3D Inspection Methods for Specular or Partially Specular Surfaces

Daniel Maestro Watson

Electronics and Computing Department
Faculty of Engineering
Mondragon University

February 6, 2020

3D Inspection Methods for Specular or Partially Specular Surfaces

A thesis submitted
in fulfillment of the requirements for the degree of
Doctor of Philosophy

Daniel Maestro Watson

Supervised by:

Dr. Nestor Arana Arexolaleiba

and

Dr. Alberto Izaguirre Altuna

Committee:

Chair:	Dr. Viviane Cadenat (LAAS-CNRS)
Member:	Dr. Dimitris Chrysostomou (Aalborg University)
Member:	Dr. Bertrand Vandeportaele (LAAS-CNRS)
Member:	Dr. Aritz Legarda Cristobal (Das-nano)
Secretary:	Dr. Luka Eciolaza Echeverria (Mondragon Unibertsitatea)

February 6, 2020

Abstract

Deflectometric techniques are a powerful tool for the automated quality control of specular or shiny surfaces. These techniques are based on using a camera to observe a reference pattern reflected on the surface under inspection, exploiting the dependence of specular reflections on surface normals to recover shape information from the acquired images. Although deflectometry is already used in industrial environments such as the quality control of lenses or car bodies, there are still some open problems. On the one hand, using quantitative deflectometry, the normal vector field and the 3D shape of a surface can be obtained, but these techniques do not yet take full advantage of their local sensitivity because the achieved global accuracies are affected by calibration errors. On the other hand, qualitative deflectometry is used to detect surface imperfections without absolute measurements, exploiting the local sensitivity of deflectometric recordings with reduced calibration requirements. However, this qualitative approach requires further processing that can involve a considerable engineering effort, particularly for aesthetic defects which are inherently subjective.

The first part of this thesis aims to contribute to a better understanding of how deflectometric setups and their calibration errors affect quantitative measurements. Different error sources are considered including the camera calibration uncertainty and several non-ideal characteristics of LCD screens used to generate the light patterns. Experiments performed using real measurements and simulations show that the non-planarity of the LCD screen and the camera calibration are the dominant sources of error. The second part of the thesis investigates the use of deep learning to identify geometrical imperfections and texture defects based on deflectometric data. Two different approaches are explored to extract and combine photometric and geometric information using convolutional neural network architectures: one for automated classification of defective samples, and another one for automated segmentation of defective regions in a sample. The experimental results in a real industrial case study indicate that both architectures are able to learn relevant features from deflectometric data, enabling the classification and segmentation of defects based on a dataset of user-provided examples.

Laburpena

Teknika deflektometrikoak tresna baliotsuak dira gainazal espekular edo distiratsuen kalitate kontrol automatikoa gauzatzeko. Teknika hauetan, kamera bat erabiltzen da ikuskatu beharreko gainazalean islatutako erreferentziazko patroi bat behatzeko, eta isladapen espekularrek gainazalen bektore normalengan duten menpekotasuna ustiatzen dute irudietatik informazio geometrikoa berreskuratzeko. Zenbait industria-aplikaziotan deflektometria jada erabiltzen bada ere –adibidez, betaurrekoen edo autoen karrozerien kalitate kontrolean-, oraindik badaude hobetu beharreko hainbat esparru. Batetik, deflektometria kuantitatiboak aukera ematen du gainazal baten bektore-eremu normala eta 3D forma lortzeko, baina gaur egun teknika hauek ez dute beren sentsibilitate lokal guztia aprobetxatzen kalibrazio-akatsek zehaztasun globalean duten eraginagatik. Bestetik, deflektometria kualitatiboa neurketa absoluturik egin gabe gainazal akatsak antzemateko erabili daiteke, kalibrazio-eskakizun murriztuekin sentsibilitate lokala ustiatuz. Hala ere, teknika horiek algoritmoen garapenean esfortzu handia ekar dezakeen prozesamendu bat eskatzen dute, bereziki bere baitan subjektiboak diren akats estetikoetarako. Hala ere, teknika horiek algoritmoen garapenean esfortzu handia ekar dezakeen prozesamendu bat eskatzen dute, bereziki bere baitan subjektiboak diren akats estetikoetarako.

Tesi honen lehen zatiaren helburua adkizizio sistema osatzen duten gailuek eta horien kalibrazioek neurketa kuantitatiboei nola eragiten dieten hobeto ulertzen laguntzea da. Hainbat errore-iturri hartzen dira kontuan, besteak beste kameraren kalibrazioaren ziurgabetasuna, eta argi-patroiak sortzeko erabilitako LCD pantailen zenbait ezaugarri ez-ideal. Neurketa errealetan eta simulazioetan egindako esperimentuek erakusten dute LCD pantailaren deformazioak eta kameraren kalibrazioak eragindako erroreak direla neurketen akats eta ziurgabetasun iturri nagusiak. Tesiaren bigarren zatian, datu deflektometrikoetatik abiatuz, inperfekzio geometrikoak eta testura-akatsak identifikatzeko ikaskuntza sakoneko metodoen erabilera ikertzen da. Helburu honekin, irudietatik informazio fotometrikoa eta geometrikoa atera eta konbinatzen duten sare neuronal konboluzionaletan oinarritutako bi arkitektura proposatzen dira: bata, lagin akastunak automatikoki sailkatzeko; eta, bestea, laginetako eremu akastunak automatikoki segmentatzeko. Automobilgintza industriako kasu praktiko baten lortutako emaitzek erakusten dute erabilitako arkitekturek datu deflektometrikoetatik ezaugarri esanguratsuak ikas ditzaketela, erabiltzaileak emandako adibide multzo batean oinarrituta gainazal akatsak sailkatu eta segmentatzea ahalbidetuz.

Resumen

Las técnicas deflectométricas son una herramienta valiosa para automatizar el control de calidad de superficies especulares o reflectantes. Estas técnicas se basan en el uso de una cámara para observar un patrón de referencia reflejado en la superficie bajo inspección, explotando la dependencia de los reflejos especulares en la normal de la superficie para recuperar información geométrica a partir de las imágenes adquiridas. Aunque la deflectometría ya se usa en algunas aplicaciones industriales, tales como el control de calidad de lentes o carrocerías de coches, todavía hay algunos problemas abiertos. Por un lado, la deflectometría cuantitativa permite obtener el campo vectorial normal y la forma 3D de una superficie, pero a día de hoy no es capaz de aprovechar al máximo su sensibilidad local ya que la precisión global se ve afectada por errores de calibración. Por otro lado, la deflectometría cualitativa se utiliza para detectar imperfecciones de la superficie sin mediciones absolutas, explotando la sensibilidad local de la deflectometría con requisitos de calibración reducidos. Sin embargo, estos métodos requieren un procesamiento adicional que puede implicar un esfuerzo considerable en el desarrollo de algoritmos, particularmente para defectos estéticos que son inherentemente subjetivos.

La primera parte de esta tesis tiene como objetivo contribuir a una mejor comprensión de cómo el sistema de adquisición y su calibración afectan a las mediciones cuantitativas. Se consideran dife-rentes fuentes de error, incluida la incertidumbre de calibración de la cámara y varias características no ideales de las pantallas LCD utilizadas para generar los patrones de luz. Los experimentos realizados con mediciones reales y simulaciones indican que los errores inducidos por la deformación de la pantalla LCD y la calibración de la cámara son las principales fuentes de error e incertidumbre. La segunda parte de la tesis investiga el uso del aprendizaje profundo para identificar imperfecciones geométricas y defectos de textura a partir de datos deflectométricos. Se adoptan dos enfoques diferentes para extraer y combinar información fotométrica y geométrica utilizando sendas arquitecturas basadas en redes neuronales convolucionales: una para la clasificación automatizada de muestras defectuosas y otra para la segmentación automatizada de regiones defectuosas en una muestra. Los resultados experimentales en un caso de estudio industrial real indican que ambas arquitecturas pueden aprender características relevantes de los datos deflectométricos, permitiendo la clasificación y segmentación de defectos en base a un conjunto de datos de ejemplos proporcionados por el usuario.

Acknowledgments

This journey has been a truly life-changing experience for me and it would not have been possible to accomplish it without the aid and support of many people.

First and foremost, I would like to thank my supervisors Nestor Arana and Alberto Izaguirre for their guidance and support, not only during this thesis, but also during the time I worked in the computer vision lab as an undergraduate. As my teachers and mentors, they have taught me more than I could ever give them credit for here.

I am grateful to Ekide S.L. for the financial support of this thesis; my sincere thanks to Aitor Zubiaurre and Oscar Sevillano for all the trust put in me.

I want to thank Mondragon University for providing the opportunity and resources to conduct this research. I would also like to express my gratitude to Aitzol Iturrospe and to Luka Eciolaza for respectively introducing me to the fields of deflectometry and deep learning, and for all the invaluable help they provided. I am in debt with Julen Balzategui for all the help with the segmentation part, and also with Imanol Andonegui, Marcos Alonso, Aitor Osa, and all the members of the Robotics and Automation group that have helped me get on with this thesis in a way or another.

I would also like to thank LAAS-CNRS and the Robotics, Action, and Perception team (RAP) for accepting me as a visiting PhD, and to Patrick Danes for making it possible. I wish to express my deepest gratitude to Bertrand Vandeportaele for all the ideas and improvements proposed for my research, for taking the time to explain various concepts and tools, and for answering tons of my questions. It was an honor and a privilege to share the office with you.

My sincere thanks to all the PhD students with whom I have shared this adventure. Aritz, Aitor A/L, Alain, Ander, Ane, Enaitz, Iñaki, Javi, Jon, Maite, Oscar, Pablo, Raul, Unai... Thank you for your help, for sharing your knowledge, and for creating a wonderful working atmosphere during these years.

Greetings also to all the people I met during my stage in Toulouse. Camille, Dimitri, Francisco, Jessica, Jorge, Julien, Luis, Philippe-Antoine, Tristan, and the rest of members of RAP, thank you for making me feel at home. To Raúl, my flatmate, thanks for the nice meals and talks. It was a pleasure to share these months with all of you.

Last but not least, I would like to thank my family and friends for supporting me spiritually throughout this thesis, specially to my mother who has been my proofreader so many times, and to Judit who has suffered by my side along the whole process. Without their understanding and unconditional support it would not have been possible to finish this work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter details the context and motivation of the thesis, formulates the problem statement and describes the outline of the rest of this document.

1.1 Context

There is a worldwide trend in advanced manufacturing countries called Industry 4.0, which encourages the use of the latest advances in information and communications technologies to improve productivity and efficiency in manufacturing. The automatic and flexible adaptation of the production chain to short batches or changing requirements is among the key aspects addressed in Industry 4.0 [PTB+15]. Quality control is a significant element in the production chain, and specifically, the zero defect premise still remains as one of the main concerns in industry.

In some technical surfaces, an imperfection or a shape distortion may affect the correct operation or the lifetime of the produced good, and an immaculate surface is needed. In other cases, the perceived quality of a product is affected by defects that are merely aesthetic: when clients pay for a high-end product, they expect that it will not only accomplish its function, but that it will also exhibit the quality paid for. The customer associates a defect-free surface to a global quality guarantee and any visible deviation from the ideal surface, such as a difference in relief, color, shape and/or contrast may affect the perceived quality of an item [MPB13].

In several industrial sectors the quality requirements impose the need for a unitary control of 100% of the production in order to detect any surface imperfection before the product reaches the client. Manufacturers from the automotive industry, home appliances, electronic devices, or jewelery are among the affected sectors. Not only are the final manufacturers affected, but their suppliers are as well. The manufacturers impose demanding quality standards on their suppliers, and include clauses relative to the surface quality in their contracts. Non-compliance with these clauses results in rejections, loss of coverage or, in some cases, even economic penalties and contract terminations.

Often the quality control is performed by trained operators or quality experts. Humans are able to infer surface properties by interpreting the complex interactions between the light and the surface, drawing on their experience and memories [Ade01]. This allows the operators to detect a wide range of defects in different objects. Moreover, the operators are capable of easily adapting to new objects or design changes in the objects to be inspected. However, human operators perform the inspection from a subjective point of view. Their

perception of defects depends on several factors, such as their emotional or physical state. Moreover, surface inspection is a laborious task that requires concentration and operators get tired throughout the working day; fatigue influences the attention that the operators pay to their job. According to several studies in the textile industry [SSGJ99, MP06, SWS08, Mal12], even the most experienced operators are only capable of detecting around 70% of the defects. This clearly shows the need for automated inspections.

