temporal cosines, each of these sample vectors should be premultiplied with the transpose of their corresponding permutation matrices to get the samples in their non-permuted order. Furthermore, the samples should be mapped from the range $[0, 1]$ onto the range $[-1, 1]$ to remove bias. The resulting $Y_C \times X_C \times N_P$ reconstructed sample tensor $\mathbf{U}$ is made according to (46). Each entry $(\mathbf{U})_{y_c x_c}$ should now be a vector in which the samples appear in the order that resembles a cosine. An example of this reconstruction is shown in Figure 39b, and it was performed on the samples shown in Figure 39a.

$$(\mathbf{U})_{y_c x_c} = [(\mathbf{L})_{y_c x_c}]^T \cdot \Big[2 \cdot (\mathbf{M_{PPS}})_{y_c x_c} - 1\Big] \tag{45}$$

With the reconstructed signals obtained, the phase estimation algorithm is similar to the phase shifts in GCPS, the only difference being that N_P patterns are needed instead of the typical 4 patterns. Recall that the phase unwrapping is 2π-periodic due to the nature of arctan2. The coefficient $\frac{W_F}{2\pi}$ scales the unwrapped phase to projector pixels, and the equation becomes

$$(\mathbf{Q_{PPS}})_{y_c x_c} = \frac{W_F}{2\pi} \cdot \arctan2\left\{\sin \mathbf{\Phi}_{\mathbf{N_P}}^{\mathbf{T}} \cdot (\mathbf{U})_{y_c x_c}, \quad \cos \mathbf{\Phi}_{\mathbf{N_P}}^{\mathbf{T}} \cdot (\mathbf{U})_{y_c x_c}\right\} \tag{46}$$

7.2.3 Limitations

The reordering of cosines through permuted fringes seems to mitigate a lot of the issues with interreflections. However, the patterns still suffers some issues, which will be explained in this section.

Figure 40 shows a simplified model of a scene from the camera's point of view.

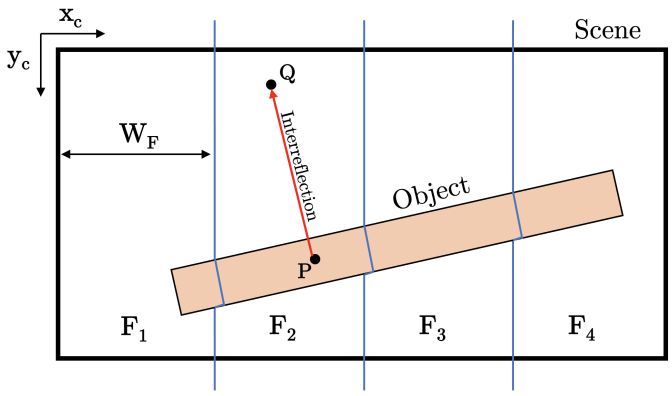


Figure 40: Intra-fringe interreflection for the permuted phase shifts patterns.

Here, the blue lines indicate the border between each of the fringes $\mathbf{F_i}$. An object (orange) causes an interreflection from $\mathbf{P}$ to $\mathbf{Q}$, with its corresponding signal $\mathbf{w'}$ observed in the camera pixel in point $\mathbf{Q}$. At the same time, the signal $\mathbf{s'}$ from the direct reflection in point $\mathbf{Q}$ is also captured in this camera pixel. These signals originate from the same fringe $\mathbf{F_2}$, and are therefore subject to the same permutation matrix $\mathbf{S_2}$. Therefore, the reconstructed signal $\mathbf{i}$ can be derived from the captured signal $\mathbf{i'}$ as follows:

$$
\begin{aligned}
\mathbf{i'} &= \mathbf{s'} + \mathbf{w'} \\
\mathbf{i'} &= \mathbf{S_2 s} + \mathbf{S_2 w} \\
\mathbf{i} &= \mathbf{S_2^T i} \\
&= \mathbf{S_2^T} \left(\mathbf{S_2 s} + \mathbf{S_2 w} \right) \\
&= \mathbf{s} + \mathbf{w}
\end{aligned}
$$

As shown above, this interreflection will *not* be whitened as previously mentioned. Instead, it will behave in the same way as for phase shifts in GCPS and cause distortions of larger magnitudes. Interreflections that originate and terminate within the same fringe are hereby known as *intra-fringe interreflections*. These are typically seen in vertical reflections such as the one shown in the model. In such reflections, the signal does not travel far in the x-direction which increases the chance of it terminating in the same fringe.

Another limitation of these patterns occurs in the fringe borders, particularly when the system is out of focus. The phenomenon is best explained through a simple visualization. Consider a particular pattern using permuted phase shifts with $f_s = \frac{1}{10}\text{ppx}^{-1}$ and two fringes. Due to the permutations, the cosines appear in different orders depending on the fringe number, which results in discontinuities at the fringe border. This discontinuity is visualized by the blue curve in Figure 41, which shows the intensity values of the projector columns. For this particular example there is a phase difference of π between the two fringes. The phase difference causes no problem when there is no lens defocus ($\sigma_{\text{PSF}} = 0\,\text{ppx}$), as no blur occurs between neighboring pixels.

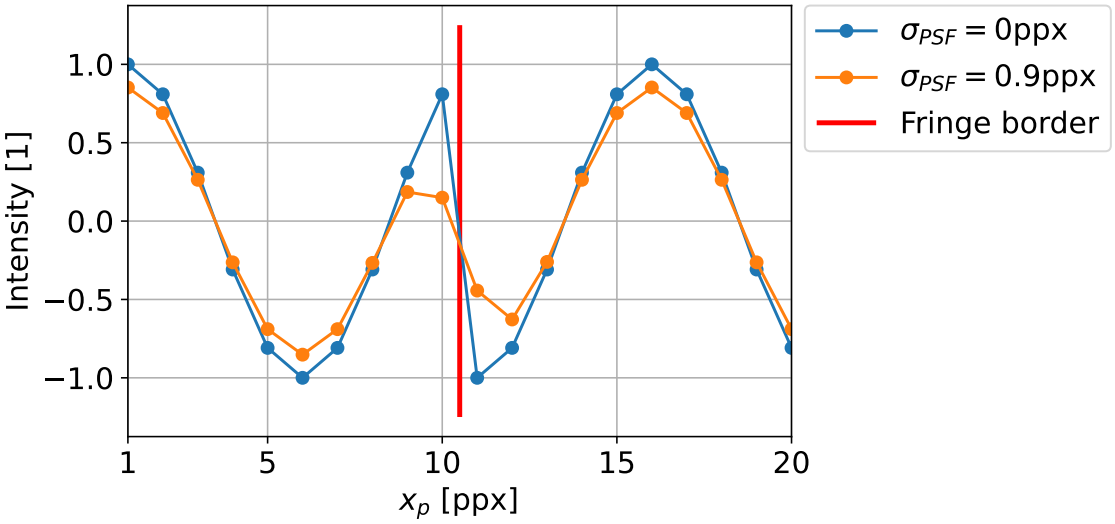


Figure 41: The spatial effects of lens defocus in the fringe borders.

However, the case is quite different when this defocus effect is introduced. Recall from Section 6.1 that the empirical PSF was found to reach values of $\sigma_{\text{PSF}} = 0.9\,\text{ppx}$ for camera distances $Z_C \overset{+}{\rightarrow} 550\,\text{mm}$ and $Z_C \overset{-}{\rightarrow} 1400\,\text{mm}$. The same pattern has also been plotted after convolving it with a Gaussian kernel with $\sigma = 0.9\,\text{ppx}$, and the resulting intensity curve is shown by the orange curve in the same plot. Notice in particular that the two pixels in the closest vicinity to the fringe border (red) are significantly altered compared to the blue (perfect-focus) curve. The phenomenon occurs due to the discontinuity at the fringe border. As previously seen, the convolution operator is equivalent to cross-talk between neighboring pixels. In contrast to the pure cosine, which is not distorted by such cross-talk, it is an issue at this fringe border. Due to the different permutation matrices, the signal from the neighboring fringe will behave just as an interreflection here. The problem is then that distortions similar to interreflections will occur even when no such reflections are present, purely due to optical phenomena which cannot be avoided. Also, it occurs at regular intervals due to the periodicity of the fringe borders. If this cross-talk is significant enough, it could result in the patterns performing even worse than the traditional phase shifts used in GCPS.

7.2.4 Tuning

As seen in the previous section, there are several limitations that could significantly affect the performance of these patterns. Therefore, great care must be taken to avoid distortions. The encoding algorithm introduced several parameters that can be adjusted to improve performance. Their effect on the distortions will be explored in this section.

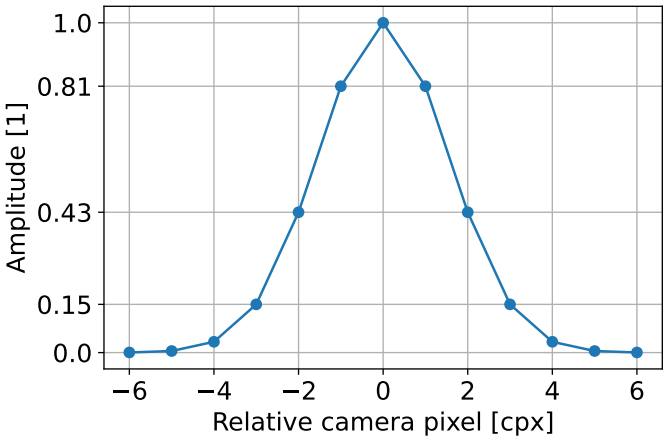


Figure 42: The kernel of the PSF with $\sigma_{\text{PSF}} = 0.9\,\text{ppx}$ as seen from the camera.

7.2.4.1 Fringe width

A particularly important parameter is the fringe width W_F, which influences the performance in many ways. First and foremost, it affects the occurrence of intra-fringe interreflections. Vertical reflections will always have a certain horizontal component, and decreasing W_F makes it more likely that a vertical reflection terminates in a different fringe.

Unfortunately, lowering the fringe width comes at a disadvantage. The total number of fringes $N_F = \left\lceil \frac{X_P}{W_F} \right\rceil$ is inversely proportional to the width of the fringe. Therefore, a low fringe width corresponds to a high number of fringes and thus a high number of fringe *borders*. The fringe borders have previously been identified as potentially problematic. Therefore, they should be kept at a minimum.

The choice of fringe width ultimately becomes a compromise between having a low enough occurrence of both intra-fringe interreflections and number of fringe borders. Some rough calculations can be made to find a good compromise. An intra-fringe interreflection has to originate and terminate within a section corresponding to the spatial width of a fringe. It is therefore of interest to find out how wide a fringe is. The projector in Zivid Two has a throw ratio of $r_t = 0.9$. At a distance Z_C, this means that the projector patterns will occupy a width of $w_{\text{proj}} = \frac{Z_C}{r_t}$. Since the projector in total has X_P pixels in this direction, each pixel would occupy a spatial width of $w_{\text{px}} = \frac{w_{\text{proj}}}{X_P}$. For the whole fringe, it would occupy

$$\begin{aligned} w_{\text{fringe}} &= W_F \cdot w_{\text{px}} \\ &= W_F \cdot \frac{Z_C}{r_t \cdot X_P} \end{aligned}$$

Consider a fringe width $W_F = 10\,\text{ppx}$ at a distance of $Z_C = 1000\,\text{mm}$ and using the numbers belonging to Zivid Two ($X_P = 1280\,\text{ppx}$ and $r_t = 0.9$). Then, $w_{\text{fringe}} \approx 8.7\,\text{mm}$. Such a span is assumed to be within the acceptable range. Next, the effects of defocus should also be considered. Figure 42 shows the kernel of the PSF

with $\sigma_{\text{PSF}} = 0.9\,\text{ppx}$ as seen from the camera. It is apparent that the PSF causes significant cross-talk from a pixel to its closest neighbor, and moderate cross-talk to its second closest neighbor. With a fringe width of $W_F = 10\,\text{ppx}$, a fringe will span $S_X W_F = 17\,\text{cpx}$. Accounting for the distortions at both sides of the fringe borders, it is estimated that approximately $\frac{2\,\text{cpx}}{17\,\text{cpx}} = 12\%$ of the camera pixels will be subject to *significant* fringe border distortions, and $\frac{2\,\text{cpx}}{17\,\text{cpx}} = 12\%$ will be subject to *moderate* fringe border distortions. Neither the resilience to intra-fringe interreflections nor the fringe border distortions seem to be particularly good or bad, which leads to the conclusion that $W_F = 10\,\text{ppx}$ seems to be a good compromise.

7.2.4.2 Spatial frequency

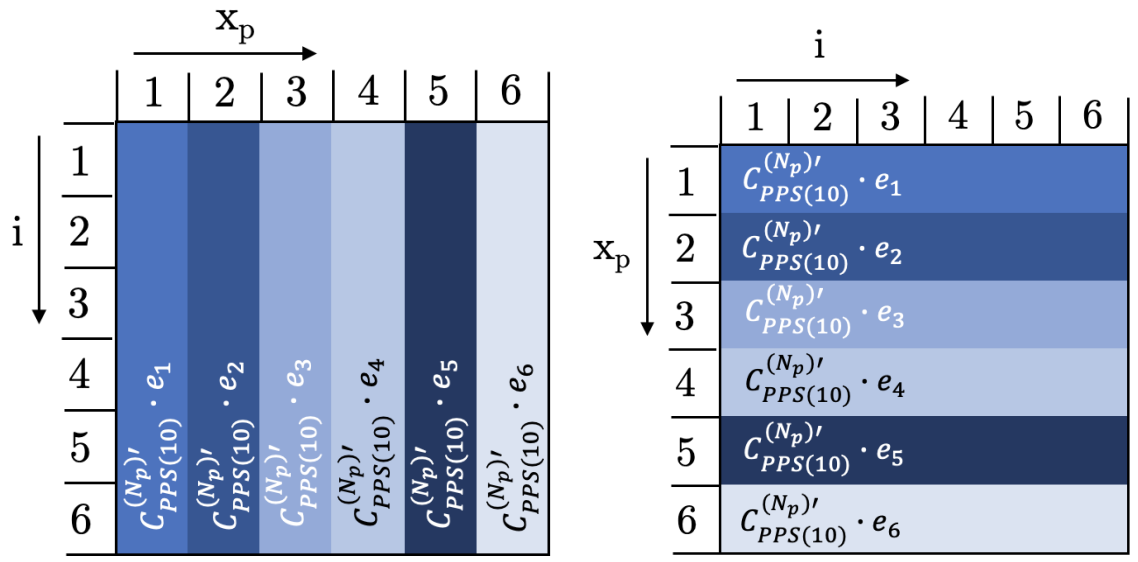
After choosing the fringe width, the spatial frequency should be decided upon. From the development of GCPS in Section 3.1, it was established that W_F places an upper bound on the spatial frequency f_s through $f_s \leq \frac{1}{W_F}$. Otherwise, phase ambiguities would arise. From the distance response shown in Figure 29, it was found that the frequencies $f_s \leq \frac{1}{8}\text{ppx}^{-1}$ should work adequately within the optimal working distance. For GCPS it was also stated that a higher spatial frequency increases the accuracy of the phase estimate. Therefore, the highest possible spatial frequency $f_s = \frac{1}{W_F} = \frac{1}{10}\text{ppx}^{-1}$ has been chosen for the configuration used in the thesis.

7.2.4.3 Pattern count

The last parameter to choose it the pattern count N_P. Through the development of the encoding algorithm, it was suggested that increasing N_P should lead to a decrease in distortions caused by interreflections. The argument that lead up to this was that it would whiten out the interreflection over a larger frequency span. Each pattern uses a finite time to capture depending on the chosen exposure time, and increasing N_P should also lead to a longer processing time due to more multiplications and additions through e.g. (46). Therefore, a compromise between error rate and time usage should be made.

Recall that the permutation matrices are randomly generated. Therefore, some pairs of permutation matrices may work better at dampening interreflections than others. In order to gain insight into how the system as a whole performs when subject to interreflections, numerical estimates will be made in the following paragraphs. There are in total X_P different projector columns where each projects its unique temporal signal. For the sake of simplicity, it is in this part assumed that at most two projector columns can terminate in the same camera pixel. This would mean that $X_P \cdot (X_P - 1)$ combinations of direct reflections and interreflections are possible. Given a certain SNR and σ_{PSF}, it is possible to estimate the effects that interreflections will have on the signals for all these combinations. This will be done below by constructing a tensor containing all of the possible combinations, and then estimating the errors for all of them.

Let $\mathbf{C}^{(\mathbf{N_P})}_{\mathbf{PPS(10)}}$ denote the code matrix using permuted phase shifts with $W_F = 10\,\text{ppx}$,



(a) The tensor $\mathbf{A_S}$.

(b) The tensor $\mathbf{A_W}$.

Figure 43: Visualizations of the tensors used for error estimates in permuted phase shifts. Each cell stores the vector as indicated in their covering rectangle.

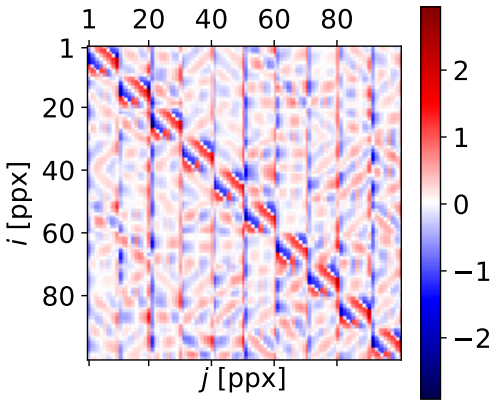
$f_s = \frac{1}{10}\text{ppx}^{-1}$ and N_P patterns. In order to account for the blurring effects caused by the PSF, a new code matrix $\mathbf{C}^{(\mathbf{N_P})'}_{\mathbf{PPS(10)}}$ of same dimensions should be derived according to

$$\left(\mathbf{C}^{(\mathbf{N_P})'}_{\mathbf{PPS(10)}}\right)_i = \left(\mathbf{C}^{(\mathbf{N_P})}_{\mathbf{PPS(10)}}\right)_i * \boldsymbol{\mathcal{N}}_{\boldsymbol{\sigma}_{\mathbf{PSF}}} \quad \forall \quad i \in [1, N_P] \subset \mathbb{N}$$

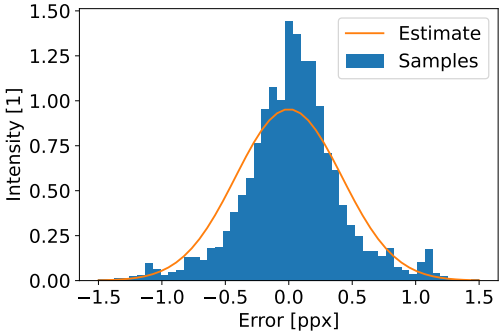
where $\boldsymbol{\mathcal{N}}_{\boldsymbol{\sigma}_{\mathbf{PSF}}}$ denotes a Gaussian kernel with SD equal to σ_{PSF}. Each pattern of this new code matrix is now blurred corresponding to the amount specified by the PSF. Define a signal tensor $\mathbf{A_S}$ of dimensions $X_P \times X_P \times N_P$ such that $(\mathbf{A_S})_{ix_p} = \mathbf{C}^{(\mathbf{N_P})'}_{\mathbf{PPS(10)}}\mathbf{e}_{\mathbf{x_p}}\, \forall i, x_p \in [1, X_P] \subset \mathbb{N}$. Also, initialize the $X_P \times X_P \times N_P$ noise tensor $\mathbf{A_W}$ to $(\mathbf{A_W})_{x_p i} = \mathbf{C}^{(\mathbf{N_P})'}_{\mathbf{PPS(10)}}\mathbf{e}_{\mathbf{x_p}}\, \forall i, x_p \in [1, X_P] \subset \mathbb{N}$. The noise tensor $\mathbf{A_W}$ is then constructed in such a way that $(\mathbf{A_W})_1$ contains N_P duplicates of the first code in the blurred code matrix, $(\mathbf{A_W})_2$ contains N_P duplicates of the second code in the blurred code matrix, etc. The signal tensor is simply the transpose of the noise tensor, that is, $(\mathbf{A_S})_{ij} = (\mathbf{A_W})_{ji}$. The tensors have been visualized in Figure 43 to illustrate this. For a certain SNR, a combination tensor $\mathbf{B}$ of dimensions $X_P \times X_P \times N_P$ is constructed according to

$$(\mathbf{B})_{ij} = \frac{1}{\text{SNR}+1} \cdot \left[\text{SNR} \cdot \left(\mathbf{A}^{\mathbf{S}}\right)_{ij} + \left(\mathbf{A}^{\mathbf{W}}\right)_{ij}\right]$$

Each cell $(\mathbf{B})_{ij}$ will now contain the superpositioning of a signal coming from the projector column j and noise coming from the projector column i with the correct PSF and SNR taken into account. Each of these cells should be decoded according to the decoding algorithm developed in Section 7.2.2.2 by assuming that $(\mathbf{B})_{ij}$ ori-



(a) The residual matrix $\mathbf{E}$.



(b) Histogram of the residual matrix $\mathbf{E}$ with a Gaussian fit superpositioned.

Figure 44: The residuals obtained with permuted phase shifts using SNR $= 2$, $\sigma_{\text{PSF}} = 0.9\,\text{ppx}$ and $N_P = 20$.

ginates from fringe number $\left\lfloor \frac{j}{W_F} \right\rfloor$. The decoded results are stored in a matrix $\mathbf{Q}$ of dimensions $X_P \times X_P$ such that $(\mathbf{Q})_{ij}$ contains the decoded pixel where the signal comes from projector column j and the interreflection comes from projector column i. This whole procedure should then also be done for SNR $\to \infty$ and $\sigma_{\text{PSF}} = 0$, with the decoded results stored in a matrix $\mathbf{Q}'$. With both the output and ground truth at hand, the residual matrix is found through calculating the difference between the two:

$$\mathbf{E} = \mathbf{Q} - \mathbf{Q}'$$

An example of such a residual matrix is visualized in Figure 44a using SNR $= 2$, $\sigma_{\text{PSF}} = 0.9\,\text{ppx}$ and $N_P = 20$. Notice first that the matrix has block diagonals of dimensions $W_F \times W_F$ with values higher than the off-diagonals. These represent the residuals observed in intra-fringe interreflections, as they correspond to the superpositioning of two signals originating from the same fringe. The residuals are further visualized through a histogram shown in Figure 44b. Centered around zero, the residuals indicate that the estimator is unbiased even in the presence of noise. Also, notice how the residuals seem to have a distribution close to a Gaussian. The standard deviation $\sigma_{\text{error}} = 0.42\,\text{ppx}$ has been estimated from the samples and is used to form the Gaussian fit indicated by the orange curve in the same plot. The fit is more heavy-tailed than the distribution from the estimate, which means that the SD will overestimate the variability in the errors. Nevertheless, the standard deviation σ_{error} of the error matrices will be used as a metric for the residuals of the permuted phase shifts.

To recap, an algorithm has been developed for finding the standard deviation of the residuals occurring in permuted phase shifts given a certain SNR, σ_{PSF} and N_P. It is then of interest to visualize for pairs of SNR and σ_{PSF} how σ_{error} will be affect over a range of N_P. This algorithm is used to make the plots shown in Figure 45. The baseline visualized through its dashed blue line is made through the same algorithm as stated above with the difference being that all of the permutation matrices are

identity matrices, meaning that the fringes will not be shuffled at all.

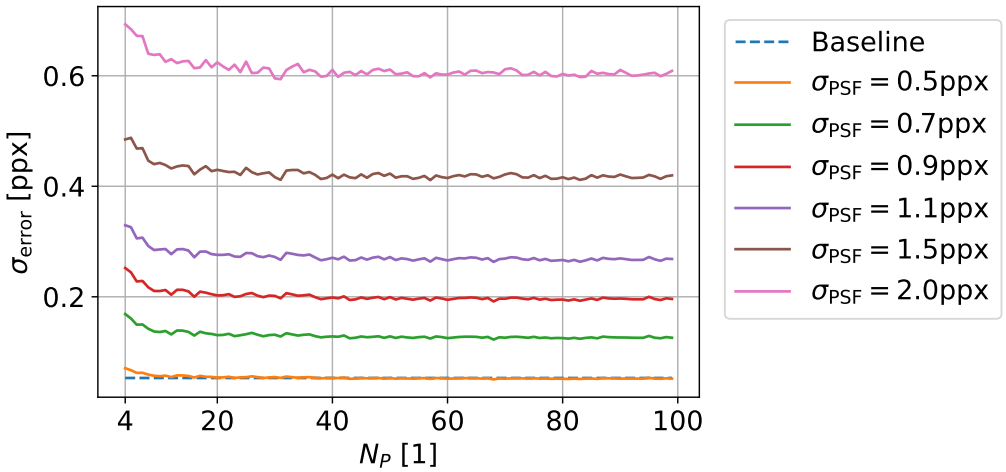
Several observations can be made from the plots. Consider first the case when no interreflections are present (SNR $\to \infty$) as depicted in Figure 45a. As expected the baseline has zero error in its estimate. However, the curve for $\sigma_{\text{PSF}} = 0.5\,\text{ppx}$ is also tangent to it. This means that the algorithm decodes the patterns correctly when no interreflections are present and the lens is within its focus. Moreover, the σ_{error} increases with increasing σ_{PSF}. This is also expected, as the fringe borders experiences more blur from neighboring pixels which come from different fringes. Nevertheless, it should be noted from the plot that permuted phase shifts seem to perform worse than non-permuted phase shift when no interreflections are present, particularly within focus.

The second plot, depicted in Figure 45b, shows the performance of permuted phase shifts for SNR = 2. This corresponds to $\frac{1}{3}$ of the captured signal originating from an interreflection. From the plot, it is clear that the permuted phase shifts seem to follow a logarithmic convergence with increasing N_P. Moreover, the patterns outperforms the non-permuted phase shifts for $\sigma_{\text{PSF}} < 2.0\,\text{ppx}$. In Figure 45c, permuted phase shifts is compared to the baseline for SNR = 0.5. The case is equivalent to $\frac{2}{3}$ of the captured signal originating from interreflections. Here, the convergence is still logarithmic, but the rate of convergence is far slower.