Machine vision provides several techniques that can be used to perform an objective automated surface inspection. These inspection systems can be placed at the production line, allowing results to be obtained in real time. In this way, the cause of the defects can be corrected almost immediately, minimizing the number of defective parts produced. Moreover, if the inspection is performed in the same operation that generates the defects, further processing operations of the defective parts can be avoided, preventing unnecessary resource consumption, and therefore reducing the cost of the rejections.

Machine vision techniques can evaluate the quality of a surface by interpreting the images using several tools based on geometry, physics, statistics or machine learning. Despite the continuous advances in these techniques, they are still not able to recreate human perception and none of them is adaptable to all applications. The most obvious limitations of machine vision techniques are the measurement speed and accuracy, but the boundary conditions of the product (material, shape, size, ...) to be inspected and the manufacturing process also need to be taken into account [KRH06].

One of the key conditions is the optical properties of the surface [KRH06]. The apparent shape, the texture and the color of a surface depend on the received illumination and the way in which it is reflected. This interaction between the light and the surface depends on the optical properties of the material and the microstructure of the surface. Likewise, machine vision techniques are conditioned by this optical behavior. Cameras and 3D sensors capture the light reflected from the surface to generate images from which geometrical information must be extracted. This is done based on models that explain the optical phenomena related to light and its reflection, and due to the different ways in which surfaces reflect light, there is no machine vision technique that is valid for all kinds of surfaces.

The measurement of surfaces that exhibit specular reflection characteristics, *e.g.*, mirrors or shiny surfaces, poses a problem for traditional machine vision techniques that are widely used in industry, like stereo vision, laser triangulation or fringe projection systems. These techniques provide a good solution for many applications, but assume a diffusely reflecting surface and fail on specular surfaces [KRH06]. The reflections of the environment are visible on such surfaces and the appearance of the observed scene changes depending on the angle between the viewing direction and the surface orientation. Depending on the degree of specularity, the observed surface is invisible and only the reflection of surrounding environment is visible on the captured images. This view-dependent appearance results in a challenging problem for acquisition systems [IKL+10], as it introduces noise to the measurements.

Deflectometric techniques exploit the dependence of specular reflections on surface orientation to recover information about the shape and surface finish of specular surfaces [BW10]. A controlled environment is created by means of one or more structured patterns, and their reflection on the surface is observed with one or more cameras. The distortions on the observed patterns allow the normal vector field and 3D shape of the surface to be recovered. Deflectometry has mainly two types of applications: quantita-

tive measurements of precision surfaces, such as lenses, telescopes or solar concentrators, and qualitative inspection of more general surfaces, *e.g.*, defect detection in car bodies [KL08] or windshields [HFOE13]. Although there has already been much progress in the field, there is still a series of open problems that need to be solved to fully exploit the capabilities of deflectometry.

On the one hand, the achievable local accuracy is several orders of magnitude higher than the global accuracy, meaning that the local height sensitivity in the nanometer range, cannot be extended to a global height uncertainty valid for all measurement points regardless of the measurement area [FOKH12]. Measurement errors introduced by deflectometric setups and their miscalibration are the main shortcomings of quantitative measurements. Deflectometric setups are extremely prone to calibration errors, and the resulting low-frequency measurement errors increase with the size of the measured area. Several researches have pointed out that in order to reduce these errors, more complex mathematical models should be considered for both the camera and the LCD display used to generate the light patterns. However, little research has been done regarding the impact of these error sources. Thus, there is still room for research and improvement regarding the employed models and calibration procedures.

On the other hand, exploiting deflectometric data for automatic defect detection requires further processing that can involve a considerable engineering effort. Typically pattern recognition algorithms based on shallow learning techniques have been employed for automatic defect classification [Cau10, Zie14, KSM17]. These methods require finding effective/meaningful descriptors and pre-processing operations, which results in a time-consuming trial and error process. The recent advances in deep learning methods allow such feature engineering tasks to be learned from a set of examples, which could be very useful for qualitative inspection.

Advances in the above mentioned models, calibration procedures, and processing techniques can lead to a more wide-spread adoption of deflectometry in industrial applications. This thesis aims to help filling the gap by addressing several of these ideas.

1.2 Contributions

The work of this thesis is divided into two main parts: the first one concentrates on analyzing the main sources of error in quantitative deflectometric measurements, focusing on the Active Reflection Grating Photogrammetry approach [PT05]. This will allow us to identify the limits of actual systems and point out the improvements that should be addressed in future works, in order to make deflectometric systems usable in more applications. The second part focuses on the use of deep learning techniques for exploiting deflectometric data in qualitative surface inspection.

The work presented in this doctoral thesis consists of three main contributions:

- We analyze the uncertainties arising from a camera calibration based on the pinhole model. These uncertainties are propagated through the different stages of the measurement process in order to quantify the uncertainties in the measured normal vector field and the triangulated surface. An analysis of a real setup is performed both analytically and based on simulations, so as to identify the camera parameters that contribute most to the measurement uncertainty.

- We analyze the influence of the non-planar shape and refractive layers of an LCD screen in the final measurements. In order to investigate their effects, a real setup is characterized and used to measure a planar first surface mirror, and further simulations are performed in order to isolate the investigated error sources.

- We propose two different approaches for qualitative inspection of specular surfaces using deep learning methods for exploiting deflectometric data obtained with a monoscopic setup. Both approaches use architectures based on convolutional neural networks (CNN) to learn to extract and combine features based on a user-provided example dataset. The former is oriented to defective sample classification and the latter to defective region segmentation.

1.3 Publications

Part of the work covered in this thesis has already been published in a peer-reviewed journal and different international conferences:

Journal papers

- D. Maestro-Watson, J. Balzategui, L. Eciolaza, and N. Arana-Arexolaleiba. Deflectometric data segmentation for surface inspection: a fully convolutional neural network approach. Deflectometric data segmentation for surface inspection: a fully convolutional neural network approach, *Journal of Electronic Imaging*, 29(4), 041007, 2020.

Conference papers

- D. Maestro-Watson, A. Izaguirre, N. Arana-Arexolaleiba, and A. Iturrospe. A simple deflectometric method for measurement of quasi-plane specular surfaces. In *2015 IEEE International Workshop of Electronics, Control, Measurement, Signals and their Application to Mechatronics (ECMSM)*, 2015.

- D. Maestro-Watson, A. Izaguirre, and N. Arana-Arexolaleiba. LCD screen calibration for deflectometric systems considering a single layer refraction model. In *2017 IEEE International Workshop of Electronics, Control, Measurement, Signals and their Application to Mechatronics (ECMSM)*, 2017.

- D. Maestro-Watson, J. Balzategui, L. Eciolaza, and N. Arana-Arexolaleiba. Deep learning for deflectometric inspection of specular surfaces. In *International Joint Conference SOCO'18-CISIS'18-ICEUTE'18*, 2019.

- D. Maestro-Watson, J. Balzategui, L. Eciolaza, and N. Arana-Arexolaleiba. Deflectometric data segmentation based on fully convolutional neural networks. In *SPIE International Conference on Quality Control by Artificial Vision (QCAV)*, volume 11172, 2019. [Best Paper Award]

1.4 Research Projects

Part of the work covered in this thesis has been developed during research projects funded by the Department of Education, Language Policy and Culture of the Basque Government, including:

- IDEFIX (No. PUE 2018-06)
- DEFECTECH (No. ZL-2016/00261)
- PANORAMIX (No. UE-2014-14)

1.5 Outline

The rest of the document is organized as follows:

In Ch. 2 the background necessary for the understanding of the performed work is presented. In particular, the state of the art in deflectometry systems and the theory of error and uncertainty propagation is considered. In Ch. 3 the pinhole camera model and calibration are reviewed, and an uncertainty model is developed for the camera rays, which is afterwards employed to estimate the resulting measurement uncertainties in an Active Reflection Grating Photogrammetry setup. In Ch. 4 different error sources arising from the LCD screen model are analyzed. In particular the effects of ignoring the non-planar shape and the refraction in the translucent layers of an LCD screen are simulated and compared to the errors in a real measurement. In Ch. 5 a review of the state of the art in qualitative deflectometric techniques for surface defect detection is performed, and two different methods based on deep learning are proposed to process the data obtained with a monoscopic deflectometric setup. Finally, Ch. 6 ends with the conclusion and future research lines.

Chapter 2

Deflectometry - Theoretical Framework

This chapter provides the technical background required to understand the material in the following chapters. In order to do so, first we introduce the concept of specular surfaces (Sec. 2.1). Afterwards, the basics of deflectometry and deflectometric measurements are reviewed (Sec. 2.2), with an special emphasis on the Active Reflection Grating Photogrammetry (ARGP) approach (Sec. 2.3), which will be used for the absolute measurements later in Ch. 3 and Ch. 4. Finally, the mathematical concepts and tools that will be employed to analyze the errors and uncertainty in such measurements are described (Sec. 2.4).

2.1 Surface Appearance

The appearance of a surface is defined by the way it interacts with light. Light is an electromagnetic radiation that is propagated through a uniform medium until it makes contact with a different material. For example, when a surface is illuminated, light propagates through the air until it hits the surface. When light encounters a different material there are a series of interactions between light photons and the particles of the surface, and as a result, the light can be absorbed, transmitted or reflected. These interactions depend on the physical properties of the surface, the wavelength of the light and the incident light direction.

This work is only concerned with opaque surfaces, *i.e.*, surfaces that do not allow light to be transmitted through them. These surfaces are visible because they reflect light and their appearance is defined by the optical properties of the material and microstructure of the surface. The color of a surface is perceived because the material of which it is made reflects some wavelengths and absorbs others. For example, a surface appears white when it reflects all the wavelengths and a surface appears red when it only reflects wavelengths close to the color red. The way in which light is reflected allows us to infer other properties of the surface such as its texture or its apparent shape.

The reflection law describes the relation between a light ray that hits the surface at one point and the light ray that is reflected from it. The angle of an incident light ray (Θ_i) is the same as the reflected angle (Θ_r), both measured with respect to the surface normal as shown in fig. 2.1. This is true at a microscopic level, when we analyze the way in which a photon is reflected when it collides with a particle or an atom at the surface.

But if we analyze how light is reflected at a surface area in a macroscopic way, the effect of the microstructure of the surface can be observed by the way it reflects light.

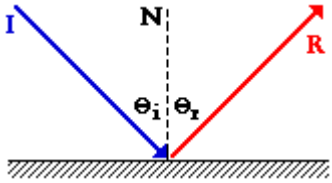

Figure 2.1: Reflection of a light ray

If a perfectly flat surface is illuminated from a constant direction, the light will also be reflected in a constant direction as shown in Fig. 2.2(a). This kind of surface is called a specular surface. In reality no such "perfectly flat" surface exists, all surfaces have certain roughness, even if it is at such a small scale that the irregularities are formed by atoms. A surface is considered optically flat at a microscopic level if its roughness is negligible with respect to the wavelength λof the light that illuminates it. According to [HE11], a surface is considered optically smooth if its height variation within the lateral resolution of the observation is smaller than $\lambda/4$. As shown in Fig. 2.3(a), in this kind of surfaces the surface itself is invisible and only the reflection of the surrounding environment is visible on them. A mirror is a typical example of this kind of surface.

Figure 2.2: Reflection in (a) smooth surfaces and (b) rough surfaces

What happens with surfaces that are not optically smooth? In rough surfaces the orientation of its particles is random and, if they are illuminated from a constant direction, the light is reflected in a diffuse way; it is scattered in different directions, as shown Fig. 2.2(b). A surface is considered to be Lambertian or ideally diffuse when it reflects the same quantity of light in all directions, as shown in Fig. 2.3(b), and therefore appears equally illuminated from all viewpoints, like for example in raw rubber surfaces.

The majority of surfaces are not ideally Lambertian nor ideally specular, but a combination of both, as can be seen in Fig. 2.3(c). These surfaces scatter light from a single source into many directions, but unequally. This kind of surfaces can be modeled as a combination of a diffuse component and a specular component. Surfaces with small roughness reflect light in a way similar to perfectly smooth surfaces, but as the surface roughness increases, the specular reflectance is attenuated, and a directional diffuse lobe emerges. The appearance of these surfaces is also a combination of the lambertian and specular appearance, resulting in a glossiness of different degrees depending on its properties. In these surfaces, the surface itself is visible, but some viewpoint-dependent highlights appear on it.

In this research, we are only interested in surfaces that present a sufficiently high specular component, such that the specular direction can be clearly distinguished.