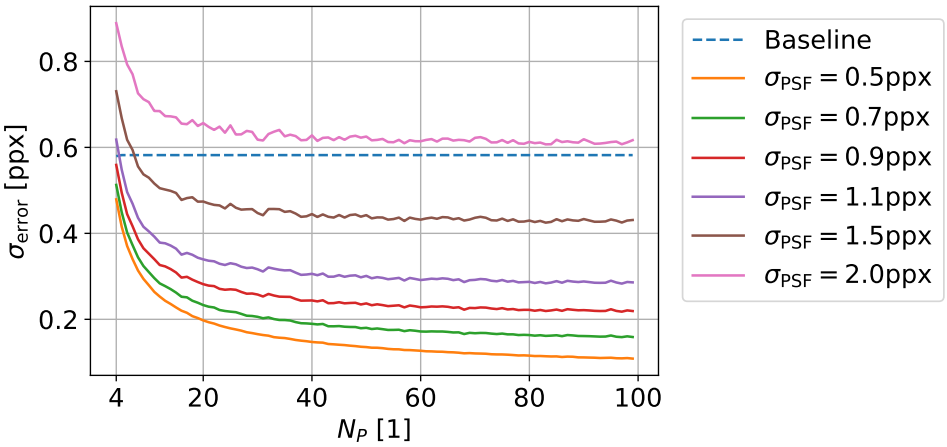
As seen from the convergence plots in Figure 45, the permuted phase shifts converges logarithmically to decreasing σ_{error} for increasing N_P. The choice of N_P cannot be made from these plots alone, as it would depend on the preceding patterns used for identifying the correct fringes. If those patterns do not function in SNR ≈ 0.5, then it is not necessary to use $N_P = 60$ patterns. By assuming SNR ≥ 2, it is reasonable to choose $N_P = 20$. The highest defocus experienced within the restriction $550\,\text{mm} \leq Z_C \leq 1400\,\text{mm}$ was previously found to be $\sigma_{\text{PSF}} = 0.9\,\text{ppx}$. From Figure 45b it is seen that little performance is gained by increasing N_P any further under those constraints. The permuted phase shifts would then typically have $\sigma_{\text{error}} = 0.3\,\text{ppx}$ for SNR = 2.0 and $\sigma_{\text{error}} = 0.2\,\text{ppx}$ for SNR $\to \infty$. Recall that the Empirical Rule states that 68% of the observations falls within one SD and 95% falls within two SD. Needless to say, most of the observations should thus reach sub-pixel accuracy under these assumptions. Also, approximately 68% of the pixels should have residuals less than 0.2 ppx under the worst defocus conditions, which is required for most applications as stated in the problem statement.

7.2.5 Fringe border filter

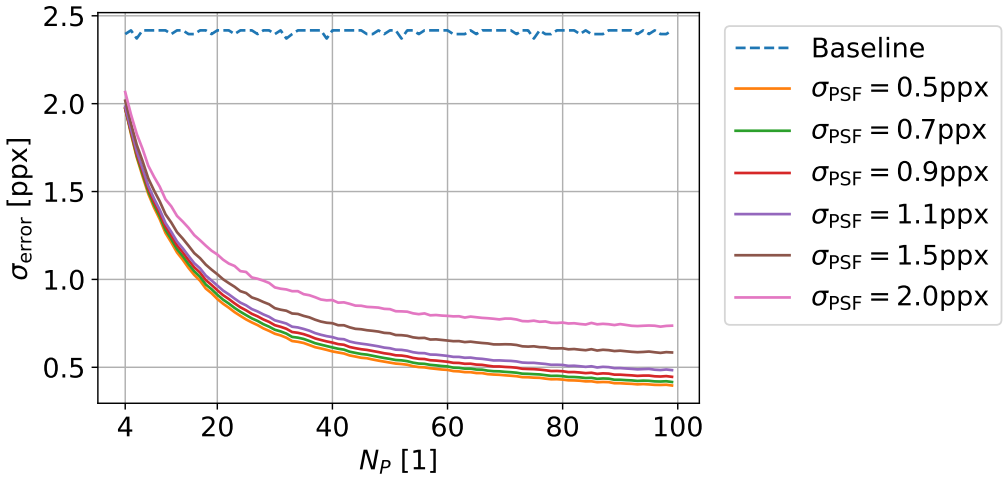
One of the limitations of this type of patterns is the fringe border distortions. These are periodic distortions that occur on the border between two fringes. Assuming that the distortions cause low residual, it should be possible to filter out these distortions to some extent. As these distortions occur on the border, this corresponds to entries $(\mathbf{Q_{PPS}})_{y_c x_c}$ that are either close to 0 or close to W_F. Therefore, it should be possible to reduce the residuals simply by keeping only the entries $T_F \leq (\mathbf{Q_{PPS}})_{y_c x_c} \leq W_F - T_F$ given a certain threshold T_F. This would of course result in fewer decoded pixels, but they are expected to have lower decoding residuals.



(a) SNR $\to \infty$



(b) SNR $= 2$



(c) SNR $= 0.5$

Figure 45: The standard deviation of the decoding error for permuted phase shifts plotted as functions of the number of patterns N_P.

8 A Novel Pattern Codification Strategy

Chapter 7 introduced two new types of patterns. The first type of pattern was the correlation-identified fringes (CIF). It utilizes sequences with strong cross- and auto-correlation properties in order to successfully decode the originating fringes even in the presence of interreflection. The main issue with this type of patterns is the compromise between resilience to defocus and its accuracy; resilience to defocus requires a large fringe width W_F, but this comes at the cost of reducing accuracy. The second type of pattern introduced in the previous chapter was permuted phase shifts (PPS). This kind of pattern offers far better accuracy than the correlation-identified fringes. Its main limitation was that it required the knowledge of each of the originating fringe for each of the camera pixels.

The attributes of the previously mentioned patterns lead to the suggestion that the correlation-identified fringes can be combined with permuted phase shifts to form a pattern codification strategy. Compared to Gray-Coded Phase Shifts, correlation-identified fringes would then correspond to the gray codes, whereas permuted phase shifts would correspond to the standard phase shifts in GCPS. In other words, correlation-identified fringes could be used to identify the originating fringe for each camera pixel. The permuted phase shifts could then use this knowledge to correctly decode all of the camera pixels. This pattern codification strategy, now known as Correlation-Fringed Permuted Phase Shifts (CFPPS), will be explored in this section. First, encoding and decoding algorithms will be developed that combine these two types of patterns. Lastly, the performance of the pattern codification strategy will be tested and evaluated using the test scenes described in Section 4.2.

8.1 Algorithm

The combination of correlation-identified fringes and permuted phase shifts places several constraints on the attributes of the patterns themselves, and therefore the parameters belonging to them must be chosen in tandem to make the patterns work together. The ultimate goal of this combination of patterns is that CIF is used to identify the fringes, and then PPS uses these fringes to decode the phases. For this to work correctly, the fringes should be the same for both CIF and PPS. Recall from their respective tuning sections that CIF required $W_F \geq 3\,\text{ppx}$ and PPS required $W_F = 10\,\text{ppx}$. Therefore, $W_F = 10\,\text{ppx}$ should work well for CFPPS. Moreover, the spatial frequency $f_s = \frac{1}{W_F} = \frac{1}{10}\text{ppx}^{-1}$ was found to work well for the permuted phase shifts. The last parameter to be chosen is N_P for the permuted phase shifts. Recall from the tuning section for PPS that SNR > 2 meant that $N_P = 20$ should be sufficient.

With all the parameters chosen, the code matrices for each of CIF and PPS can be made according to their encoding algorithms previously described. Stacking the two matrices on top of each other leads to the combined code matrix for this particular realization of CFPPS, which is shown in Figure 46. Notice that the patterns belonging to the CIF part appear first in the code matrix. The PPS

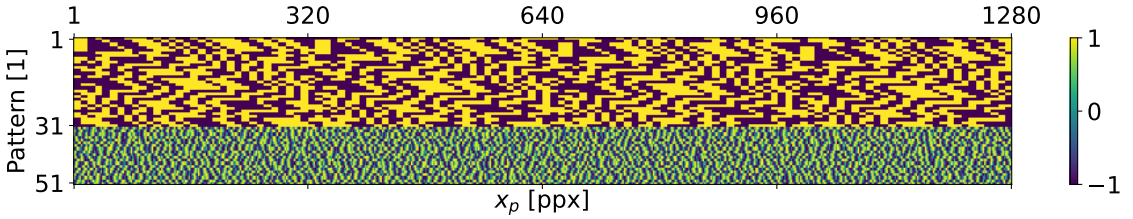


Figure 46: The code matrix $\mathbf{C_{CFPPS}}$ for Correlation-Fringed Permuted Phase Shifts using $W_F = 10\,\text{ppx}$ and $N_P = 20$.

patterns requires the decoding of CIF. By first projecting the CIF patterns, they can be decoded while the PPS patterns are projected and captured.

A decoding algorithm for the entire pattern codification strategy should also be developed. It is largely based on the decoding algorithms for CIF and PPS respectively. The only missing connection between them is to use the decoded fringes from the correlation-identified fringes to choose the right permutation matrices in the permuted phase shifts. From CIF, the decoding algorithm leads to a tensor $\mathbf{D}$ which stored the two possible solutions for the camera pixel (x_c, y_c) in $(\mathbf{D})_{y_c x_c}$. As mentioned previously, finding *the* correct solution will not be considered. Instead, the ground truth as explained in Chapter 4 will be used to find the solution that is closest to the ground truth and store it in a $Y_C \times X_C$ matrix $\mathbf{Q_{CIF}}$. Notice then that a $Y_C \times X_C$ matrix $\mathbf{R}$ defined by $(\mathbf{R})_{y_c x_c} = \left\lfloor \frac{1}{W_F} (\mathbf{Q_{CIF}})_{y_c x_c} \right\rfloor$ will contain the fringe numbers for all camera pixels. The tensor $\mathbf{L}$ required for decoding PPS is then constructed according to

$$(\mathbf{L})_{y_c x_c} = \mathbf{S}_{(\mathbf{R})_{y_c x_c}}$$

Decoding the permuted phase shifts captures results in a matrix $\mathbf{Q}$. The entire decoded solution will be the sum of the two solutions:

$$\mathbf{Q_{CFPPS}} = \mathbf{Q_{CIF}} + \mathbf{Q_{PPS}}$$

8.2 Tests

The CFPPS pattern codification strategy has been developed and should be tested to analyze its performance. The codification strategy will first be used on the *Diffuse plane* scene. Here, it will be analyzed under ideal circumstances, mainly focusing on how the defocus affects the performance at various distances. Lastly, the *Objects in bin* scene will be used. This particular scene focuses mainly on how the codification strategy works to remove distortions originating from interreflections at various distances. Here, comparisons will also be made with GCPS using the same fringe width $W_F = 10\,\text{ppx}$.

8.2.1 Diffuse plane

The *Diffuse plane* scene will be rendered using CFPPS at distances $Z_C \in \{550\,\text{mm}, 800\,\text{mm}, 1000\,\text{mm}, 1400\,\text{mm}\}$. The distances 800 mm and 1000 mm will be used to observe how the codification strategy performs both in focus and slightly out of focus. The last distances will be used to analyze the worst-case performance at the distance extremities.

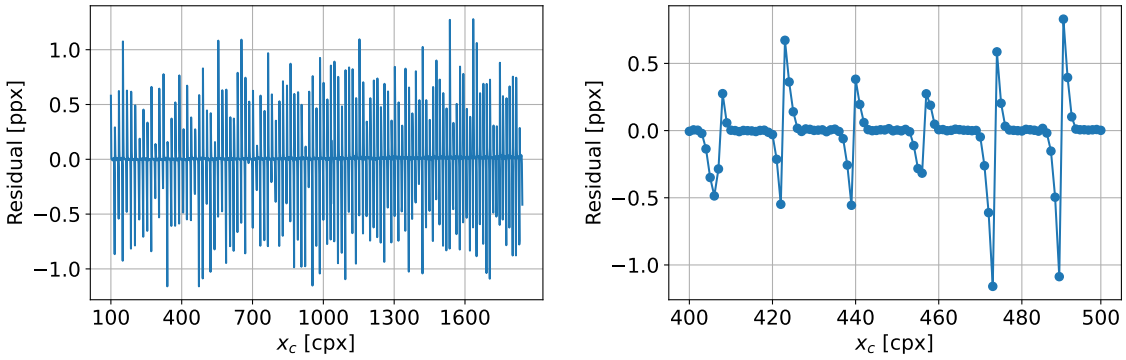
Multiple plots will be used to show the results. First, plots of the center row at $y_c = \lfloor \frac{Y_C}{2} \rfloor$ of the residual matrix will be included. This should be particularly useful for viewing the residuals that occur as a result of the fringe border distortions.

Next, a histogram of the residuals will be made. As it shows the numerical distribution of the residuals, it will reveal whether the distribution is unimodal, bimodal, etc. In addition, the symmetry of the residuals will be shown.

Lastly, an empirical CDF plot of the residuals will be included. The plot shows the cumulative distribution of the residuals and is useful for e.g. obtaining information on the subpixel accuracy. A table showing the CDF at certain limits will also be made.

The rendering is done using the simulator introduced in Section 4.1, and the results are shown below. They are grouped according to the camera distance.

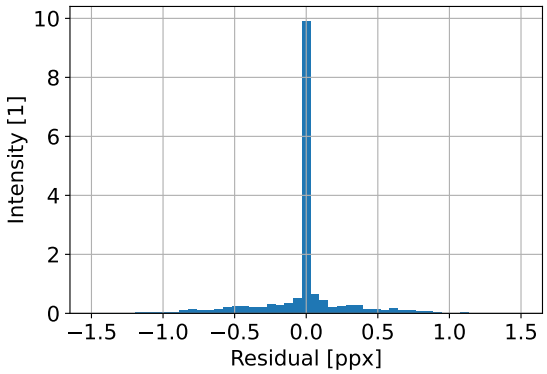
8.2.1.1 At distance $Z_C = 550\,\text{mm}$



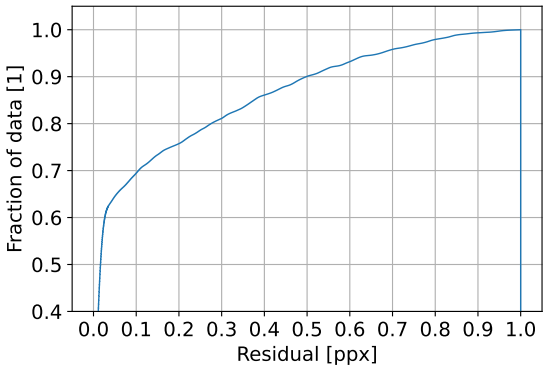
(a) $100\,\text{cpx} \leq x_c \leq 1794\,\text{cpx}$.

(b) $400\,\text{cpx} \leq x_c \leq 500\,\text{cpx}$.

Figure 47: Plots of the residuals for $y_c = \lfloor \frac{Y_C}{2} \rfloor$ cpx obtained using CFPPS on a diffuse plane at $Z_C = 550\,\text{mm}$.



(a) Histogram of the residuals.



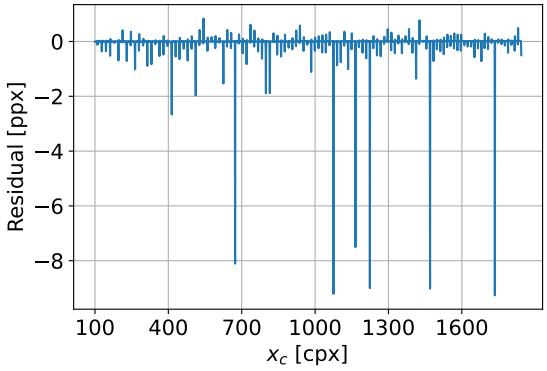
(b) Empirical CDF of the residuals.

Figure 48: Histogram and empirical CDF of the residuals obtained using CFPPS on a diffuse plane at $Z_C = 550\,\text{mm}$.

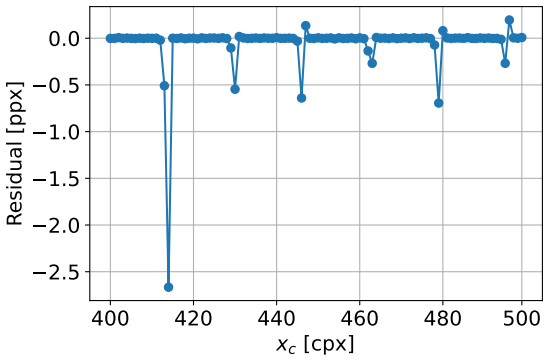
Residual limit	**Occurrence**
$< 0.05\,\text{ppx}$	63.8%
$< 0.1\,\text{ppx}$	68.7%
$< 0.2\,\text{ppx}$	74.9%
$< 0.5\,\text{ppx}$	89.0%
$< 0.75\,\text{ppx}$	95.7%
$< 1.0\,\text{ppx}$	98.9%

Table 2: Cumulative occurrence of the residuals obtained using CFPPS on a diffuse plane at $Z_C = 550\,\text{mm}$.

8.2.1.2 At distance $Z_C = 800\,\text{mm}$

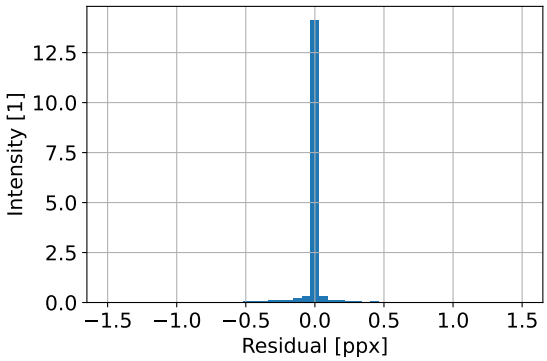


(a) $100\,\text{cpx} \leq x_c \leq 1794\,\text{cpx}$.

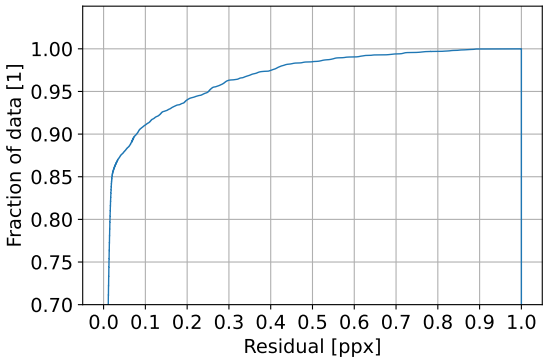


(b) $400\,\text{cpx} \leq x_c \leq 500\,\text{cpx}$.

Figure 49: Plots of the residuals for $y_c = \left\lfloor \frac{Y_C}{2} \right\rfloor\,\text{cpx}$ obtained using CFPPS on a diffuse plane at $Z_C = 800\,\text{mm}$.



(a) Histogram of the residuals.



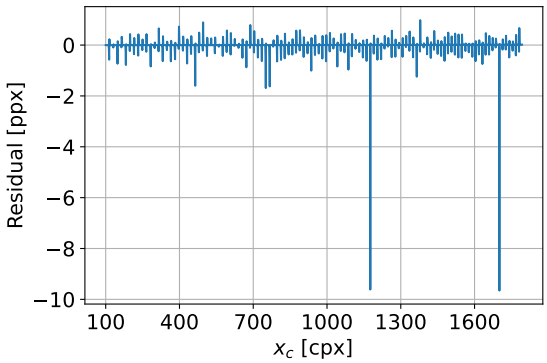
(b) Empirical CDF of the residuals.

Figure 50: Histogram and empirical CDF of the residuals obtained using CFPPS on a diffuse plane at $Z_C = 800\,\text{mm}$.

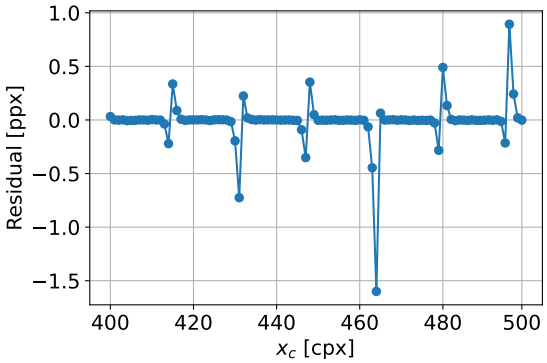
Residual limit	**Occurrence**
$< 0.05\,\text{ppx}$	87.2%
$< 0.1\,\text{ppx}$	90.3%
$< 0.2\,\text{ppx}$	93.2%
$< 0.5\,\text{ppx}$	97.6%
$< 0.75\,\text{ppx}$	98.7%
$< 1.0\,\text{ppx}$	99.1%

Table 3: Cumulative occurrence of the residuals obtained using CFPPS on a diffuse plane at $Z_C = 800\,\text{mm}$.

8.2.1.3 At distance $Z_C = 1000\,\text{mm}$

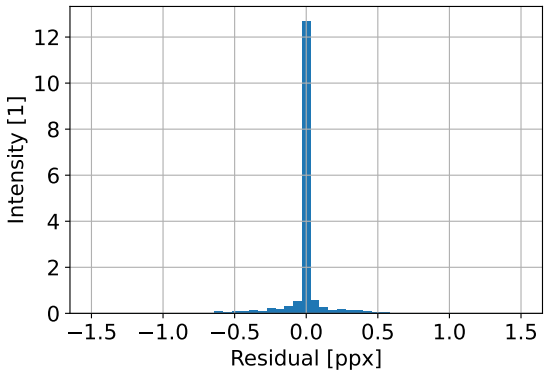


(a) $100\,\text{cpx} \leq x_c \leq 1794\,\text{cpx}$.

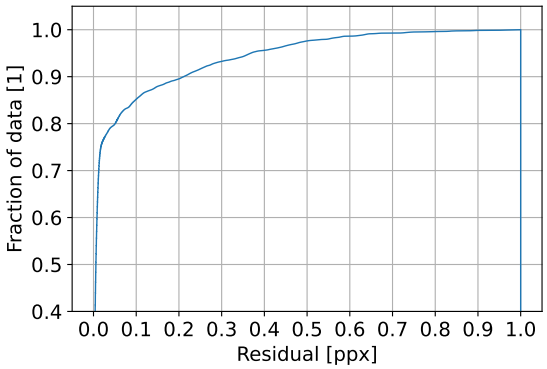


(b) $400\,\text{cpx} \leq x_c \leq 500\,\text{cpx}$.

Figure 51: Plots of the residuals for $y_c = \left\lfloor \frac{Y_C}{2} \right\rfloor$ cpx obtained using CFPPS on a diffuse plane at $Z_C = 1000\,\text{mm}$.



(a) Histogram of the residuals.



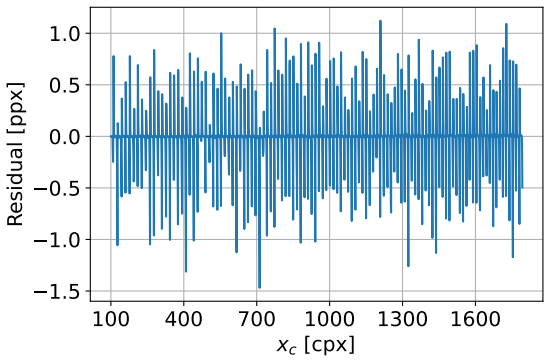
(b) Empirical CDF of the residuals.

Figure 52: Histogram and empirical CDF of the residuals obtained using CFPPS on a diffuse plane at $Z_C = 1000\,\text{mm}$.

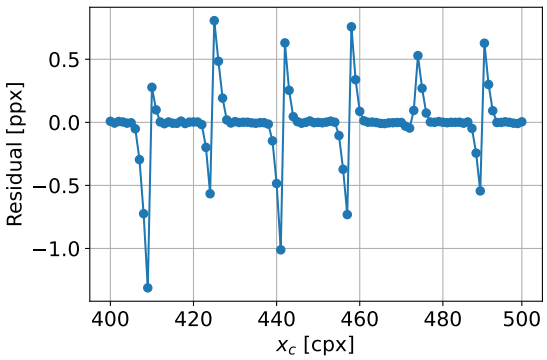
Residual limit	**Occurrence**
$< 0.05\,\text{ppx}$	79.6%
$< 0.1\,\text{ppx}$	84.9%
$< 0.2\,\text{ppx}$	89.2%
$< 0.5\,\text{ppx}$	97.2%
$< 0.75\,\text{ppx}$	99.2%
$< 1.0\,\text{ppx}$	99.6%

Table 4: Cumulative occurrence of the residuals obtained using CFPPS on a diffuse plane at $Z_C = 1000\,\text{mm}$.

8.2.1.4 At distance $Z_C = 1400\,\text{mm}$

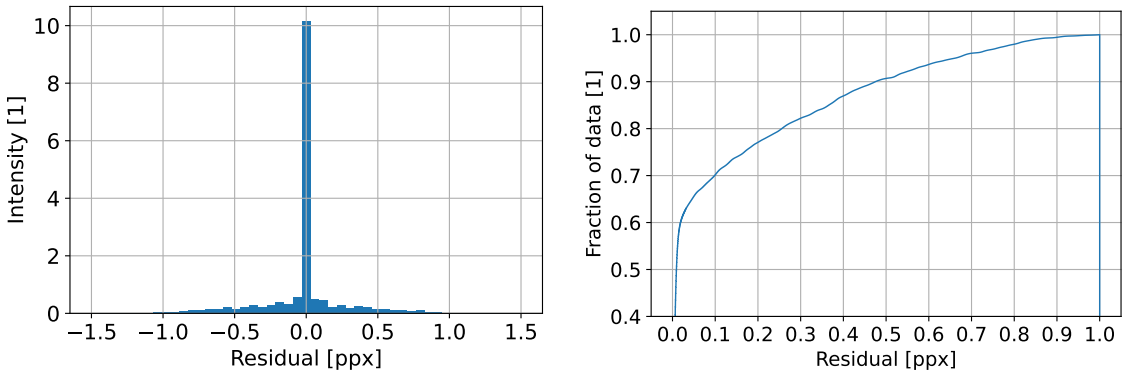


(a) $100\,\text{cpx} \leq x_c \leq 1794\,\text{cpx}$.



(b) $400\,\text{cpx} \leq x_c \leq 500\,\text{cpx}$.

Figure 53: Plots of the residuals for $y_c = \left\lfloor \frac{Y_C}{2} \right\rfloor\,\text{cpx}$ obtained using CFPPS on a diffuse plane at $Z_C = 1400\,\text{mm}$.



(a) Histogram of the residuals.

(b) Empirical CDF of the residuals.

Figure 54: Histogram and empirical CDF of the residuals obtained using CFPPS on a diffuse plane at $Z_C = 1400\,\text{mm}$.

Residual limit	**Occurrence**
$< 0.05\,\text{ppx}$	65.0%
$< 0.1\,\text{ppx}$	69.5%
$< 0.2\,\text{ppx}$	76.3%
$< 0.5\,\text{ppx}$	89.9%
$< 0.75\,\text{ppx}$	96.0%
$< 1.0\,\text{ppx}$	99.1%

Table 5: Cumulative occurrence of the residuals obtained using CFPPS on a diffuse plane at $Z_C = 1400\,\text{mm}$.

8.2.2 Objects in bin

The scene *Objects in bin* is used primarily to compare CFPPS to GCPS when there are many interreflections, for a range of distances. The renders will be performed for $Z_C \in \{550\,\text{mm}, 800\,\text{mm}, 1400\,\text{mm}\}$ to cover the whole distance range, and $Z_C^L = 550\,\text{mm}$ and $Z_C^U = 1500\,\text{mm}$ has been used for all camera distances. While the results from the *Diffuse plane* scene focused primarily on the numerical distribution of residuals and the fringe border residuals, the results from this scene will be more qualitative.