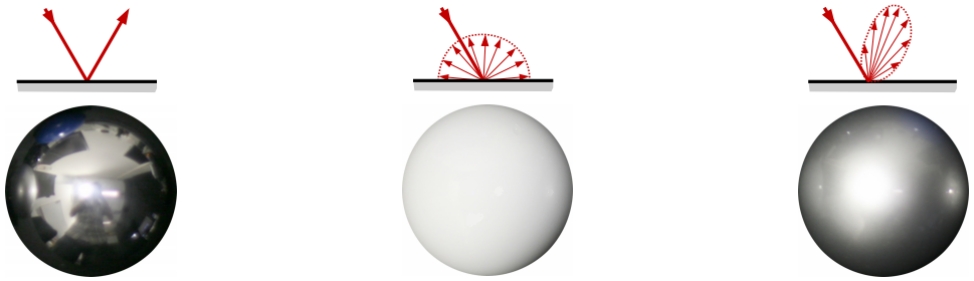

Figure 2.3: Reflection model and appearance of surfaces with different optical properties [HIH+13]: (a) specular reflection, (b) Lambertian reflection, and (c) mixed reflection

2.2 Deflectometry

Deflectometry is a field of optical metrology that encompasses a series of full-field measurement techniques that allow to obtain the shape of specular or partially specular surfaces. It has been a field in which many researchers have focused on over the last 15 years, resulting in a series of similar techniques called by a variety of names, such as Shape from Distortion [TLGS02], Phase Measuring Deflectometry [KKH04], Fringe Reflection [BLvKJ04], Reflection Grating Photogrammetry [PT05] or Shape from Specular Reflection [BW10], among others. A brief overview of its working principles is given in this section, the interested reader can refer to more detailed reviews of the latest advances in the field in Refs. [ZWH+17, HIZA18].

2.2.1 Foundations

A typical deflectometric system consists of a camera, or several cameras, focused on the surface under inspection and an LCD screen displaying a series of spatially coded patterns placed nearby, such that the camera observes such reference structure from its reflection on the surface. As these patterns encode the spatial position of each LCD pixel, their recordings provide a mapping from every camera pixel to the corresponding observed LCD location. This mapping is called a light map. The high sensitivity of specular reflections to surface orientation produces changes in the observed LCD locations, resulting in distortions on the captured pattern for even slight shape deviations, as shown in fig. 2.4.

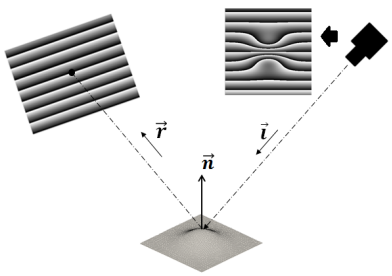

Figure 2.4: Sketch of a typical deflectometric measurement setup. An LCD displays a sinusoidal pattern and the surface is in the light path between the LCD and the camera. The captured pattern is visibly distorted by the shape of the surface.

Quantitative deflectometric techniques exploit the acquired mapping to provide the

normal vector field of a surface by measuring the deflection that light rays suffer when they are reflected on the surface under inspection. Assuming that a considerable part of the light is reflected following the specular reflectance model, the reflection law (eq. 2.1) can be used to model the recovery of the normal vector field for each pixel in the camera:

$$\vec{\boldsymbol{n}} = \hat{\boldsymbol{r}} - \hat{\boldsymbol{i}} \tag{2.1}$$

where $\vec{\boldsymbol{n}}$ is the normal of the surface, $\hat{\boldsymbol{i}}$ is the unit vector representing the direction of the incident ray and $\hat{\boldsymbol{r}}$ is the unit vector representing the direction of the reflected ray.

As shown in fig. 2.4, the incident ray $\vec{\boldsymbol{i}}$ corresponds to the viewing direction, and the reflected ray $\vec{\boldsymbol{r}}$ corresponds to the direction of the ray that goes from the surface to the point on the pattern. The normal vector $\vec{\boldsymbol{n}}$ corresponding to the surface point viewed by the camera is defined as the half vector between the incident and reflected directions. In order to recover the normal vectors, the light-map is used to obtain the location in 3D space of the point on the screen viewed on each pixel of the image. With a calibrated camera, the observation direction for each camera pixel is also known. However, the direction of the corresponding reflected ray, and therefore the normal vector, is not easy to calculate due to the fact that the height and orientation of an observed surface point are coupled in the measurement. This is known as the height-ambiguity problem [BHB11], and there is no unique solution if there is no additional information available.

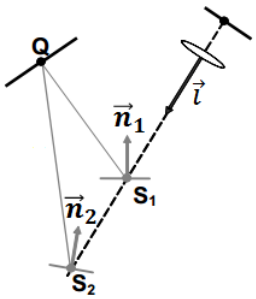

Figure 2.5: Height-ambiguity problem in deflectometric measurements: Reflected ray direction $\vec{\boldsymbol{r}}$ is undetermined if the height of the point on the surface is not known because the same point in the screen $\boldsymbol{Q}$ can be seen through many hypothetical surface points $\boldsymbol{S_i}$ along the camera ray $\vec{\boldsymbol{i}}$.

Fig. 2.5 shows an example of the ambiguity for the measurement of a single point. The incident ray $\vec{\boldsymbol{i}}$ is known from camera calibration, $\boldsymbol{Q}$ is the point on the LCD viewed by the camera, which is computed from observations, but the reflection direction $\vec{\boldsymbol{r}}$ depends on the location of the surface point $\boldsymbol{S}$, which in turn is what we want to measure. In the example the normal $\vec{\boldsymbol{n_1}}$ corresponds to the real surface point $\boldsymbol{s_1}$, which sees the point $\boldsymbol{Q}$ through the reflected ray $\vec{\boldsymbol{r_1}}$. A hypothetical point $\boldsymbol{s_2}$, with a different normal $\vec{\boldsymbol{n_2}}$, can also reflect the point $\boldsymbol{Q}$, following a different path $\vec{\boldsymbol{r_2}}$. If no additional information is used, there is an infinite set of possible solutions, one for each point $\boldsymbol{S_i}$ along $\vec{\boldsymbol{i}}$, which also fulfill the reflection law with a different normal vector $\vec{\boldsymbol{n_i}}$. The main differences between the quantitative deflectometric methods that can be found in the literature lie in the way they solve this ambiguity, which is sometimes called measurement regularization.

In addition to the normal vector field, the 3D shape of the surface can also be reconstructed. Some techniques make it possible to obtain the surface shape by triangulation, but generally a more accurate height map can be obtained by integrating the normal field

[EKKH08, KLF+14, QDA18]. Surface curvatures can also be calculated by differentiating the obtained normal field [KKH03]. Deflectometric techniques can provide a local sensitivity in the order of some ηm, so they can be considered ideal to detect and measure local surface defects. In contrast, the global accuracy is usually several orders of magnitude worse, mainly due to calibration errors that get spread in the integration, and it is quite difficult to obtain global measuring errors better than 1 µm in 100 mm diameter optical surfaces [HFOE13]. Figure 2.6 shows the difference between local and global accuracy.

Figure 2.6: Deflectometric sensitivity: (a) Local accuracy can reach down to a few nm. (b) Global accuracy is usually several orders of magnitude worse. [FKR+12]

A typical deflectometric measurement consists of several procedures that can be arranged into three groups: Calibration, Acquisition, and Processing. Fig 2.7 shows a schematic work-flow of these procedures, the remainder of this section will briefly introduce them.

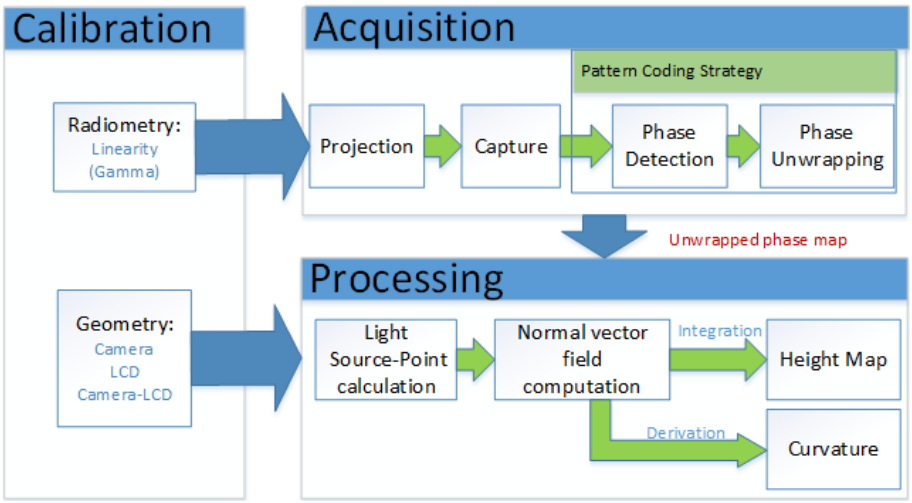

Figure 2.7: Work-flow of a typical deflectometric measurement

2.2.2 Calibration

The calibration of a quantitative deflectometric system involves several procedures, most works only deal with the first two.

- **Camera calibration**: This procedure consists in obtaining the camera parameters corresponding to a projection model, such that the direction of the camera rays can be computed. The extended pinhole model is the most commonly employed one, although more complex models have also been proposed. The procedure is based on the observation of a reference target from different viewpoints. Each viewpoint provides a series of correspondences between the camera and the reference target, which are afterwards used to estimate the internal camera parameters as well as the relative poses of the calibration target in a non-linear minimization procedure [Zha00, Hei00, Tsa87].

- **Camera-Screen calibration**: This procedure consists on obtaining the pose of the screen with respect to the camera frame. As the camera does not usually have a direct view of the screen, an additional mirror is required to perform the calibration. The procedure is similar to the camera calibration, but the reference pattern is displayed on the screen instead of using a calibration target [KKH04, FY07, PT05].

- **Screen calibration**: This procedure can involve the estimation of different non-ideal characteristics of the screen, such as its non-planar shape or the refraction the light rays suffer in its translucent layers [KKH04, PFT13, OH14, BKB19].

- **Radiometric calibration**: This procedure deals with the non-linear radiometric response of screen and camera. The obtained curves are used to adapt the intensities of the displayed patterns [FPT10, PFT13].

2.2.3 Acquisition

The acquisition procedure consists of sequentially displaying a series of patterns that are captured by the camera through the reflection on the surface under inspection. These patterns encode the coordinates for each pixel on the LCD, allowing to relate each pixel on the image to the point on the screen that is visible in it. Several pattern coding strategies can be employed, including binary patterns [SPW$^+$10], sinusoidal fringes [KKH04], crossed fringes [LOYH14], or colored fringes [TLGS05], among others [SFPL10]. Sinusoidal fringes that encode the LCD position in their phase are the most common ones in deflectometry due to the fact that they are continuous, allowing sub-pixel detection, and that they are robust to blurring and illumination changes. The phase recovery can be done using several methods such as phase-shifting [SB06] or Fourier transform [TIK82]. These methods provide phase values that are in the range $[0, 2\pi]$. Thus, an unwrapping operation has to be performed in order to obtain the absolute phase [GP98]. Finally, the light-map is obtained, providing a mapping from each camera pixel to the point on the screen viewed on it.

2.2.4 Processing

The processing stage consists of the operations required to obtain absolute shape information from the acquired light-maps. Note that the deflectometric setup itself depends on the employed regularization method. The following paragraphs outline the main approaches.

One approach has been to simplify the problem by assuming a previous knowledge of the surface. Assuming that the surface is plane, the distance to the screen can be considered constant, and if the placement of the surface is approximately known, the normal vector field can be easily obtained [BLvKJ04, Sur04, MWIAAI15]. For non-planar surfaces, the use of mathematical models based on Chebyshev polynomials, Zernike polynomials, or B-splines has also been proposed, which is called modal phase measuring deflectometry (MPMD) [HXG$^+$16]. These simplifications considerably relax calibration requirements and simplify the processing operations.

Another approach is to use an external measurement to obtain the location of at least one point. From there it is straight forward to obtain the normal at such point. Having a point and a normal, the location of the adjacent points can be approximated, assuming a smooth and continuous surface [TLGS05]. The whole surface can thus be

reconstructed iteratively in a region growing approach. The initial point can be obtained using a laser pointer or a confocal sensor [BLH+13], for example. According to [Rap12], the integration error that accumulates as the distance from the starting point increases can not be avoided. In the case of surfaces with a diffuse reflection component, the combination of deflectometric measurements with fringe projection [San14] or shape from shading techniques [BWB06] has also been proposed.