Residual matrix plots will be made for both CFPPS and GCPS for all distances using the material **metal-80**. These will be used to qualitatively compare the spatial distribution of residuals. Multiple colored arrows will be placed on these plots at various locations to aid in the discussion of the results.

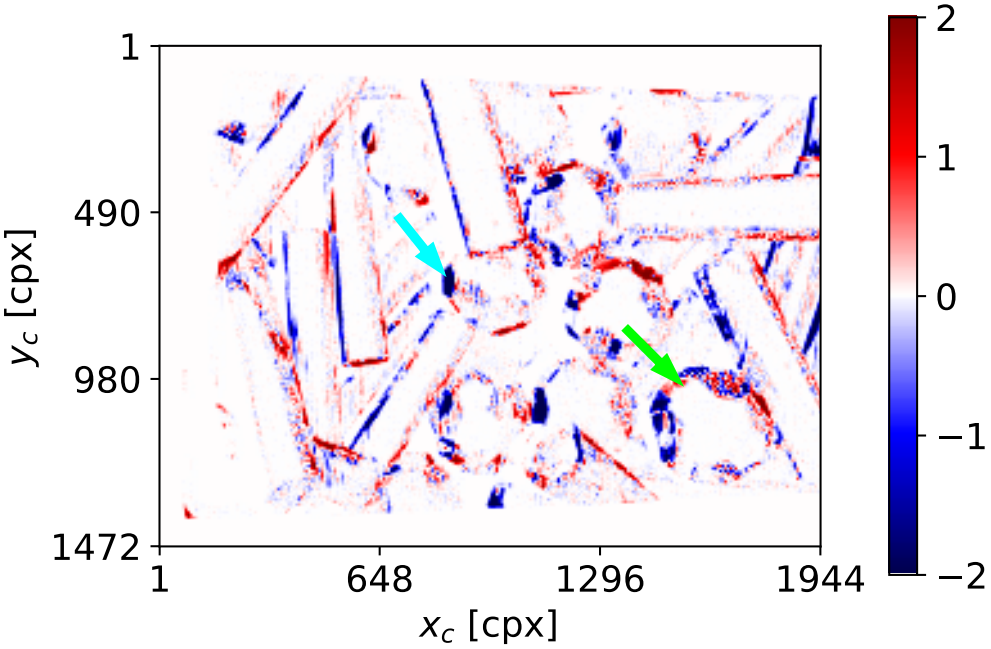
The second type of metric is the empirical CDF distribution plot. In these plots, the CDF of the residuals compared to the ground truth for both CFPPS and GCPS are visualized, allowing a direct comparison between these two. In addition, the empirical CDF of CFPPS with direct reflections will be included. This serves as a comparison of how much the accuracy is reduced by the interreflections. Here, several acronyms will be used as labels. The suffix "-D" in "CFPPS-D" indicates that

the plot was made considering direct reflections only for CFPPS. Correspondingly, "-C" is used for the *combined* signal, consisting of both the interreflections and the direct reflections. Lastly, the "-C(F)" suffix means that the combined signal is used with a fringe border filter, as explained in the following.

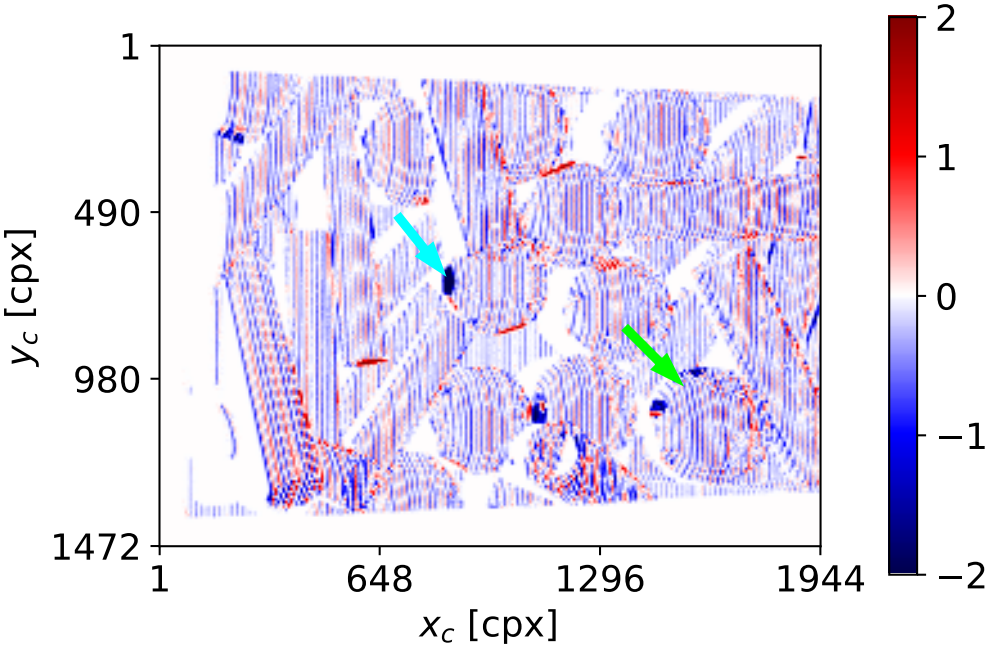
For the distance $Z_C = 800\,\text{mm}$, some additional renderings have been done. In particular, residual matrices using the material **metal-50** have been made and included to observe how the amount of specularity affects the spatial distribution of residuals. Moreover, residuals have been calculated for the CFPPS with interreflections by also applying a fringe border filter as defined in Section 7.2.5. The filter has been configured with a threshold of $T_F = 1.25\,\text{ppx}$ and is used with the material **metal-80**. Notice that with this threshold, approximately $\frac{2 \cdot T_F}{W_F} = 25\%$ of the camera pixels are removed from the render when the filter is applied.

The rendering is done using the simulator introduced in Section 4.1, and the results are shown below. They are grouped according to the camera distance Z_C.

8.2.2.1 At distance $Z_C = 550\,\text{mm}$



(a) GCPS



(b) CFPPS

Figure 55: Residual matrices obtained for the *Objects in bin* scene at $Z_C = 550\,\text{mm}$ with the material **metal-80**.

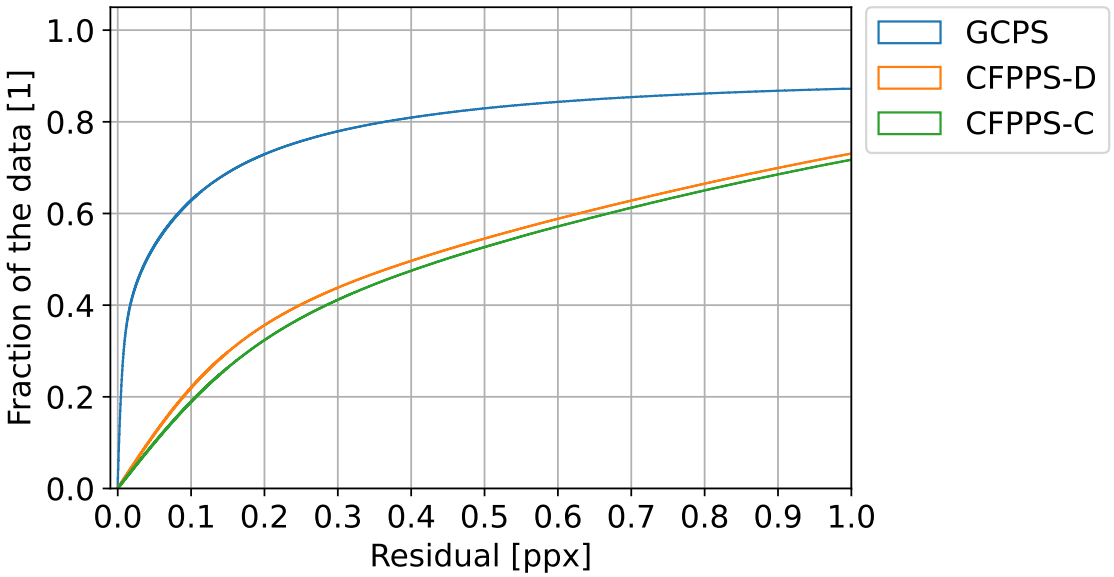


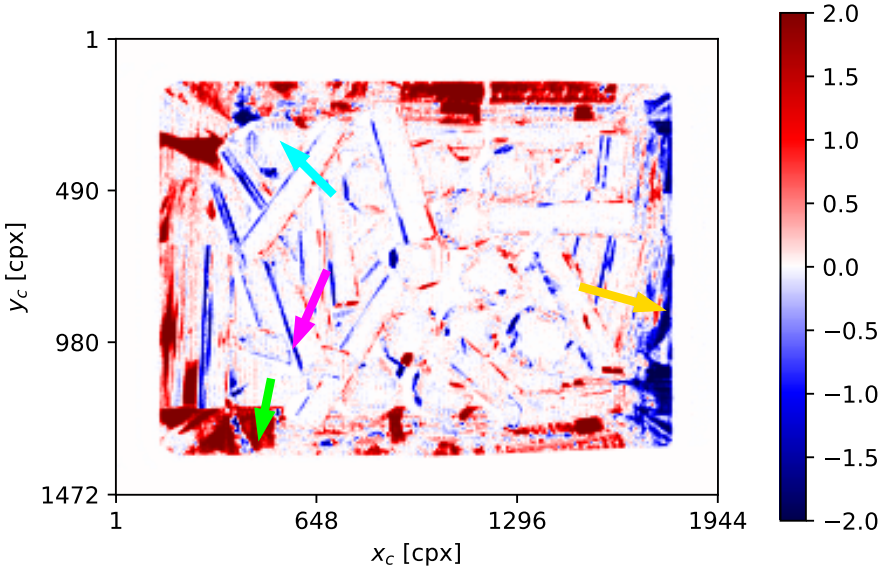
Figure 56: Empirical CDF obtained for the *Objects in bin* scene at $Z_C = 550\,\text{mm}$ with the material **metal-80**.

Residual limit	**GCPS**	**CFPPS-C**	**CFPPS-D**
$< 1.0\,\text{ppx}$	87.3%	71.7%	73.1%
$< 5.0\,\text{ppx}$	90.4%	92.1%	92.7%
$< 10.0\,\text{ppx}$	94.1%	99.1%	99.8%
$< 20.0\,\text{ppx}$	96.7%	99.5%	99.9%

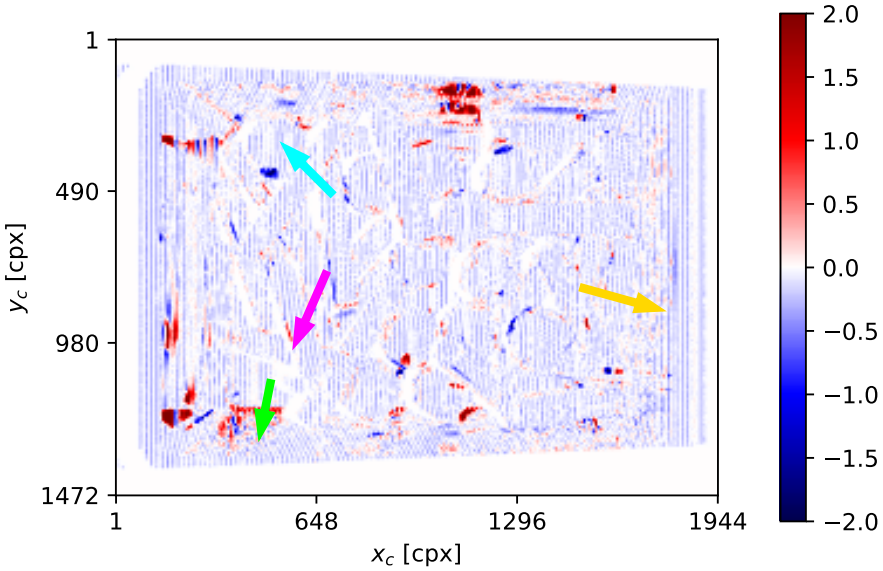
Table 6: Cumulative occurrence of the residuals obtained using CFPPS on the *Objects in bin* scene at $Z_C = 550\,\text{mm}$ with the material *metal-80*.

Fringe decoding errors occurred at a rate of 1.27% for CFPPS and 8.36% for GCPS at distance $Z_C = 550\,\text{mm}$ when using the material **metal-80**.

8.2.2.2 At distance $Z_C = 800\,\text{mm}$

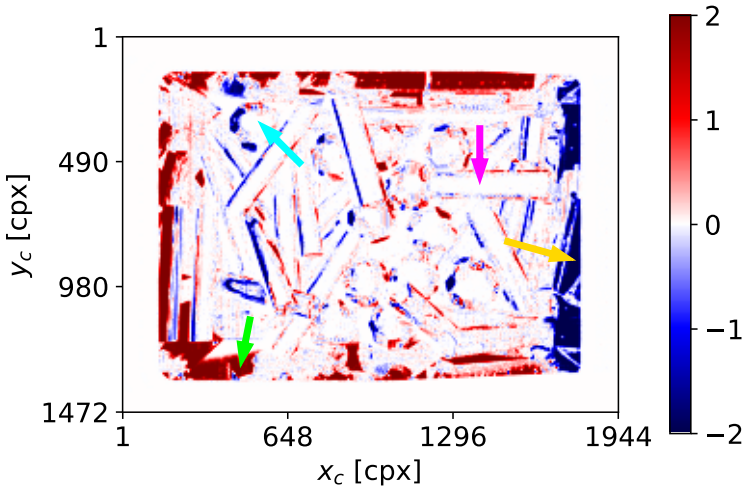


(a) GCPS

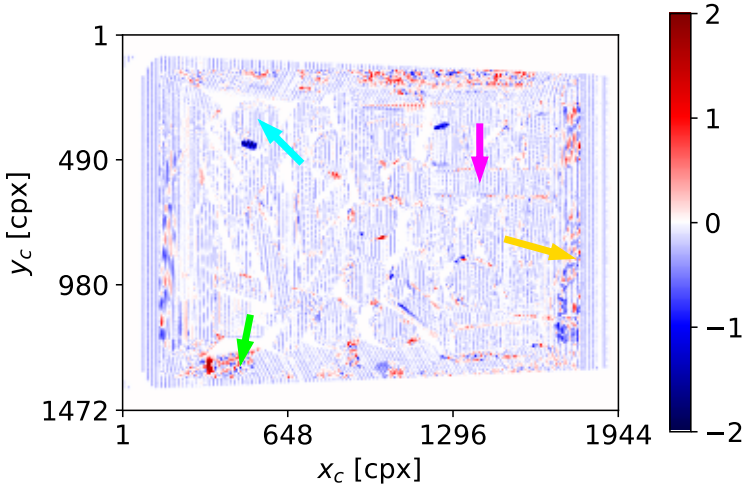


(b) CFPPS

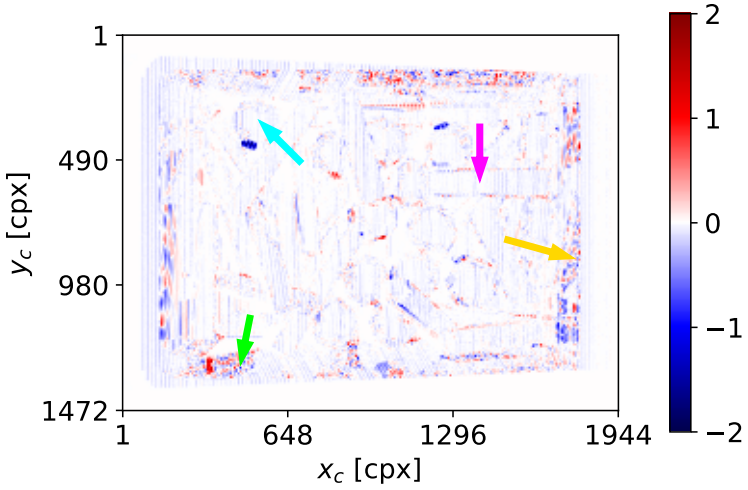
Figure 57: Residual matrices obtained for the *Objects in bin* scene at $Z_C = 800\,\text{mm}$ with the material **metal-50**.



(a) GCPS



(b) CFPPS



(c) CFPPS w/ fringe border filter

Figure 58: Residual matrices obtained for the *Objects in bin* scene at $Z_C = 800\,\text{mm}$ with the material **metal-80**.

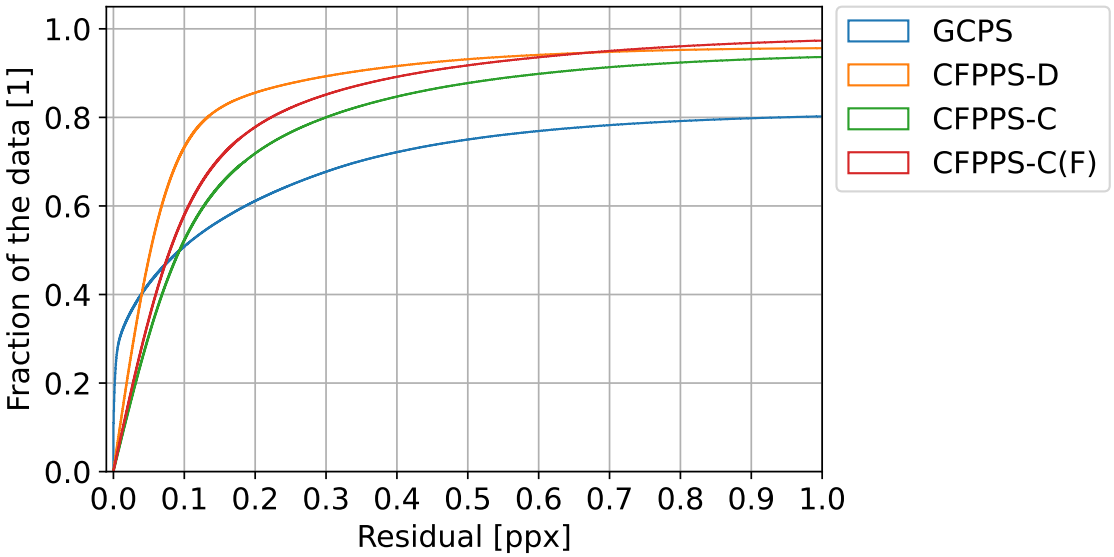


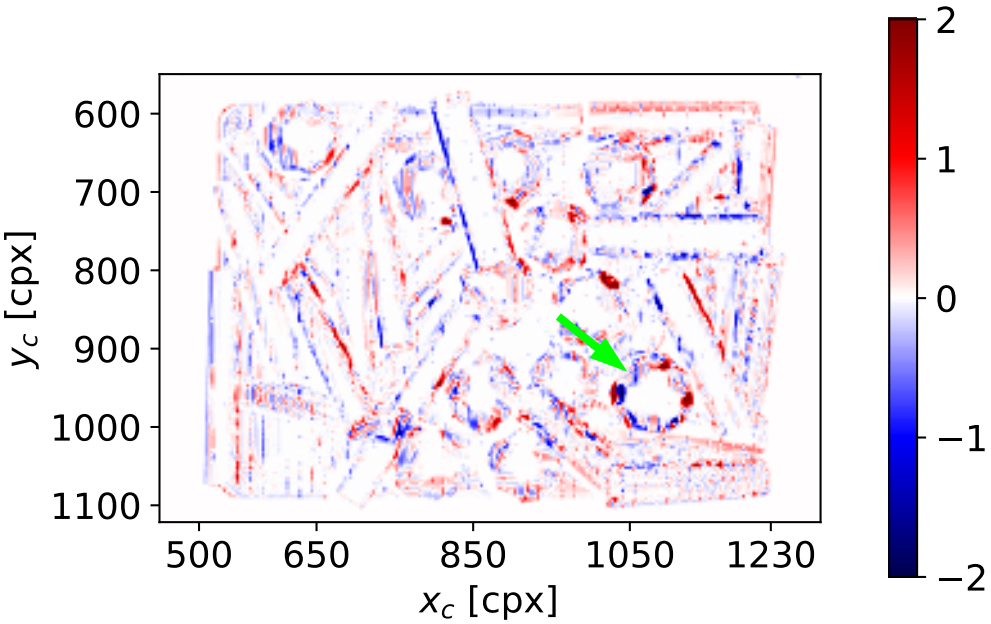
Figure 59: Empirical CDFs obtained for the *Objects in bin* scene at $Z_C = 800\,\text{mm}$ with the material **metal-80**.

Residual limit	**GCPS**	**CFPPS-C**	**CFPPS-D**	**CFPPS-C(F)**
$< 1.0\,\text{ppx}$	80.2%	93.6%	95.6%	97.6%
$< 5.0\,\text{ppx}$	81.7%	95.5%	95.8%	99.4%
$< 10.0\,\text{ppx}$	84.8%	99.7%	99.9%	99.7%
$< 20.0\,\text{ppx}$	88.1%	99.8%	99.9%	99.9%

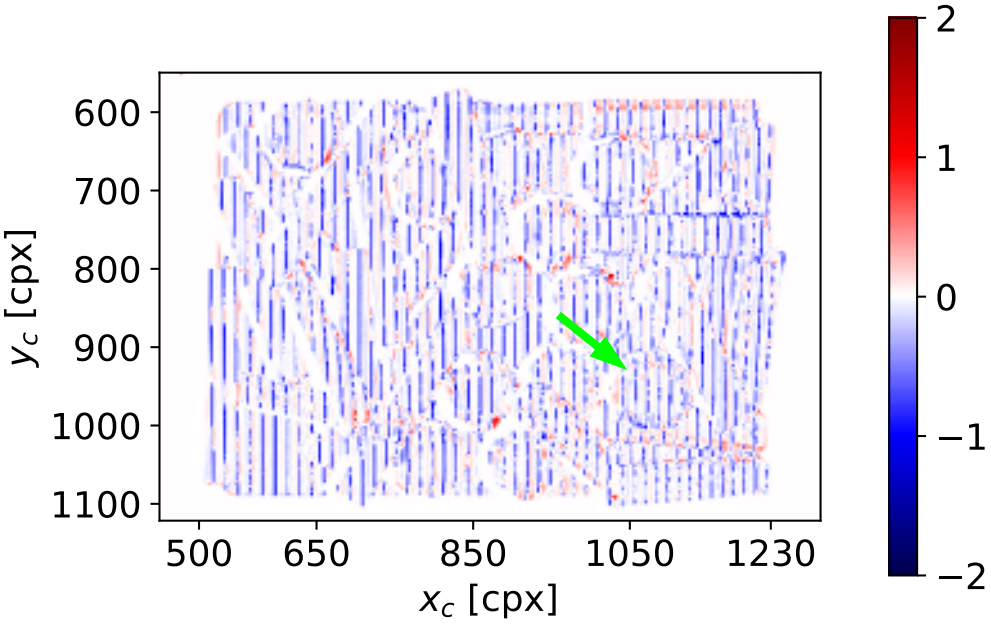
Table 7: Cumulative occurrence of the residuals obtained using CFPPS on the *Objects in bin* scene at $Z_C = 800\,\text{mm}$ with the material **metal-80**.

Fringe decoding errors occurred at a rate of 0.48% for CFPPS and 17.3% for GCPS at distance $Z_C = 800\,\text{mm}$ when using the material **metal-80**.

8.2.2.3 At distance $Z_C = 1400\,\text{mm}$



(a) GCPS



(b) CFPPS

Figure 60: Residual matrices obtained for the *Objects in bin* scene at $Z_C = 1400\,\text{mm}$ with the material **metal-80**.

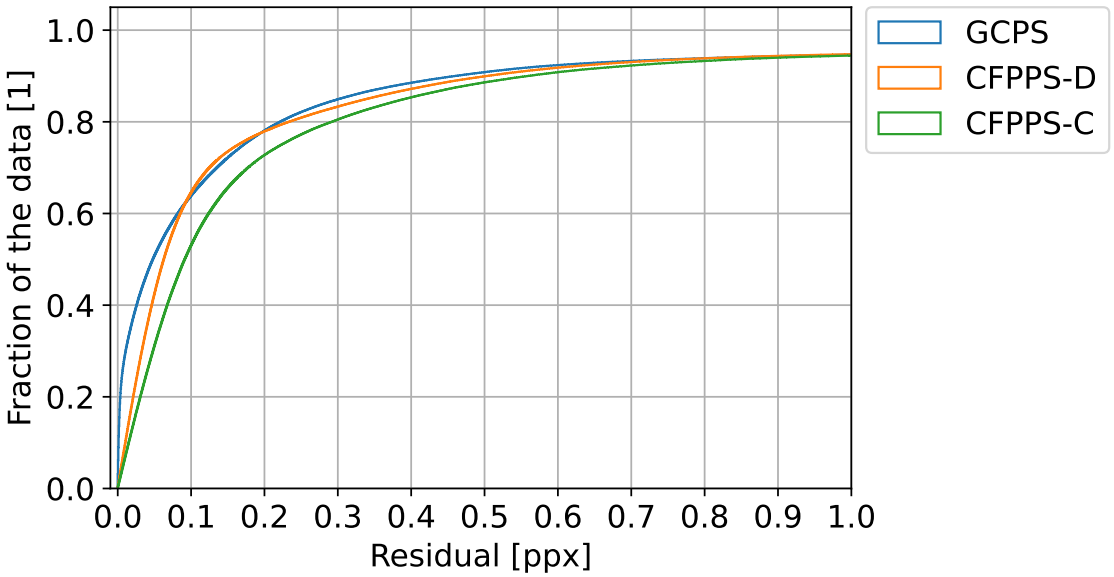


Figure 61: Empirical CDF obtained for the *Objects in bin* scene at $Z_C = 1400\,\text{mm}$ with the material **metal-80**.

Residual limit	**GCPS**	**CFPPS-C**	**CFPPS-D**
$< 1.0\,\text{ppx}$	94.6%	94.5%	94.7%
$< 5.0\,\text{ppx}$	95.6%	95.5%	95.5%
$< 10.0\,\text{ppx}$	98.0%	99.9%	99.9%
$< 20.0\,\text{ppx}$	99.5%	99.9%	99.9%

Table 8: Cumulative occurrence of the residuals obtained using CFPPS on the *Objects in bin* scene at $Z_C = 1400\,\text{mm}$ with the material **metal-80**.

Fringe decoding errors occurred at a rate of 0.07% for CFPPS and 3.40% for GCPS at distance $Z_C = 1400\,\text{mm}$ when using the material **metal-80**.