The most common approach is to obtain additional information with multiple cameras or views [BS03, KKH04, PT05, AHG05, NWR08, Rap12, WORK13]. These approaches generally assume a certain degree of smoothness on the surface and solve the ambiguity with optimizations based on normal consistency measures. These methods can fail in certain cases, as additional ambiguities can arise in the vicinity of edges or holes [Rap12], or, in rare cases, when certain surface and stereo configuration combinations induce multiple similar local disparity fields [NWR08, BW10].

A different approach consists on using a global procedure based on finite-elements method [Bal12] that takes into account that the location of the points and the corresponding normal vectors are not independent upon each other. Using an initial guess for the surface shape, e.g. a simple geometric primitive like a plane or sphere, the corresponding normals are computed. The initial shape guess is also used to compute normals that conform with deflectometric observations. The differences between these sets of normals are reduced by a finite-elements analysis, in which the surface is iteratively deformed until an agreement between all measurements is achieved.

An alternative approach, called active reflection grating photogrammetry, consist in adding measurements with additional LCD positions that allow the reflection direction to be identified unambiguously [PT05, BSG06, RJGZ14]. This approach requires a precise mechanical translation stage, but has the advantage that each point in the image is computed individually, without taking into account the neighboring points, nor making assumptions on the underlining surface shape. Chapters 3 and 4 are based on this approach, which will be explained in the next section.

In addition, there have been many works that focus on more qualitative approaches that do not provide absolute shape measurements, but rather exploit the distortions of the acquired patterns in order to obtain shape information. These approaches can be used to detect local surface defects with less strict calibration requirements. This approach is adopted in Ch. 5, and a literature review regarding these methods can be found in Sec. 5.2.1.

2.3 Active Reflection Grating Photogrammetry

Active Reflection Grating Photogrammetry (ARGP) [PT05] provides quantitative measurements, allowing the computation of surface normal vectors, as well as absolute 3D object coordinates. Measurements are done pixel-wise, i.e. each camera pixel obtains an independent result, therefore enabling the measurements of discontinuous surfaces.

The measurement principle of ARGP is based on reflection law (eq. 2.1), that relates the directions of the incident and reflected light at a surface point $\mathbf{M}$ with its surface normal $\mathbf{n}$. Fig. 2.8 shows a scheme of the measurement principle.

The measurement procedure consists on identifying the camera ray $\mathbf{i}$ and the reflected ray $\mathbf{r}$ for each camera pixel $\mathbf{m}$. Obtaining the direction of $\mathbf{i}$ is straightforward from the camera model if the camera has been previously calibrated (this will be explained

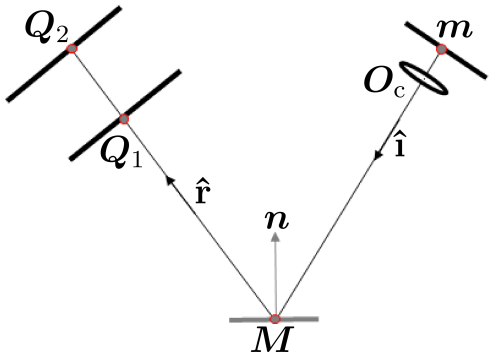

Figure 2.8: Reflection Grating Photogrammetry measuring principle: A surface point **M** and its normal vector **n** are defined by the intersection of the incident and reflected rays

in ch. 3). In order to identify **r**, a deflectometric registration is performed: a series of patterns is displayed on the screen and captured by the camera. These patterns are constructed following a coding strategy that allows unambiguously identifying the pattern location observed in each camera pixel, which can be converted to 3D point $\mathbf{Q}_1$ using the geometric calibration of the system (this will be explained in depth in ch. 4). As has been previously explained, this is not enough to obtain **r** due to the deflectometric ambiguity. ARGP estimates the reflected ray direction by obtaining at least another point $\mathbf{Q}_2$ from a pattern located in a different position, which is achieved by moving the LCD screen with a translation stage. In this way, the surface normal $\hat{\mathbf{n}}$ is determined following Eq. 2.1 and the surface point **M** can be obtained by triangulation. In this work we will not consider the integration of the surface normals.

2.4 Measurement Errors and Uncertainty

The goal of a measurement is to determine the value of a particular quantity (the measurand). A measurement is never perfect because, at some scale, there will always be errors that will affect its accuracy. In this context, the measurement accuracy refers to the closeness of agreement between the measured and the true quantity values of a measurand, and the measurement error is defined as the difference between these two quantities.

Measurement errors are generally caused by a combination of different sources. The most obvious source of measurement errors resides in the measuring instrument itself. For example, errors occur due to imperfections or limitations of a device, or inexactness in its calibration. Also, errors can be originated from incorrect theory or simplifications of the system model, or inexact values of constants and parameters imported from external sources. In addition, the environment can also be a source of measurement errors, as external influences such as temperature, pressure, humidity, or vibrations can affect both the instrument and the measurand.

Some of these errors will result in random effects that vary in an unpredictable manner in replicate measurements, thereby introducing uncertainty in the measurements. Other errors will have systematic effects on the measurements, meaning that they will remain constant or vary in a predictable manner in replicate measurements. However, even if these systematic errors can be corrected, such corrections will not be perfect; they will be an approximation that will still contribute to uncertainty in the measurement. For this

reason, in several fields such us physics or metrology an uncertainty statement is required in order to fully describe the result of a measurement.

The rest of this section briefly introduces the concepts and mathematical tools that will be used in the first part of this thesis in order to study the effects of several sources of error in an Active Reflection Grating Photogrammetry Setup.

2.4.1 Uncertainty

Uncertainty is *a parameter, associated with the result of a measurement, that characterizes the dispersion of the values that could reasonably be attributed to the measurand* [JCGM 100:2008]. It is common to express the spread of a measured scalar quantity x by its standard uncertainty $u(x)$, which represents the standard deviation of the distribution that would result from an infinite series of repeated measurements. In this way, the standard coverage interval containing the true quantity value is defined as $X = x \pm u(x)$. Similarly, when the measurand corresponds to a vector valued quantity, the covariance matrix Σ_x is used in order to also account for the correlation of the components of the vector. In this case, the standard coverage interval is replaced by a standard coverage region.

In some applications, particularly in the areas of health and safety, the expandend uncertainty is preferred instead of the standard uncertainty. The expanded uncertainty represents a larger fraction of the distribution of values that could be attributed to the measurand. This parameter is computed by multiplying the standard uncertainty by a coverage factor k, which is selected according to the coverage probability required for the interval (usually k is in the range of 2 to 3) [JCGM 100:2008]. If the underlaying probability distribution is known, k provides a coverage interval $X = x \pm ku(x)$ corresponding to a particular level of confidence p_k. For example, if x corresponds to an approximately normally distributed scalar variable, the level of confidence resulting from $k = 1, 2, 3$ respectively corresponds to approximately 68%, 95%, and 99%.

For a normally distributed 2D random variable $\mathbf{x} = [x, y]$, the coverage region is described by an elliptical iso-contour of the Gaussian beam. The major and minor axis of such elliptical region are determined by the covariance matrix Σ_x. If x and y are not correlated, Σ_x is a diagonal matrix, and the corresponding coverage region will be described by an axis aligned ellipse:

$$\left(\frac{x}{u(x)}\right)^2 + \left(\frac{y}{u(y)}\right)^2 = k \tag{2.2}$$

with k being a scale factor which determines the p coverage interval. Thus, the half length of the ellipse axes will be given by $u(x)\sqrt{k}$ and $u(y)\sqrt{k}$. If x and y are actually correlated, the major axis will be rotated by α with respect to the x axis. The directions of the major and minor axes are then given by the eigenvectors (x'=$\mathbf{v}_1$ and y'=$\mathbf{v}_2$).

The distribution of the squared sum of two normally distributed and independent random variables follows a χ^2 distribution with two degrees of freedom, thus k must be selected such that there is a probability of p that the left side of the equation is smaller or equal to k. This value can be computed from the cumulative χ^2 distribution, but usually it is taken from precomputed tables. Typical values of k are 2.2977 ($p_k = 0.683$, standard error ellipse), 5.99146 ($p_k = 0.95$), or 9.21034 ($p_k = 0.99$).

Similarly, for n dimensional random variables, an ellipsoidal (3D case) or hyper-ellipsoidal coverage region can be described using the corresponding chi-square distribution with n degrees of freedom.

2.4.2 Uncertainty Propagation

The measurement result is not usually obtained directly from a single measured quantity, but is instead computed from several operations involving one or more input quantities. The involved quantities can be values directly measured or values imported from an external source, *e.g.*, calibration parameter. Thus, in order to obtain an estimate of the uncertainty in the measurand, the uncertainties of the individual input quantities need to be propagated through the different operations composing the measurement process.

For this purpose, the Joint Committee for Guides in Metrology defines two methodologies for uncertainty assessment: the propagation of uncertainties (GUM) and propagation of distributions (MCM). In both cases it is necessary to model the measurement process as a function of the input parameters, *e.g.*, $f = (x_1, x_2, ..., x_i)$, and to have some knowledge of their uncertainty, but their underlying concept is different.

The GUM approach is based on the analytical propagation of standard uncertainties using the generalized law of propagation of uncertainty, which is rooted on first order Taylor expansions and provides a first-order approximation of the standard uncertainty. This approach has several limitations, the main ones being (i) the non-linearity of f needs to be negligible compared to the magnitude of the uncertainties, (ii) the probability density function (PDF) of the measurand has to be reasonably symmetric, and (iii) it must be possible to compute the partial derivatives $\frac{\partial f}{\partial x_i}$. This methodology is defined in [JCGM 100:2008] for scalar valued measurement results and in [JCGM102:2011] for vector valued ones.

In contrast, the MCM approach is based on computer simulations using the Monte Carlo method. With this approach, the PDF of every input quantity needs to be known, thus it requires more knowledge about the uncertainty of the input quantities. However, there are no requirements regarding the linearity of f or the symmetry of the PDFs, and no partial derivatives need to be computed. Therefore, MCM can be applied to a broader class of problems and it is often used to validate the results obtained by the GUM methodology. The MCM methodology is defined in [JCGM101:2008].

In this work both methodologies are employed and they will be briefly described in the following sections.

2.4.3 Taylor Approximations (GUM)

Scalar input, scalar output The uncertainty propagation for a measurement consisting on a simple mathematical model $y = f(x)$, can be derived as follows:

A Taylor series expansion provides an approximation of f around a point a in the form of an infinite series. If f is linear in a small region around a, the series can be simplified to:

$$f(x) \approx f(a) + (x - a) \cdot f'(a) \tag{2.3}$$

where $f'(a)$ represents the first derivative of f evaluated at the point a.

Let the input parameter of the measurement have a small error δ_x around the true value $\bar{x}$, such that $x = \bar{x} + \delta_x$. Equation 2.3 can be used to quantify the error δ_y on the measured value, such that $y = \bar{y} + \delta_y$, by substituting $a = \bar{x}$ and $x - a = \delta_x$.

$$f(x) = \bar{y} + \delta_y \approx f(\bar{x}) + \delta_x f'(\bar{x}) \tag{2.4}$$

Considering that $\bar{y} = f(\bar{x})$, then δ_y is given by:

$$\delta_y \approx \delta_x f'(\bar{x}) \tag{2.5}$$

Usually the true value $\bar{x}$ of the input quantity x is not exactly known during a measurement, but instead an estimate $\hat{x}$ with a standard uncertainty u_x is available. Then, the measured value y is a random variable in the form of $y = \hat{y} \pm u_y$, where $\hat{y} = f(\hat{x})$ is the estimated measure and u_y is its associated standard uncertainty. Using an approach similar to eqs. 2.3-2.5, the best estimate of the output quantity is given by [Arr98]:

$$\hat{y} = \mu_{\mathrm{y}} = \mathbb{E}[y] = \mathbb{E}[f(\hat{x}) + (x - \hat{x}) \cdot f'(\hat{x})] = f(\hat{x}) + (\mathbb{E}[x] - \hat{x})\, f'(\hat{x}) = f(\hat{x}) \tag{2.6}$$

and the uncertainty in the estimated value is given by [Arr98]:

$$\sigma_y^2 \;=\; \mathbb{E}\Big[(y - \mu_{\mathrm{y}})^2\Big] \;\approx\; \mathbb{E}\Big[\big((x - \hat{x})\, f'(\hat{x})\big)^2\Big] \;=\; \mathbb{E}\Big[(x - \hat{x})^2\Big]\big(f'(\hat{x})\big)^2 \;=\; \big(u_x f'(\hat{x})\big)^2 \tag{2.7}$$

Thus, the standard uncertainty is given by:

$$u_y = \sqrt{\sigma_y^2} = \sqrt{u_x^2 \cdot \big(f'(\hat{x})\big)^2} \tag{2.8}$$

Vector input, scalar output Similarly, for more complex functions that depend on several input quantities, such as $y = f(\mathbf{x})$, with the input quantity vector defined as $\mathbf{x} = [x_1, x_2, \ldots, x_n]$, the multivariate Taylor approximation around $\mathbf{a} = [a_1, a_2, \ldots, a_n]$ gives us:

$$f(\mathbf{x}) \approx f(\mathbf{a}) + \sum_{i=1}^{n} (x_i - a_i)\, \frac{\partial f(\mathbf{a})}{\partial x_i} \tag{2.9}$$

Evaluating 2.9 at $\mathbf{a} = \bar{\mathbf{x}}$, the output error δ_{y} can be estimated by propagation of the inputs errors $\boldsymbol{\delta}_{\mathrm{x}} = [\delta_{x1}, \delta_{x2}, ..., \delta_{xn}]^{\mathrm{T}}$:

$$\delta_{\mathrm{y}} \approx \sum_{i=1}^{n} \delta_{x_i} \frac{\partial f(\bar{\mathbf{x}})}{\partial x_{\mathrm{i}}} = \boldsymbol{\delta}_{\mathrm{x}} \mathbf{J}_{\mathrm{y}} \tag{2.10}$$

where $\delta_{x_{\mathrm{i}}} = x_{\mathrm{i}} - \bar{x}_{\mathrm{i}}$, and $\mathbf{J}_{\mathrm{y}}$ represents the Jacobian vector containing the sensitivity coefficients:

$$\mathbf{J}_{\mathrm{y}} = \frac{\partial y}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial f(\bar{\mathbf{x}})}{\partial x_1} & \frac{\partial f(\bar{\mathbf{x}})}{\partial x_2} & \frac{\partial f(\bar{\mathbf{x}})}{\partial x_3} & \cdots & \frac{\partial f(\bar{\mathbf{x}})}{\partial x_n} \end{bmatrix} \tag{2.11}$$

with $\frac{\partial f(\bar{\mathbf{x}})}{\partial x_{\mathrm{i}}}$ representing the partial derivative of the function with respect to the ith input, evaluated at $\bar{\mathbf{x}}$.

In the same way, when only estimates of the input are known, taking $\mathbf{a} = \hat{\mathbf{x}}$, the estimate for the output quantity is given by [Arr98]:

$$\hat{y} = \mu_{\mathrm{y}} = \mathbb{E}[y] = \mathbb{E}\left[f(\hat{\mathbf{x}}) + \sum_{i=1}^{n} (x_i - \hat{x_i}) \frac{\partial f(\hat{\mathbf{x}})}{\partial x_i} \right] = f(\hat{\mathbf{x}}) + \sum_{i=1}^{n} (\mathbb{E}[x_i] - \hat{x_i}) \frac{\partial f(\hat{\mathbf{x}})}{\partial x_i} = f(\hat{\mathbf{x}}) \tag{2.12}$$

and the uncertainty propagated from the input quantities is given by [Arr98]:

$$\sigma_y^2 = \mathbb{E}\Big[(y - \mu_{\mathrm{y}})^2\Big] \approx \mathbb{E}\left[\left(\sum_{i=1}^{n} (x_i - \hat{x_i}) \frac{\partial f(\hat{\mathbf{x}})}{\partial x_i} \right)^2 \right] = \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\partial f(\hat{\mathbf{x}})}{\partial x_i} \frac{\partial f(\hat{\mathbf{x}})}{\partial x_j} \mathbb{E}[(x_i - \hat{x_i})(x_j - \hat{x_j})] = \mathbf{J}_{\mathrm{y}} \mathbf{\Sigma}_{\mathbf{x}} \mathbf{J}_{\mathrm{y}}{}^{\mathrm{T}} \tag{2.13}$$

where $\mathbf{\Sigma}_{\mathrm{x}}$ is the covariance matrix describing the pairwise joint variability of the input parameters, with $\sigma_{x_i x_{\mathrm{i}}} = \sigma_{x_i}^2 = u_{x_i}^2$:

$$\Sigma_{\mathrm{x}} = \begin{bmatrix} \sigma_{x_1}^2 & \sigma_{x_1 x_2} & \sigma_{x_1 x_3} & \cdots & \sigma_{x_1 x_{\mathrm{n}}} \\ \sigma_{x_1 x_2} & \sigma_{x_2}^2 & \sigma_{x_2 x_3} & \cdots & \sigma_{x_2 x_{\mathrm{n}}} \\ \sigma_{x_1 x_3} & \sigma_{x_2 x_3} & \sigma_{x_3}^2 & \cdots & \sigma_{x_3 x_{\mathrm{n}}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sigma_{x_1 x_{\mathrm{n}}} & \sigma_{x_2 x_{\mathrm{n}}} & \sigma_{x_3 x_{\mathrm{n}}} & \cdots & \sigma_{x_{\mathrm{n}}}^2 \end{bmatrix} \tag{2.14}$$

Vector input, vector output The extension of 2.13 to a measurement function with multiple outputs such as $\mathbf{y} = \mathbf{f}(\mathbf{x})$, with the output quantity vector defined as $\mathbf{y} = [y_1, y_2, \ldots, yx_n]$ is quite straight-forward. As $y_i = f_i(\mathbf{x})$, the previous derivations hold for the individual output quantities, and thus, the estimate and the variance for each output quantity are still given by 2.13 and 2.13 respectively. However, in order to fully describe the uncertainty of the measurement result, the covariance matrix $\Sigma_{\mathbf{y}}$ must be considered, as it accounts for the pairwise joint variability of the output quantities.

$$\Sigma_{\mathbf{y}} = \begin{bmatrix} \sigma_{y_1}^2 & \sigma_{y_1 y_2} & \sigma_{y_1 y_3} & \cdots & \sigma_{y_1 y_{\mathrm{n}}} \\ \sigma_{y_1 y_2} & \sigma_{y_2}^2 & \sigma_{y_2 y_3} & \cdots & \sigma_{y_2 y_{\mathrm{n}}} \\ \sigma_{y_1 y_3} & \sigma_{y_2 y_3} & \sigma_{y_3}^2 & \cdots & \sigma_{y_3 y_{\mathrm{n}}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sigma_{y_1 y_{\mathrm{n}}} & \sigma_{y_2 y_{\mathrm{n}}} & \sigma_{y_3 y_{\mathrm{n}}} & \cdots & \sigma_{y_{\mathrm{n}}}^2 \end{bmatrix} \tag{2.15}$$

Following the derivations in the previous section, evaluating 2.9 at $\mathbf{a} = \hat{\mathbf{x}}$, the covariance terms σ_{y_i, y_j} can be computed as [Arr98]:

$$\sigma_{y_i, y_j} = \mathbb{E}\Big[\big(y_i - \mu_{y_i}\big)\big(y_j - \mu_{y_j}\big)\Big] = \mathbb{E}[y_i y_j] - \mathbb{E}[y_i]\,\mathbb{E}[y_j] = \mathbb{E}\left[\left(f_i(\hat{\mathbf{x}}) + \sum_{k=1}^{n} (x_k - \hat{x_k}) \frac{\partial f_i(\hat{\mathbf{x}})}{\partial x_k} \right) \left(f_j(\hat{\mathbf{x}}) + \sum_{l=1}^{n} (x_l - \hat{x_l}) \frac{\partial f_j(\hat{\mathbf{x}})}{\partial x_l} \right) \right] - f_j(\hat{\mathbf{x}})\, f_j(\hat{\mathbf{x}}) = \sum_{k=1}^{n} \sum_{l=1}^{n} \frac{\partial f_i(\hat{\mathbf{x}})}{\partial x_k} \frac{\partial f_j(\hat{\mathbf{x}})}{\partial x_l} \mathbb{E}[(x_i - \hat{x_i})(x_j - \hat{x_j})] \tag{2.16}$$

Therefore, the general uncertainty propagation formula can be used to compute the covariance matrix $\Sigma_{\mathbf{y}}$ describing the uncertainty of the multiple output quantities [JCGM102:2011]:

$$\Sigma_{\mathbf{y}} = \mathbf{J}_{\mathrm{y}} \mathbf{\Sigma}_{\mathbf{x}} \mathbf{J}_{\mathrm{y}}^{\mathrm{T}} \tag{2.17}$$

with

$$\mathbf{J}_{\mathrm{y}} = \frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \frac{\partial y_1}{\partial x_3} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} & \frac{\partial y_2}{\partial x_3} & \cdots & \frac{\partial y_2}{\partial x_n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_m}{\partial x_1} & \frac{\partial y_m}{\partial x_2} & \frac{\partial y_m}{\partial x_3} & \cdots & \frac{\partial y_m}{\partial x_n} \end{bmatrix} \tag{2.18}$$

Note that the error propagation equations above all assume that the measurement function is linear around a small region, validating the truncation of higher order terms.

Computing the sensitivity coefficients In previous explanations it has been assumed that the measurement function is known in a explicit form, *i.e.*, $\mathbf{y} = f(\mathbf{x})$, and that the Jacobian $\mathbf{J}_{\mathbf{y}}$ characterizing the sensitivity of the output quantity vector $\mathbf{y}$ can be obtained simply by computing all the partial derivatives $\frac{\partial y_i}{\partial x_i}$. In cases where such function is defined implicitly by another function, *e.g.*, $\Phi(x, y) = 0$ or $\Phi(x, f(x)) = 0$, the implicit function theorem allows to estimate the sensitivity of the output parameters [Cla98]:

$$\mathbf{J}_{\mathbf{y}} = -\left(\frac{\partial \mathbf{\Phi}}{\partial \mathbf{y}}\right)^{-1} \frac{\partial \mathbf{\Phi}}{\partial \mathbf{x}} \tag{2.19}$$

Furthermore, in cases where even the implicit function is not explicitly known, *i.e.*, when using numerical methods where the result is obtained by minimizing some cost function $C(x, y)$, the Jacobian can still be computed. In these cases, the estimated parameters $\hat{\mathbf{y}}$ correspond to a local minimum of the cost function, *i.e.*, $\frac{\partial C(\mathbf{x},\mathbf{y})}{\partial \mathbf{y}} = 0$, and the implicit function can be defined as $\mathbf{\Phi} = \frac{\partial C}{\partial \mathbf{y}}$, thus using 2.19, the sensitivities are computed as [Cla98]:

$$\mathbf{J}_{\mathbf{y}} = -\left(\frac{\partial^2 C}{\partial \mathbf{y}^2}\right)^{-1} \left(\frac{\partial^2 C}{\partial \mathbf{y} \partial \mathbf{x}}\right)^{\mathrm{T}} \tag{2.20}$$

2.4.4 Monte Carlo Simulations (MCM)

The MCM methodology is much simpler than GUM as it only requires implementing the measurement model in a computer program, and performing repeated calculations of the measured values by drawing the values of the input parameters from their corresponding PDFs. The downside is that these calculations might require large amounts of processing power and time.

The step-by-step procedure for a scalar valued measurand $y = f(\mathbf{x})$ that depends on several input quantities $\mathbf{x} = [x_1, x_2, \ldots, x_N]$ is defined as follows [JCGM101:2008]. Note that the joint PDF corresponding to the input quantities must be known.

1. Select the number of Monte Carlo realizations M. Note that M needs to large compared to $1/(1-p)$ (*e.g.*, 10^4 times greater), where p corresponds to the coverage probability.

2. Generate M vectors $\mathbf{X}$ corresponding to the N input quantities X_i by sampling their corresponding PDFs with appropriate random number generators. Note that the accuracy of the estimated uncertainty increases with this value.

3. Compute the model value Y_j corresponding to each realization j.

4. Compute the estimate of the output quantity $\hat{y}$ and the associated standard uncertainty $u(y)$ by computing the mean and standard deviation of the model values Y_j.

5. Sort the M model values in increasing order to obtain the PDF of Y.

6. Compute the coverage interval with the desired covarage probability p. Note that a coverage factor k is not required for determining the coverage interval.

For a vector valued measurand $\mathbf{y} = f(\mathbf{x})$ the standard uncertainty $u(y)$ would be replaced by a covariance matrix Σ_y and the coverage interval would be replaced by a coverage region, the rest of the procedure remains the same.

2.5 Summary

In this chapter, first the concepts of specular surfaces and deflectometric measurements have been introduced. Afterwards, the measurement procedure corresponding to the ARPG setup has been detailed, and, finally, the basic mathematical tools for error/uncertainty analysis of the measurements have been introduced. The following chapters will make use of these concepts in order to analyze the measurement uncertainty introduced by the camera calibration (Ch. 3) and the systematic errors arising from the screen (Ch. 4).