9 Discussion

The starting point of this thesis was some correlation-based patterns that worked well when the system was in focus, but failed outside. Through system identification, a distance response was obtained and plotted in Figure 29. It correlated well with the energy plot of correlation-identified fringes in Figure 34, leading to the suggestion that an increase in fringe width W_F should make the patterns more resilient to lens defocus. The scene *Objects in bin* has been rendered using CFPPS, which includes CIF patterns with $W_F = 10\,\text{ppx}$ for fringe identification. For any distance rendered, fringe decoding errors have been observed to occur at a rate of no more than 1.27% using this codification strategy. In the project thesis (Lima-Eriksen 2022), the *Grouped Gold* pattern codification strategy was rendered at $Z_C = 1000\,\text{mm}$ using the scene *Metal cylinders in picking bin*. No numerical metric of the error rate was made for that distance, only the decoded rendering. However, a visual inspection of that decoded rendering makes it apparent that the pattern codification strategy used in this thesis attains a significantly lower error rate. Therefore, it is reasonable to conclude that the increase in fringe width and thus the decrease in spatial frequency content makes the correlation-based patterns more resistant to defocus distortions.

Notice also from the renderings of *Objects in bin* that GCPS achieved a fringe error rate of 17.3% at $Z_C = 800\,\text{mm}$ with **metal-80**, while the same rate was only 0.43% for CFPPS. First, these numbers show that the decoding errors for the gray codes in GCPS can actually be significant for some scenes; more than one in seven camera pixels has incorrectly decoded gray codes in GCPS in this particular case. More importantly, the large difference in this rate between the codification strategies indicates that correlation-based patterns can be really useful in reducing such types of errors. With a fringe decoding error rate of no more than 1.27% for all distances, it appears that the correlation-based patterns handles interreflections really well. Also, observe that the residual plots using **metal-50** in Figure 57b depicts this same trend of decreasing large residuals, suggesting that the same results may be found for objects with lower specularity as well.

The residual matrices for the *Objects in bin* scene with **metal-80** at $Z_C = 800\,\text{mm}$ are shown in Figure 58. Here, it is apparent that large areas are erroneously decoded when using GCPS. Notice in particular the large horizonal reflection in the rightmost wall shown by the yellow arrow. This is almost completely removed when using CFPPS. Also, it is seen that most objects have decoding errors around their edges, such as the one indicated by the cyan-colored arrow. Recall that 3D cameras are particularly useful for pick-and-place operations. By distorting these edges, it might be difficult for an object recognition software to find the exact position of these objects, as it cannot accurately capture their spatial extent. These boundary errors are mostly removed in CFPPS. However, there are some larger areas of residuals still present using CFPPS, in particular the blue area in the object with the cyan arrow. The residual is located in the bottom half of the object, which could suggest that a vertical reflection causes an intra-fringe interreflection here. As shown previously, these errors cannot be completely avoided due to the working mechanisms of the patterns. Nevertheless, most of the vertical reflections are actually significantly

reduced. For instance, observe the vertical reflection occurring at the lower left corner of the bin indicated by the green arrow. This is a vertical reflection, but is significantly reduced when using CFPPS. Perhaps it could be due to the fringe width not being high enough to cause intra-fringe interreflections here.

Object-border residuals are also present for GCPS at distances $Z_C \in \{550\,\text{mm}, 1400\,\text{mm}\}$, as seen in the residual matrices in Figure 55 and Figure 60, respectively. However, there is a lower incidence of them. Notice from the occlusion exclusion masks listed in Appendix A that the bin walls are illuminated only for $Z_C = 800\,\text{mm}$. It could be the case that the walls are the origin of most of the interreflections and, therefore, resulting in a reduced incidence for the other distances. Nevertheless, the residual matrices reveal that the object-border residuals which are present are reduced significantly for these objects as well, indicated by e.g. the objects with green arrows.

Although it is apparent that many of the interreflections are reduced or removed using CFPPS, the codification strategy still has its shortcomings. From the residual matrices using the *Objects in bin* scene, it is quite apparent that residuals are present at regular intervals as vertical lines. This phenomenon is also present when rendering CFPPS on the scene *Diffuse plane* – a scene which does not contain any interreflection. Therefore, it cannot be attributed to interreflections, and must instead be a phenomenon that occurs due to the nature of the patterns themselves. Recall from the limitations mentioned in Section 7.2.3 that the correlation-identified fringes patterns should cause periodic distortions at the fringe borders because of the lens defocus. This seems to be the phenomenon causing these distortions, as the nature of the distortions matches well with e.g. the periodicity. Consider, for instance, the residuals of this scene with $Z_C = 800\,\text{mm}$ as seen in Figure 49b. Here, roughly 4 to 5 camera pixels are erroneously decoded at the fringe border within each fringe. Considering that a fringe is $S_X \cdot W_F = 17\,\text{cpx}$ wide, up to $\frac{5}{17} \approx 30\%$ of the camera pixels are subject to this type of distortion, which is significant. Nevertheless, Section 7.2.4 predicted approximately 24% of the pixels will be affected by fringe-border distortions, and so it is within what was expected. Taking into account the residual plots Figure 47b for $Z_C = 550\,\text{mm}$ and Figure 53b for $Z_C = 1400\,\text{mm}$, this distortion only increases in magnitude when out of focus, also as expected. A consequence of these fringe border errors is that objects such as the cylinder in Figure 58 indicated by the pink arrow becomes completely covered by periodically occurring residuals. This is in stark contrast to the almost completely correctly decoded cylinder depicted in its residual matrix using GCPS. Therefore, it is obvious that CFPPS is not free of distortions.

There is however one major difference in the *nature* of the distortions for these two pattern codification strategies. The distortions from interreflections seen in GCPS are scene-dependent and are therefore random in nature. On the other hand, the fringe border residuals seen in CFPPS occur at regular intervals that correspond to the fringe width W_F, indicating that it is a systematic type of error. Moreover, notice from the empirical CDF plots for the *Objects in bin* scene that the vast majority of residuals are less than 0.5 ppx for CFPPS. This means that the camera pixels that decode projector columns that are close to a fringe border are very susceptible to fringe border distortions, as suggested in Section 7.2.5 through the concept of a

fringe border filter. By applying the fringe border filter with $T_F = 1.25\,\text{ppx}$, the residual matrix shown in Figure 58c is obtained when the material **metal-80** is used at distance $Z_C = 800\,\text{mm}$. It is quite apparent from this plot that a large fraction of the fringe border residuals are removed. Since such systematic errors can be predicted, they are, to some extent, possible to reduce. Therefore they should be less of a concern compared to the random interreflection errors seen in GCPS. However, it should be noted that this method of removing camera pixels reduces the amount of camera pixels by $\frac{2 \cdot T_F}{W_F} = 25\%$ and is probably not the best way to remove systematic errors. It has been included merely to state that it is possible to predict the errors to some extent.

Recall from the problem description that most applications of the Zivid Two camera require residuals less than 0.2 ppx. Empirical CDF graphs for the scene *Objects in bin* show the numerical comparison of residuals between GCPS and CFPPS. Due to the fringe decoding errors for $Z_C = 800\,\text{mm}$ using GCPS, it is evident that CFPPS outperforms GCPS, reaching a residual rate of $> 70\%$ being less than 0.2 ppx, compared to GCPS with a rate of $\approx 60\%$. This rate is more similar for $Z_C = 1400\,\text{mm}$, with a rate of $\approx 70\%$ for both here. The similarity could probably be due to the fact that not many interreflections are present at this distance, and that the interreflection error rate of GCPS becomes similar to the fringe border error rate of CFPPS. For all these distances, CFPPS has a high rate of residuals less than 0.2 ppx. Depending on the requirements for the use-case, it could very well be that this rate is good enough to be used in applications. Either way, it reduces most of the interreflections while not significantly affecting the accuracy at these distances.

Lastly, the situation is quite the opposite for $Z_C = 550\,\text{mm}$. Here, CFPPS reaches a rate of merely 30% of the residual being less than 0.2 ppx, compared to GCPS with a rate of $> 70\%$. It seems that the fringe-border distortions dominate the residuals, and so it is quite a problem for this distance. It is unknown why these distortions are so significant at this distance; perhaps they are due to an underestimate of σ_{PSF} at this distance. Probably a higher value of W_F could decrease these distortions.

The major problem with CFPPS seems to be the fringe border distortions, and so it is interesting to look at how much this type of distortions actually contributes towards the residuals of CFPPS. Such insights can be found when comparing the residuals of the combined reflections (CFPPS-C) to the ones using direct reflections only (CFPPS-D) with this codification strategy. The empirical CDFs of these residuals are shown in the same plots for the *Objects in bin* scene, allowing for the comparison between them. Since the residuals of the direct signal are not subject to interreflections, it will represent the best possible results for the particular scene. Consider the empirical CDFs for $Z_C = 800\,\text{mm}$ with **metal-80** as seen in Figure 59. Here, the orange curve represents the residuals without interreflections, and the green curve denotes the residuals when interreflections are present. The difference between these two curves should thus correspond to the distortions caused by the interreflections alone. Notice that for all these empirical CDF plots, the differences between these curves are quite significant and accounts for 50% of the residuals. Therefore, it is expected that the reduction of fringe border distortions could substantially improve the performance of CFPPS. Figure 59 also visualizes the residuals after applying the fringe border filter as a red curve. The residuals are lower with

the filter than without, further strengthening the hypothesis that a large fraction of the residuals occur at the fringe borders.

It is important to bear in mind that the fringe border errors occur due to the optical properties of the system, in particular the amount of lens blurring. The Zivid Two camera used for the evaluation in this thesis was specifically developed to work well with pattern codification strategies such as GCPS. Therefore, even better performance can be achieved by adjusting the optical properties to fit CFPPS better. In particular, different lenses can be used to reduce the amount of blur that occurs at extreme camera distances. Also, it can be shown that increasing the baseline, i.e. having a larger displacement between the camera and the projector, would mean that the system tolerates a larger projector column error while still being within the 0.1% error margin (Bouquet et al. 2017). None of these parameters have been explored further. Nevertheless, the pattern codification strategy shows promising results with regard to performance in challenging environments with many shiny objects. The observed errors are systematic and could possibly be reduced or avoided by further development.

However, little attention has been paid to *efficiency* of the two pattern codification strategies. With 31 correlation-based patterns and 20 cosine patterns, this codification strategy uses 4.1 times the number of patterns compared to the 11 patterns required by GCPS, when taking into account the fully black and white patterns required for normalization. Recall that the temporal pattern codification strategies requires the scene to be static under acquisition. This means that if a robot was to be used in conjunction with the system, it would have to stay still 4.1 times longer for each acquisition compared to when using CFPPS. Robot operations should typically be done as quickly as possible, and so this could, in fact, be so limiting that the codification strategy simply cannot be used.

The codification strategy is also much more computationally demanding when it comes to decoding the result. For GCPS, the codification strategy requires first normalization of the images and then a lookup to find the corresponding binary representation of each of the normalized binary codes. Lastly, eight multiplications, six summations, and arctan are required for the cosine samples, referring to (10). For CFPPS, it is significantly more complex. Here, correlation should be performed between CIF captures and the valid codes. Also, the permuted phase shifts captures should be premultiplied by the correct permutation matrices. Referring to (46), phase estimates of these non-permuted phase shifts require 38 additions and 40 multiplications, in addition to the arctan. While an exact performance comparison cannot be made on the basis of this, it is apparent that CFPPS is significantly more complex computation-wise. This could further limit the use cases.

10 Conclusion

In this thesis, a novel temporal pattern codification strategy for structured light systems has been developed. Its main goal was to work well in the presence of shiny objects over a wide range of distances. The codification strategy consists of two types of patterns. The first type uses correlation to correctly identify the originating projector column sector (fringe) in each camera pixel. The second type uses temporally reordered phase shifts to estimate the originating projector column relative to the fringe in each camera pixel with sub-pixel accuracy. Together, these patterns make it possible to use structured light in challenging environments with many shiny objects.

The codification strategy was first tested on a diffuse plane at various distances, and it revealed that decoding errors occurred periodically with the fringe width even when no interreflections were present. This was attributed to the fact that lens defocus causes noise at the fringe borders. Lastly, the codification strategy was tested in a scene with many shiny objects. Compared to GCPS, it performed significantly better when many interreflections occurred. The fringe decoding errors seen in GCPS were almost eliminated in CFPPS, reaching a fringe decoding error rate of less than 1.27% for $Z_C \in [550\,\text{mm}, 1400\,\text{mm}]$. Furthermore, the errors that are seen on the edges of objects were reduced when using CFPPS. For distances $Z_C \in [800\,\text{mm}, 1400\,\text{mm}]$, the novel pattern codification strategy had more than 70% of its residuals within the $0.2\,\text{ppx}\,(0.1\%)$ target. The codification strategy struggled at $Z_C = 550\,\text{mm}$, which could probably be due to a too low fringe width at this distance.

The main source of residuals in CFPPS was the fringe-border distortions, which were found to be systematic errors. Therefore, it is possible that these errors can be reduced by exploiting their periodic and predictable occurrence. The errors are largely caused by optical properties such as lens blurring. The Zivid Two camera has not been optimized for this pattern codification strategy, which means that better results can be achieved by developing a new camera with more suitable optical properties.

Although the codification strategy shows promising results, it is still much more demanding in terms of acquisition time and computational complexity. The acquisition time was found to be 4.1 times higher compared to GCPS. For many use-cases, this increase in complexity could mean that the codification strategy is not of any use despite its better decoding accuracy. If so, other codification strategies should be considered instead.

11 Future Work

The pattern codification strategy developed in this thesis showed promising initial results with respect to the suppression of interreflections at a wide range of distances, suggesting that it could have real-world applications. Nevertheless, it has some systematic errors that occur at regular intervals due to lens defocus, known as fringe-border errors. Future work that addresses this issue could make the codification strategy significantly better and, in turn, make it more useful. There are mainly two ways in which these fringe-border errors can be addressed, as stated in the discussion.