Chapter 3

Influence of Camera Calibration on Measurement Uncertainty

3.1 Introduction

In deflectometric measurements, camera rays act as probes that measure the 3D location and normal vector of the observed surface by estimating the deflection that light suffers after being reflected on it. The direction of the camera ray originated at a particular camera pixel is defined by a geometric camera model, which depends on a set of parameters that are obtained from a calibration procedure. An inaccurate camera characterization results in inaccuracies in the direction of camera rays, introducing errors that will directly affect the accuracy of the measurements. For a given pixel, the camera ray direction remains constant in every measurement, thus these errors will result in systematic effects. Hence, the relevance of selecting an appropriate model and performing a precise calibration.

As explained in [SRT+11], the main principle of camera model selection is to find a good compromise between the goodness of fit to the data and the complexity of the model. The pinhole model is commonly used in computer vision for cameras with non-wide field of view optics because of its simplicity, and because it has proven to be robust and to perform well in many applications [Zha00, Hei00, Tsa87]. Furthermore, the calibration can be done using software implementations that are easy to use and freely available, *e.g.*, [Bou05, SPS+, Hei, Bra]. Despite this, some researchers have worked with more complex camera models, generally generic models that characterize the optical path for each camera pixel independently [BLS+10, KB06, RSL05]. They claim the achievable accuracy is higher than that obtained with the pinhole model, but these complex models are also more difficult to work with, and that they present their own difficulties (*e.g.*, overfitting, number of images, number of parameters to estimate, available calibration software...).

This chapter analyzes the influence of the camera calibration uncertainty when using the extended pinhole model in deflectometric measurements.

3.2 Background

A camera is a device that collects light into a photosensitive sensor array (typically CCD or CMOS) through a lens, producing an image of the scene observed from a particular vantage point. Thus, a camera performs a projection from the 3D world to a 2D image, and a geometric camera model explains how such mapping from $\mathbb{R}^3$ to $\mathbb{R}^2$ is performed.

This section briefly overviews the pinhole camera model and its calibration, together with the estimated parameters' uncertainty.

3.2.1 Pinhole Model

The pinhole model is one of the simplest geometric camera models. Although it is an idealized mathematical framework, it is used as a base for models commonly used in photogrammetry [HS96, HZ04]. A pinhole camera consists of a box with an (infinitesimal) aperture through which the light rays emitted or reflected by the scene enter the camera. These rays intersect a photosensitive film located at the back of the box, generating an inverted image of the scene.

Fig. 3.1 shows a sketch of a simple pinhole camera. The aperture is called the optical center $\mathbf{O}_c$, also referred to as the projection center or perspective center, and it defines the origin of the camera coordinate frame. The plane defined by the film, called the image plane, is located at a distance f, perpendicular from $\mathbf{O}_c$. The line passing through $\mathbf{O}_c$ perpendicularly to the image plane defines the optical axis $\mathbf{Z}_c$, and the intersection of $\mathbf{Z}_c$ with the image plane defines the principal point $\mathbf{c}$.

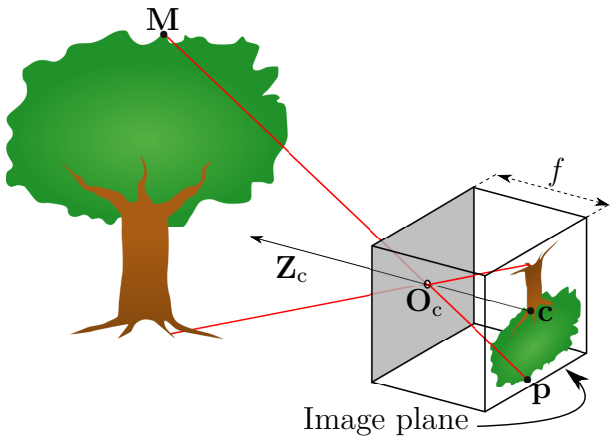

Figure 3.1: Pinhole model: Central projection camera concept[1]

The distance f is defined as the *effective* or *real* focal length, from here on focal length, and it defines the field of view and the scale of the image: the larger f is, the smaller the field of view, and the larger the scale of the imaged scene. The size of a scene object projected onto the image plane also depends on its distance perpendicular to the image plane, with distant objects appearing smaller than closer ones. Thus, the mapping from a point in the scene to its projection on the image plane is defined by the focal distance of the camera, and its distance perpendicular to the image plane.

This relation is described in projection equation shown in 3.1, which becomes obvious observing Fig. 3.2. Let $\mathbf{M} = [M_x, M_y, M_z]$ be a point in the scene, and $\mathbf{p}$ its projection onto the image plane, both given in metric units in the camera frame. Then, by intersecting the sight ray from $\mathbf{O}_c$ to $\mathbf{M}$ with the $\mathbf{Z}_C = f$ plane:

$$\mathbf{p} = \begin{bmatrix} p_x \\ p_y \end{bmatrix} = \begin{bmatrix} f\frac{M_x}{M_z} \\ f\frac{M_y}{M_z} \end{bmatrix} \tag{3.1}$$

[1] Original figure by DrBob, distributed under CC BY-SA 3.0 license

where p_x and p_y are the coordinates (in metric units) corresponding to the projection $\mathbf{p}$.

In practice, in order to avoid dealing with the inverted image, it is usually assumed that the image plane is positioned in front of $\mathbf{O}_c$ at a distance f along $\mathbf{Z}_c$, as shown in Fig. 3.2.

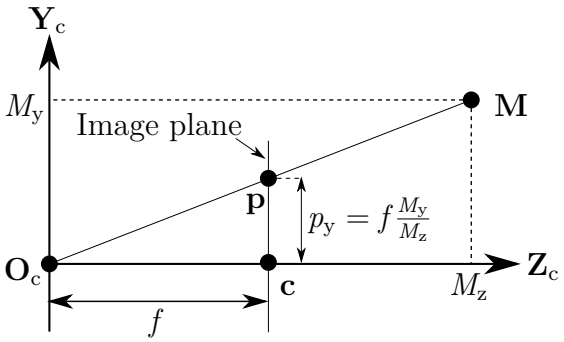

Figure 3.2: Projection of a point through the pinhole

Thus, the direction of a sight ray (camera ray) is defined by the line connecting the optical center with the projected point.

The geometric model of many industrial cameras can be conceptually described following this central projection model: the photosensitive film is replaced by a sensor array, where each of the pixels integrates the radiation incoming from a small scene area, and a lens or a set of lenses are used to focus the light onto the sensor.

Digital images are usually measured in pixels, with the origin of the image frame $\mathbf{O}_i$ located at the upper left corner of the image, as shown in Fig. 3.3. Thus, the projection equation showed in 3.1 needs to be adapted in order to account for the different origin and scaling of the image frame. The camera calibration matrix $\mathbf{K}$, also called intrinsic parameter matrix, describes such mapping in a compact form [HZ04]:

$$\mathbf{K} = \begin{bmatrix} f_u & \gamma & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{bmatrix} \tag{3.2}$$

where f_u and f_v represent the focal length in pixels. This formulation of the focal length accounts for non-square pixels, considering different sensor scales for each direction (s_x and s_y, *i.e.*, the pixel size in mm), where $f_u = f/s_x$ and $f_v = f/s_y$. c_v and c_v are the coordinates of the principal point offset, i.e. the location of $\mathbf{c}$ w.r.t. $\mathbf{O}_i$. γ corresponds to the skew parameter accounting for non orthogonal image axes, and it will not be taken into account as it is generally considered negligible for cameras of average quality [HZ04, SC06, FB18]. In this way, the projection of a point $\mathbf{M}$ onto the image plane in pixel coordinates $\mathbf{m} = [m_u, m_v]^T$ is described by

$$s\tilde{\mathbf{m}} = \mathbf{K}\mathbf{M} \tag{3.3}$$

where $\tilde{\mathbf{m}} = [m_u, m_v, 1]^T$ are the corresponding homogeneous pixel coordinates, and s stands for a scale factor.

Additionally, if the location of $\mathbf{M}$ is given by coordinates in a frame different from the camera frame, which is fairly common, the projective transformation can be extended to include this change between reference frames. The transformation between two reference frames is defined by a rotation ${}_c\mathbf{R}_w$ and a translation ${}_c\mathbf{t}_w^c$, which are called the external or

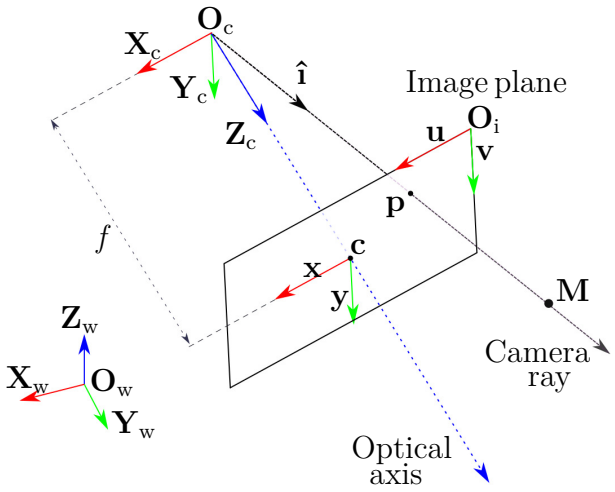

Figure 3.3: Pinhole model: camera geometry

extrinsic parameters of the camera. The transformation from the object reference frame (*e.g.*, the world frame) to the camera reference frame is defined as

$$\mathbf{M}^{\mathrm{c}} = {}_{\mathrm{c}}\mathbf{R}_{\mathrm{w}}\,\mathbf{M}^{\mathrm{w}} + {}_{\mathrm{c}}\mathbf{t}^{\mathrm{c}}_{\mathrm{w}} \tag{3.4}$$

where $\mathbf{M}^{\mathrm{c}}$ and $\mathbf{M}^{\mathrm{w}}$ correspond to the coordinates of point $\mathbf{M}$ in the camera and world frames, respectively.

The full projection can thus be described as a linear transformation in projective space:

$$s\tilde{\mathbf{m}} = \mathbf{K}\;{}_{\mathrm{c}}\mathbf{T}_{\mathrm{w}}\tilde{\mathbf{M}}^{\mathrm{w}} \tag{3.5}$$

where $\tilde{\mathbf{M}}^{\mathrm{w}} = \left[\mathrm{M}^{w}_{\mathrm{x}}, \mathrm{M}^{w}_{\mathrm{y}}, \mathrm{M}^{w}_{\mathrm{z}}, 1\right]^{\mathrm{T}}$ corresponds to the homogeneous world frame coordinates of $\mathbf{M}$, and ${}_{\mathrm{c}}\mathbf{T}_{\mathrm{w}}$ defines the frame transformation

$${}_{\mathrm{c}}\mathbf{T}_{\mathrm{w}} = \begin{bmatrix} {}_{\mathrm{c}}\mathbf{R}_{\mathrm{w}} & {}_{\mathrm{c}}\mathbf{t}^{\mathrm{c}}_{\mathrm{w}} \\ \mathbf{0}_{1\mathrm{x}3} & 1 \end{bmatrix} \tag{3.6}$$

Note that previous formulations allow the projection of $\mathbf{M}$ to $\mathbf{m}$ to be computed, but the inverse, back-projecting $\mathbf{m}$ to obtain the 3D coordinates of $\mathbf{M}$, is obviously not possible without additional information, as such $\mathbb{P}^2$ to $\mathbb{P}^3$ mapping is not unique, and any point laying along the sight ray will be projected to the same point. This ambiguity is implicit in the scale factor s in 3.3 and 3.5.

3.2.2 Extended Pinhole Model

Although the pinhole model can be accurate enough for some applications [SHL95], generally it does not suffice for metrological measuring purposes due to systematic errors caused during the optical system manufacturing. The most important effects for deflectometric measurements are related to geometric distortions. Lens systems are not perfectly manufactured, causing projections to deviate from an ideal pinhole, which results in distorted images. When high accuracy is required, as is the case in deflectometric measurements, the pinhole model is usually extended to consider this non-linear behavior. The extended pinhole model adopted in this thesis will be explained in the following.

The distortion model described by Brown [Bro66, Bro71] is most commonly adopted in computer vision applications involving non wide-angle lenses [Hei00], the reader can refer to [CF98] for an historical overview of the developments in the field. The main idea behind this model is that straight lines in the scene should be projected to straight lines in the image, and that any deviation from this straightness can be attributed to radial and decentering distortion components, as shown in Fig. 3.4:

- Radial distortions are caused by the rotating device used for the manufacturing of the components of the lens system [Han11]. Its effect on a particular pixel consists of a radially symmetric shift that only depends on the distance of the pixel to the principal point Fig. 3.4(a).
- Decentering distortions are commonly associated with misalignment of lenses in the optical system [Fra01], and as a result, the optical axis is not exactly perpendicular to the sensor plane. Its effect on a particular pixel consists of a shift with a tangential and a radial component [WCH92], as shown in Fig. 3.4(b). This distortion is many times referred to simply as tangential distortion and in the rest of the document both terms will be used interchangeably. In non wide-angle lenses, decentering distortion is often considered negligible, however, it will also be considered for the sake of completeness.

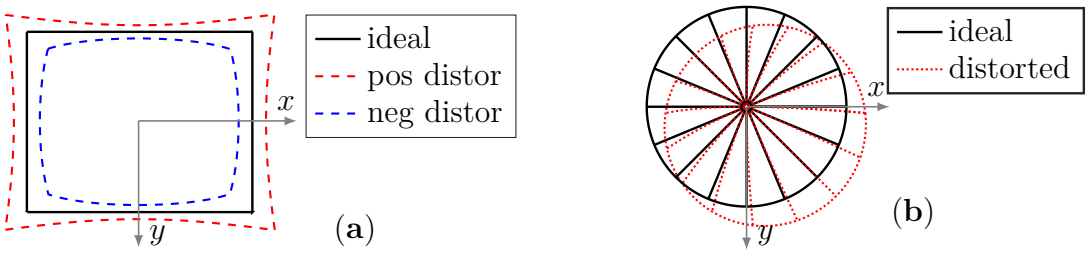

Figure 3.4: Lens Distortion effects: (a) Radial distortion; (b) Decentering distortion.

Both, radial and decentering distortions, are modeled on the normalized image plane, *i.e.*, a virtual plane, parallel to the image plane, located at $Z_c = 1\,\text{mm}$. This plane has no physical interpretation, but it leads to a convenient mathematical formulation. In this way, let $\mathbf{m}$ be the pixel coordinates corresponding to the projection of a scene point $\mathbf{M}$ through an ideal pinhole, and $\mathbf{m}^u = [u_u, v_u, 1]^T$ its mapping to the normalized image plane. Due to the lens distortions, $\mathbf{M}$ will actually be observed at a shifted pixel $\mathbf{m}'$, with its mapping to the normalized image plane defined as $\mathbf{m}^d = [u_d, v_d, 1]^T$. The projection through an extended pinhole model accounting for radial $\boldsymbol{\delta}_r$ and tangential $\boldsymbol{\delta}_t$ distortion effects is commonly modeled as [Hei00]:

$$\tilde{\mathbf{m}}' = \mathbf{K}\mathbf{m}^d = \mathbf{K}\left(\mathbf{m}^u + \boldsymbol{\delta}_r(u_u, v_u) + \boldsymbol{\delta}_t(u_u, v_u)\right) \tag{3.7}$$

with

$$\boldsymbol{\delta}_r(u_u, v_u) = \begin{bmatrix} \delta_{ru} \\ \delta_{rv} \\ 0 \end{bmatrix} = \begin{bmatrix} u_u(k_1 r^2 + k_2 r^4 + ...) \\ v_u(k_1 r^2 + k_2 r^4 + ...) \\ 0 \end{bmatrix} \tag{3.8}$$

and

$$\boldsymbol{\delta}_{\mathrm{t}}(u_{\mathrm{u}}, v_{\mathrm{u}}) = \begin{bmatrix} \delta_{\mathrm{tu}} \\ \delta_{\mathrm{tv}} \\ 0 \end{bmatrix} = \begin{bmatrix} 2p_1 u_{\mathrm{u}} v_{\mathrm{u}} + p_2(r^2 + 2u_{\mathrm{u}}^2) \\ p_1(r^2 + 2v_{\mathrm{u}}^2) + 2p_2 u_{\mathrm{u}} v_{\mathrm{u}} \\ 0 \end{bmatrix} \tag{3.9}$$

where k_1, k_2, ... correspond to the radial distortion coefficients; p_1, p_2 correspond to the tangential distortion coefficients, and $r^2 = u_{\mathrm{u}}^2 + v_{\mathrm{u}}^2$.

Many applications require a corrected image resembling a distortion free projection, which can be directly achieved by using the previous formulation: the intensity value corresponding to a pixel in the undistorted image is obtained by interpolating the values at the closest neighboring pixels in the distorted space. In 3D applications, including deflectometry, it is often desirable to obtain the undistorted coordinates $\mathbf{m}^{\mathrm{u}}$ in order to compute the sight ray direction $\hat{\mathbf{i}}$ corresponding to a certain pixel. The above expressions do not provide a direct analytical method for the actual corrections to be computed. Equations 3.8-3.9 model the distortion implicitly, as the amount of shift depends non-linearly on the (unobservable) coordinates $\mathbf{m}^{\mathrm{u}}$, and a closed form solution for the inverse mapping can not be directly derived.

Several works have presented direct distortion correction methods based on approximations to such inverse function [WM94, HS97, Hei00, MW04, DL16], which allows a fast computation of corrections, but often results in an increase in the number of employed parameters. Another approach is to perform the correction numerically using iterative methods. Starting from an initial guess (usually $\mathbf{m}^{\mathrm{d}}$), the distorted position is iteratively refined. [Mel94] proposed an approach in which the correction was performed in 2 iterations, such that:

$$\mathbf{m}^{\mathrm{u}} = \mathbf{m}^{\mathrm{d}} - \boldsymbol{\delta}\Big(\mathbf{m}^{\mathrm{d}} - \boldsymbol{\delta}\Big(\mathbf{m}^{\mathrm{d}}\Big)\Big) \tag{3.10}$$

where $\boldsymbol{\delta} = \delta_{\mathrm{r}} + \delta_{\mathrm{t}}$ corresponds to the combined distortion effects.

Heikkila pointed out [HS97] that although 2 iterations might be enough for some applications, more runs should be performed when accuracies better than *e.g.*, 0.1 pixels are required. In our experiments an average of seven iterations converged to the true value up to numerical computation accuracy. This approach is slower than using the approximated inverse functions, but the speed requirement is not critical in deflectometric measurements. As each pixel's sight ray direction will remain constant once the setup has been calibrated, the computations can be done off-line and afterwards reused in every measurement. For this reason, this approach has been chosen using an implementation based on [Bou05], as detailed in Alg. 1.

Note that the extended pinhole model described in this section is an approximation that considers the most meaningful geometrical properties of the distortions, but additional terms would be needed for a faithful model. More physically accurate models can be found in the literature, *e.g.*, [WCH92, Fra01, Han11, TvGMM17]. In practice, it is commonly accepted that with conventional non wide-angle lenses image aberrations are dominated by a radial component. Several researchers argue that more complex models can lead to numerical instabilities during calibration, which can decrease the accuracy of the other estimated camera parameters without improving the fitting [Tsa87, WM94, HS96, Zha00]. Besides, several works in deflectometric applications are working on generic camera models which define each ray independently, *e.g.*, [BLS+10, KB06, RSL05]. These models are typically more appropriate for lenses not common in industrial inspection cameras, *e.g.*, wide-angle lenses, and involve a large

Algorithm 1 Undistort pixel: $\mathbf{m}^{\mathrm{u}} = \boldsymbol{\delta}^{-1}\left(\mathbf{m}^{\mathrm{d}}, \boldsymbol{\theta}_{\mathrm{dis}}\right)$

begin
  $\mathbf{m}^{\mathrm{u}} \leftarrow \mathbf{m}^{\mathrm{d}}$
  $k_{\mathrm{r}}, \mathbf{k}_{\mathrm{t}} \leftarrow$ get_distortion($\mathbf{m}^{\mathrm{u}}$,$\boldsymbol{\theta}_{\mathrm{dis}}$)
  repeat
    $\mathbf{m}^{\mathrm{u}} \leftarrow (\mathbf{m}^{\mathrm{d}} - \mathbf{k}_{\mathrm{t}})/k_{\mathrm{r}}$
    $k_{\mathrm{r}}, \mathbf{k}_{\mathrm{t}} \leftarrow$ get_distortion($\mathbf{m}^{\mathrm{u}}$,$\boldsymbol{\theta}_{\mathrm{dis}}$)
    $\hat{\mathbf{m}}^{\mathrm{d}} \leftarrow (\mathbf{m}^{\mathrm{u}} k_{\mathrm{r}} + \mathbf{k}_{\mathrm{t}})$
    $C \leftarrow$ squared_euclidean_distance($\hat{\mathbf{m}}^{\mathrm{d}}, \mathbf{m}^{\mathrm{d}}$)
  until $C < \epsilon$
  return $\mathbf{m}^{\mathrm{u}}$
end
function get_distortion($\mathbf{m}^{\mathrm{u}}$,$\boldsymbol{\theta}_{\mathrm{dis}}$)
  $r^2 \leftarrow u_{\mathrm{u}}^2 + v_{\mathrm{u}}^2$
  $k_{\mathrm{r}} \leftarrow 1 + k_1 r^2 + k_2 r^4 + ...$
  $\mathbf{k}_{\mathrm{t}} \leftarrow \begin{bmatrix} 2p_1 u_{\mathrm{u}} v_{\mathrm{u}} + p_2(r^2 + 2u_{\mathrm{u}}^2) \\ p_1(r^2 + 2v_{\mathrm{u}}^2) + 2p_2 u_{\mathrm{u}} v_{\mathrm{u}} \end{bmatrix}$
  return $k_{\mathrm{r}}, \mathbf{k}_{\mathrm{t}}$
end function

amount of parameters, which goes in the opposite direction of what has been said in [Tsa87, WM94, HS96, Zha00]. In order to shed some light on these contradictions, this chapter considers the systematic errors introduced by the extended pinhole model by analyzing the calibration uncertainty.

3.2.3 Pinhole Model Calibration

In deflectometric measurements the extended pinhole model described in the previous sections is used to define the camera rays' directions. To do so, the camera needs to be calibrated, which means that the intrinsic camera parameters $\boldsymbol{\theta}_{\mathrm{cam}}$ need to be estimated in advance, including both the pinhole parameters $\boldsymbol{\theta}_{\mathrm{pin}} = [f_{\mathrm{u}}, f_{\mathrm{v}}, c_{\mathrm{u}}, c_{\mathrm{v}}]$ and the distortion coefficients $\boldsymbol{\theta}_{\mathrm{dis}} = [k_1, k_2, p_1, p_2]$.

The camera calibration techniques used in metrology are generally based on the observation of a calibration target containing visual landmarks or feature points whose location is precisely known in the object frame. Typically a planar target containing a series of feature points is observed from multiple viewpoints, *e.g.*, chessboard patterns [Zha00], circle patterns [Hei00], or sinusoidal patterns [SFA11]. The acquired images are processed to extract the location of these feature points, obtaining a series of correspondences $\mathbf{m}_{ij} \leftrightarrow \mathbf{M}_j^{\mathrm{w}}$, where $\mathbf{m}_{ij}$ corresponds to the measured pixel location for the jth feature point in the i th image and $\mathbf{M}_j^{\mathrm{w}}$ represents the known local 3D coordinates of such feature point. Estimating the unknown model parameters that best explain those correspondences defines an inverse problem whose solution is found using iterative methods such as Gradient descent or Levenberg–Marquardt.