The first way to reduce fringe-border interreflections is to design a new structured light system with lenses causing less defocus within the desired range of distances. This would directly reduce the standard deviation of the point-spread function, which in turn reduces the residuals at the fringe borders. Also, the baseline of the structured light system can be increased to make it tolerant to higher projector column residuals while still being within the 0.1% error margin (Bouquet et al. 2017).

The second way to reduce the fringe-border distortions is through addressing them from a signal processing perspective. The distortions are systematic errors, meaning that it is predictable where the errors will occur. Perhaps, some additional patterns that have these errors at different locations can be added. An algorithm that chooses between which patterns to decode could then possible eliminate these errors. Filters that either removes or corrects the camera pixels at the fringe borders could also yield promising results.

Bibliography

Association for Advancing Automation (2017). *Industrial Robots: 3D Vision a Driving Force of Innovation*. URL: https://www.automate.org/blogs/industrial-robots-3d-vision-a-driving-force-of-innovation (visited on 13th Feb. 2022).

— (2020). *Logistics Robots*. URL: https://www.automate.org/a3-content/service-robots-logistic-robots (visited on 13th Feb. 2022).

Barker, R. H. (1953). 'Group Synchronizing of Binary Digital Systems'. In: *Communication Theory*, pp. 273–287.

Borgan, Øyvind (2022). *Zivid Two brings human-like vision to pick-and-place robotics*. URL: https://blog.zivid.com/zivid-two-human-like-vision-for-pick-and-place-robotics (visited on 8th May 2022).

Bouquet, Gregory et al. (Oct. 2017). 'Design tool for TOF and SL based 3D cameras'. In: *Opt. Express* 25.22, pp. 27758–27769. DOI: 10.1364/OE.25.027758. URL: http://opg.optica.org/oe/abstract.cfm?URI=oe-25-22-27758.

Conrad, Jeff (July 2018). 'Depth of Field in Depth'. In.

Deeb, Rada et al. (2017). 'Interreflections in Computer Vision'. In: *Journal of Mathematical Imaging and Vision*.

Engelman, Ryan (2022). *The Second Industrial Revolution, 1870-1914*. URL: https://ushistoryscene.com/article/second-industrial-revolution/ (visited on 5th Feb. 2022).

Fabry, Thomas, Dirk Smeets and Dirk Vandermeulen (Apr. 2010). *Surface representations for 3D face recognition*. ISBN: 978-953-307-060-5. DOI: 10.5772/8951.

Fiete, Robert D. (2010). *Modeling the Imaging Chain of Digital Cameras*. SPIE. ISBN: 978-0819483393.

Gold, Robert (1967). 'Optimal binary sequences for spread spectrum multiplexing'. In: *IEEE Transactions on Information Theory*, pp. 619–621. DOI: 10.1109/TIT.1967.1054048.

Golnazarian, Wanek and Ernest L. Hall (2000). 'Intelligent Industrial Robots'. In: *Handbook of Industrial Automation*.

Graat, Martijn (2020). *270,000 New Robots Working in Logistics This Year*. URL: https://logisticsmatter.com/270000-new-robots-working-in-logistics-this-year/ (visited on 12th Feb. 2022).

Gray, Frank (1947). 2,632,058. URL: https://patentimages.storage.googleapis.com/a3/d7/f2/0343f5f2c0cf50/US2632058.pdf.

Harding, Kevin (2019). 'Methods for addressing multiple reflections in a structured light profiler'. In: *SPIE Defence*.

Hartley, Richard and Andrew Zisserman (2003). *Multiple view geometry in computer vision*. Cambridge university press.

He, Kejing et al. (2020). '3D reconstruction of objects with occlusion and surface reflection using a dual monocular structured light system'. In: *Applied Optics*.

Hung, Y. Y. (2000). 'Practical three-dimensional computer vision techniques for full-field surface measurement'. In: *Optical Engineering* 39 (1), pp. 143–149.

International Federation of Robotics (2019). *World Robotics Report 2020 by International Federation of Robots*. URL: https://ec.europa.eu/newsroom/rtd/items/700621 (visited on 15th July 2022).

International Federation of Robotics (2021). *US Robot Density in Car Industry Ranks 7th Worldwide*. URL: https://ifr.org/ifr-press-releases/news/us-robot-density-in-car-industry-ranks-7th-worldwide (visited on 12th Feb. 2022).

Jin, Xiaodan and Keigo Hirakawa (Feb. 2013). 'Approximations To Camera Sensor Noise'. In: *Proceedings of SPIE - The International Society for Optical Engineering* 8655. DOI: 10.1117/12.2019212.

Kawasaki, Hiroshi et al. (2009). 'Dynamic scene shape reconstruction using a single structured light pattern'. In: DOI: 10.1109/CVPR.2008.4587702.

Lekner, John (1987). *Theory of Reflection, of Electromagnetic and Particle Waves*. Springer, pp. 42–47.

Lima-Eriksen, Leik (Feb. 2022). *Interreflection Resilient Patterns for Structured Light 3D Camera Systems*. Project thesis in TFE4580. Department of Electronic Systems, NTNU – Norwegian University of Science and Technology.

Lu, Renfu (2016). *Light Scattering Technology for Food Property, Quality and Safety Assessment*. CRC Press, p. 26. ISBN: 9781482263350.

Manuel, Ryan (2020). *3D Camera Market Analysis Report By Technology, By Application And Segment Forecasts From 2019 To 2025*.

Mickle, Paul (1999). *1961: A peep into the automated future*. URL: http://www.capitalcentury.com/1961.html (visited on 2nd July 2022).

Moreno, D. and G. Taubin (2012). 'Simple, accurate, and robust projectorcamera calibration'. In: *2nd International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*.

OpenCV (2021). *Camera Calibration*. URL: https://docs.opencv.org/3.4/dc/dbb/tutorial_py_calibration.html (visited on 4th Dec. 2021).

OTOY (2022). *About OctaneRender*. URL: https://home.otoy.com/render/octane-render/ (visited on 13th June 2022).

Peterson, W. Wesley (1970). *Error Correcting Codes*. MIT Press, pp. 251–255.

Rifkin, Jeremy (2011). *The Third Industrial Revolution; How Lateral Power is Transforming Energy, the Economy, and the World*. Palgrave MacMillan. ISBN: 978-0230341975.

Rottenfusser, Rudi, Erin E. Wilson and Michael W. Davidson (2022). *The Point Spread Function*. URL: https://www.zeiss.com/microscopy/int/solutions/reference/basic-microscopy/the-point-spread-function.html (visited on 7th June 2022).

Salomon, David, G. Motta and D. Bryant (2006). *Data Compression: The Complete Reference 4th Edition*. Springer. ISBN: 978-1846286025.

Salvi, Joaquim, Jordi Pàges and Joan Batlle (2004). 'Pattern codification strategies in structured light systems'. In: *Agent Based Computer Vision*.

Sergiyenko, Oleg (2010). '3D laser scanning vision system for autonomous robot navigation'. In: *IEEE XPlore*. DOI: 10.1109/ISIE.2010.5637874.

Skotheim, Ø. and F. Couwelleers (2004). 'Structured Light Projection for Accurate 3D Shape Measurement'. In: *International Conference on Experimental Mechanics*.

Smith, Steven W. (1997). *The Scientist & Engineer's Guide to Digital Signal Processing 1st Edition*. California Technical Pub. ISBN: 978-0966017632.

Spinsante, Susanna, Stefano Andrenacci and E. Gambi (June 2011). 'Binary De Bruijn sequences for DS-CDMA systems: analysis and results'. In: *Eurasip Journal*

on Wireless Communications and Networking - EURASIP J WIREL COMMUN NETW 2011. DOI: 10.1186/1687-1499-2011-4.

Steyerl, A., S.S. Malik and L.R. Iyengar (1991). 'Specular and diffuse reflection and refraction at surfaces'. In: *Physica B: Condensed Matter* 173.1, pp. 47–64. ISSN: 0921-4526. DOI: https://doi.org/10.1016/0921-4526(91)90034-C. URL: https://www.sciencedirect.com/science/article/pii/092145269190034C.

Strasburger, Hans, Michael Bach and Sven P. Heinrich (n.d.). 'Blur Unblurred – A Mini Tutorial'. In: *i-Perception* (). DOI: 10.1177/2041669518765850.

Szeliski, Richard (2011). 'Stereo correspondence'. In: *Computer Vision: Algorithms and Applications*. London: Springer London, pp. 467–503. ISBN: 978-1-84882-935-0. DOI: 10.1007/978-1-84882-935-0_11. URL: https://doi.org/10.1007/978-1-84882-935-0_11.

Wallén, Johanna (2008). 'The history of the industrial robot'. In.

Zeidan, Adam (2021). *Industrial Revolution*. URL: https://www.britannica.com/event/Industrial-Revolution (visited on 11th Feb. 2022).

Zivid (2020). *Complete piece picking SKU coverage - Zivid 3D*. URL: https://sketchfab.com/3d-models/complete-piece-picking-sku-coverage-zivid-3d-fb9c3ba63f884338b67a038f2ee8e1ec (visited on 10th May 2022).

Zivid Two Datasheet (May 2021). ZVD2. Rev. 1.0. Zivid.

Appendix

A Occlusion exclusion masks

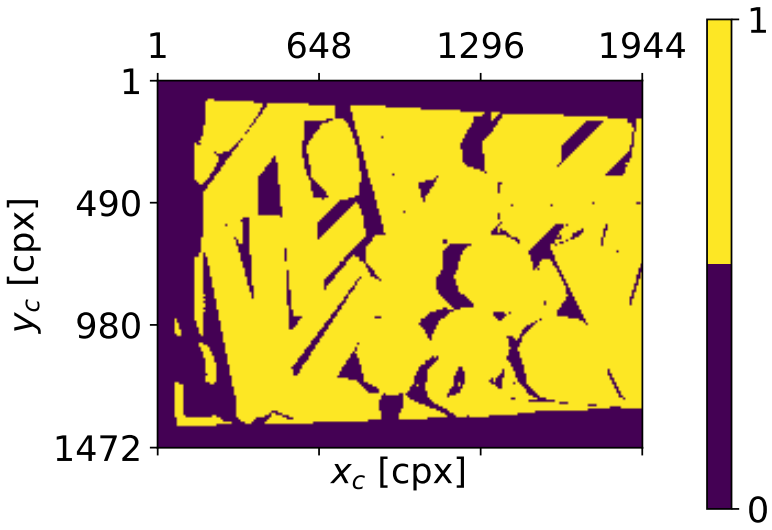


Figure 62: The occlusion exclusion mask for the *Objects in bin* scene at $Z_C = 550\,\text{mm}$.

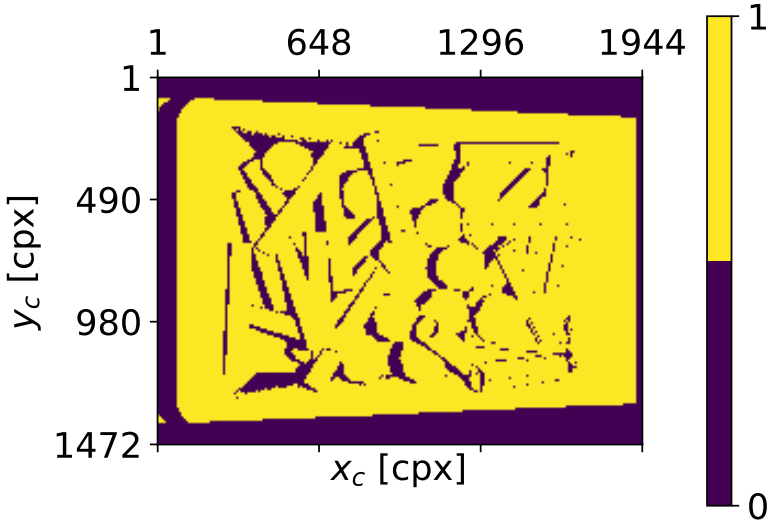


Figure 63: The occlusion exclusion mask for the *Objects in bin* scene at $Z_C = 800\,\text{mm}$.

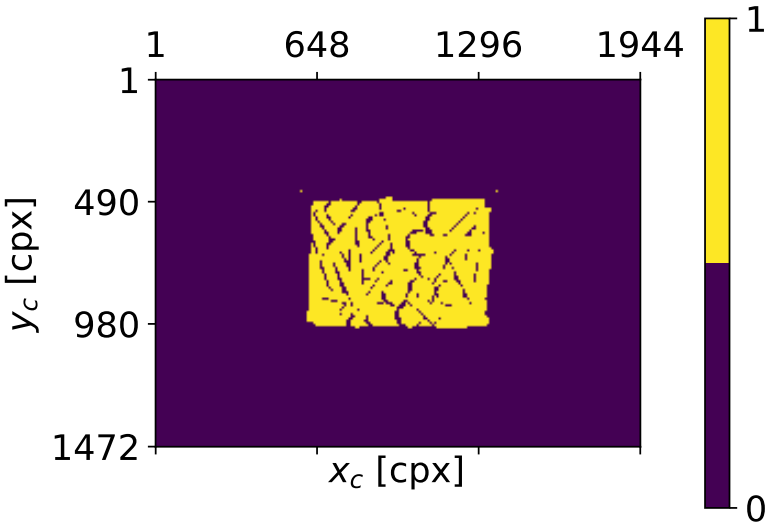


Figure 64: The occlusion exclusion mask for the *Objects in bin* scene at $Z_C = 1400\,\text{mm}$.

Leik Lima-Eriksen

A Distortion-resilient Pattern Codification Strategy for Structured Light 3D Cameras

NTNU
Norwegian University of
Science and Technology

ZiVID