Recall from 3.5 that the image coordinates $\mathbf{m}$ corresponding to the projection of a point in 3D space $\mathbf{M}^{\mathrm{w}}$ through a extended pinhole camera depend on the intrinsic parameters $\boldsymbol{\theta}_{\mathrm{cam}} = [\boldsymbol{\theta}_{\mathrm{pin}}, \boldsymbol{\theta}_{\mathrm{dis}}]$, but also on the camera pose ${}_{\mathrm{c}}\mathbf{T}_{\mathrm{w}}$. Thus, the cal-

ibration procedure also needs to estimate the unknown extrinsic parameters for each viewpoint $\boldsymbol{\theta}_{\text{ext}} = \left[{}_{\text{c}_1}\mathbf{T}_{\text{w}}, {}_{\text{c}_2}\mathbf{T}_{\text{w}}, \ldots, {}_{\text{c}_n}\mathbf{T}_{\text{w}}\right]$. In this way, let the estimated coordinates $\hat{\mathbf{m}}_{ij} = f\left(\boldsymbol{\theta}_{\text{cam}}, {}_{\text{c}_i}\mathbf{T}_{\text{w}}, \mathbf{M}_j^{\text{w}}\right)$ be the projection of $\mathbf{M}_j^{\text{w}}$ according to an extended pinhole camera with a relative pose ${}_{\text{c}_i}\mathbf{T}_{\text{w}}$, as described by 3.5 and 3.7 . The goal is to find $\boldsymbol{\theta}_{\text{cam}}, \boldsymbol{\theta}_{\text{ext}}$ that minimize the geometric distance between $\hat{\mathbf{m}}_{ij}$ and $\mathbf{m}_{ij}$, which is done by non-linear minimization of the residuals $\boldsymbol{\epsilon}_{ij} = \mathbf{m}_{ij} - \hat{\mathbf{m}}(\boldsymbol{\theta}_{\text{cam}}, \boldsymbol{\theta}_{\text{ext}_i}, \mathbf{M}_j)$:

$$\hat{\boldsymbol{\theta}}_{\text{cam}}, \hat{\boldsymbol{\theta}}_{\text{ext}} = \underset{\boldsymbol{\theta}_{\text{cam}}, \boldsymbol{\theta}_{\text{ext}}}{\arg\min} \sum_{i=1}^{n} \sum_{j=1}^{m} ||\boldsymbol{\epsilon}_{ij}||^2 \tag{3.11}$$

where n and m correspond respectively to the number of viewpoints and feature points.

As non-linear iterative methods rely on a good initialization to avoid getting caught in a local minimum, initial values are obtained through a linear estimation based on the conventional pinhole model that ignores lens distortions [Zha00, Hei00].

Note that the above solution is optimal under the assumption that the iterative algorithm is initialized in the convex region of the absolute minimum and that all the uncertainty can be attributed to the measurement noise, which implies that the camera model compensates for all systematic errors and that all feature points $\mathbf{M}_j^{\text{w}}$ are assumed error free or, at least, with an error neglectable compared to measurement errors. A common assumption is that the errors in the extracted feature locations are independent and obey a zero-mean Gaussian distribution $\mathcal{N}(\mathbf{0}, \Sigma_{\mathbf{m}})$. This error model is not strictly justified, but it is acceptable in the absence of outliers in the measured data. Under these conditions, the standard uncertainty matrix of the estimated parameters $\Sigma_{\boldsymbol{\theta}}$ can be approximated by propagating the residuals [HZ04] :

$$\Sigma_{\boldsymbol{\theta}} = \left[\mathbf{J}\Sigma_{\mathbf{m}}^{-1}\mathbf{J}^{\text{T}}\right]^{+} = \left[\begin{array}{c|c} \Sigma_{\boldsymbol{\theta}_{\text{cam}}} & \Sigma_{\boldsymbol{\theta}_{\text{cam}}, \boldsymbol{\theta}_{\text{ext}}} \\ \hline \Sigma_{\boldsymbol{\theta}_{\text{ext}}, \boldsymbol{\theta}_{\text{cam}}} & \Sigma_{\boldsymbol{\theta}_{\text{ext}}} \end{array}\right] \tag{3.12}$$

where the Jacobian matrix $\mathbf{J}$ describes the sensitivity of every estimated coordinate $\hat{\mathbf{m}}_{ij}$ with respect to each estimated parameter:

$$\mathbf{J} = \frac{\partial \hat{\mathbf{m}}}{\partial[\boldsymbol{\theta}_{\text{cam}} \mid \boldsymbol{\theta}_{\text{ext}}]} \tag{3.13}$$

and $\Sigma_{\mathbf{m}}$ corresponds to the covariance matrix characterizing the uncertainty in the measurements

$$\Sigma_{\mathbf{m}} = \begin{bmatrix} \Sigma_{\mathbf{m}_{11}} & 0 & \ldots & 0 & \ldots & 0 \\ 0 & \Sigma_{\mathbf{m}_{12}} & \ldots & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & \Sigma_{\mathbf{m}_{1m}} & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & \ldots & \Sigma_{\mathbf{m}_{nm}} \end{bmatrix} \tag{3.14}$$

where $\Sigma_{\mathbf{m}_{ij}}$ corresponds to the 2x2 covariance matrix of $\mathbf{m}_{ij}$. Since $\Sigma_{\mathbf{m}_{ij}}$ is usually not known, it is common to make the assumption that in the absence of outliers measurement errors are isotropic and identically distributed, thus $\Sigma_{\mathbf{m}_{ij}} = \sigma_{\mathbf{m}}^2 \mathbf{I}_2$, where $\sigma_{\mathbf{m}}$ corresponds to the rms re-projection error.

For the ray direction computations, only the intrinsic camera parameters $\boldsymbol{\theta}_{\text{cam}}$ are used, thus their uncertainty is described by the covariance matrix $\Sigma_{\boldsymbol{\theta}_{\text{cam}}}$, which can be obtained by truncating $\Sigma_{\boldsymbol{\theta}}$ such that

$$\Sigma_{\boldsymbol{\theta}_{\text{cam}}} = \begin{bmatrix} \sigma^2_{f_u} & \sigma_{f_u f_v} & \sigma_{f_u c_u} & \sigma_{f_u c_v} & \sigma_{f_u k_1} & \sigma_{f_u k_2} & \sigma_{f_u p_1} & \sigma_{f_u p_2} \\ \sigma_{f_v f_u} & \sigma^2_{f_v} & \sigma_{f_v c_u} & \sigma_{f_v c_v} & \sigma_{f_v k_1} & \sigma_{f_v k_2} & \sigma_{f_v p_1} & \sigma_{f_v p_2} \\ \sigma_{c_u f_u} & \sigma_{c_u f_v} & \sigma^2_{c_u} & \sigma_{c_u c_v} & \sigma_{c_u k_1} & \sigma_{c_u k_2} & \sigma_{c_u p_1} & \sigma_{c_u p_2} \\ \sigma_{c_v f_u} & \sigma_{c_v f_v} & \sigma_{c_v c_u} & \sigma^2_{c_v} & \sigma_{c_v k_1} & \sigma_{c_v k_2} & \sigma_{c_v p_1} & \sigma_{c_v p_2} \\ \sigma_{k_1 f_u} & \sigma_{k_1 f_v} & \sigma_{k_1 c_u} & \sigma_{k_1 c_v} & \sigma^2_{k_1} & \sigma_{k_1 k_2} & \sigma_{k_1 p_1} & \sigma_{k_1 p_2} \\ \sigma_{k_2 f_u} & \sigma_{k_2 f_v} & \sigma_{k_2 c_u} & \sigma_{k_2 c_v} & \sigma_{k_2 k_1} & \sigma^2_{k_2} & \sigma_{k_2 p_1} & \sigma_{k_2 p_2} \\ \sigma_{p_1 f_u} & \sigma_{p_1 f_v} & \sigma_{p_1 c_u} & \sigma_{p_1 c_v} & \sigma_{p_1 k_1} & \sigma_{p_1 k_2} & \sigma^2_{p_1} & \sigma_{p_1 p_2} \\ \sigma_{p_2 f_u} & \sigma_{p_2 f_v} & \sigma_{p_2 c_u} & \sigma_{p_2 c_v} & \sigma_{p_2 k_1} & \sigma_{p_2 k_2} & \sigma_{p_2 p_1} & \sigma^2_{p_2} \end{bmatrix}. \tag{3.15}$$

3.3 Camera Ray Uncertainty

The camera ray direction corresponding to a camera pixel $\mathbf{m}$ is represented by a unit vector $\hat{\mathbf{i}}$, which is computed directly from the camera parameters $\boldsymbol{\theta}_{\text{cam}}$. These parameters are estimated up to some uncertainty ($\Sigma_{\boldsymbol{\theta}_{\text{cam}}}$) during camera calibration, but their true values are not known. This section deals with how such an imperfect knowledge of $\boldsymbol{\theta}_{\text{cam}}$ affects the direction of a camera ray, with the uncertainty in $\hat{\mathbf{i}}$ described by a 3x3 covariance matrix $\Sigma_{\hat{\mathbf{i}}}$ representing the spread of the reasonable values that could be attributed to $\hat{\mathbf{i}}$ due to $\Sigma_{\boldsymbol{\theta}_{\text{cam}}}$.

As shown in Fig. 3.5, $\hat{\mathbf{i}}$ is computed in three steps: First, the (distorted) image coordinates $\mathbf{m}$ (in pixels) are mapped onto the normalized image plane ($Z_c = 1\,\text{mm}$) using $\boldsymbol{\theta}_{\text{pin}}$, obtaining the projection $\mathbf{m}^{\text{d}}$, which will be known up to an uncertainty $\Sigma_{\mathbf{m}^{\text{d}}}$ due to $\Sigma_{\boldsymbol{\theta}_{\text{pin}}}$. Afterwards, the lens distortion effects are corrected using $\boldsymbol{\theta}_{\text{pin}}$, resulting in the undistorted projection $\mathbf{m}^{\text{u}}$ coordinates, with an uncertainty $\Sigma_{\mathbf{m}^{\text{u}}}$ arising from $\Sigma_{\mathbf{m}^{\text{d}}}$ and $\Sigma_{\boldsymbol{\theta}_{\text{dis}}}$. Finally, the ray direction can be computed, with its uncertainty $\Sigma_{\hat{\mathbf{i}}}$ defined by $\Sigma_{\mathbf{m}^{\text{u}}}$.

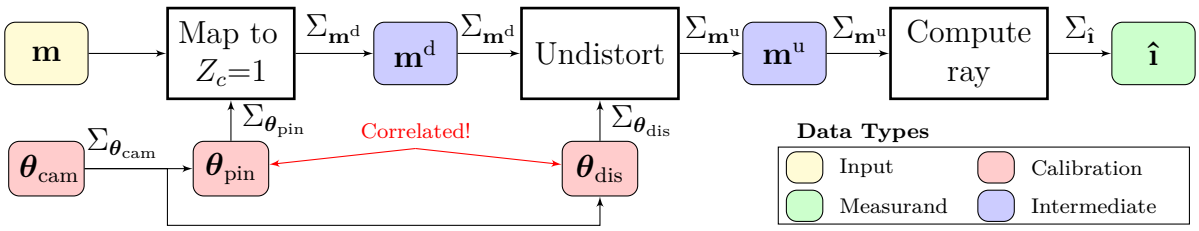

Figure 3.5: Propagation of camera parameter uncertainty through the steps of the camera ray computation.

The remainder of the section details these steps, showing how the camera uncertainties are propagated through each of them. All the computations will be performed in the camera frame, and the steps are the same for all pixels in the sensor. Thus, a generic pixel $\mathbf{m}$ will be considered, and the frame superscript will be omitted in all variables.

3.3.1 Projection to Normalized Image Plane

In order to compute the direction of the camera ray $\hat{\mathbf{i}}$ corresponding to pixel $\mathbf{m}$, first the distortions need to be removed. For this, the pixel coordinates must be projected to

the normalized image plane in the camera frame ($\mathbf{z}_\mathrm{c} = 1\mathrm{mm}$), which is done following the pinhole model explained in Sec. 3.2.2. Therefore, the computation is defined by the following equation:

$$\mathbf{m}^\mathrm{d} = \mathbf{K}^{-1}\tilde{\mathbf{m}} = \begin{bmatrix} -c_\mathrm{u}f_\mathrm{u}^{-1} + f_\mathrm{u}^{-1}m_\mathrm{u} \\ -c_\mathrm{v}f_\mathrm{v}^{-1} + f_\mathrm{v}^{-1}m_\mathrm{v} \\ 1 \end{bmatrix} \tag{3.16}$$

where $\tilde{\mathbf{m}} = [m_\mathrm{u}, m_\mathrm{v}, 1]^\mathrm{T}$ corresponds to the homogeneous coordinates of pixel $\mathbf{m}$, $\mathbf{m}^\mathrm{d} = \left[m_\mathrm{u}^d, m_\mathrm{v}^d, 1\right]^\mathrm{T}$ represents the distorted coordinates of its projection to the normalized image plane, and $\mathbf{K}^{-1}$ is the inverse of the calibration matrix defined as:

$$\mathbf{K}^{-1} = \begin{bmatrix} f_\mathrm{u}^{-1} & 0 & -c_\mathrm{u}f_\mathrm{u}^{-1} \\ 0 & f_\mathrm{v}^{-1} & -c_\mathrm{v}f_\mathrm{v}^{-1} \\ 0 & 0 & 1 \end{bmatrix} \tag{3.17}$$

As $\tilde{\mathbf{m}}$ does not have any associated uncertainty, the uncertainty of $\mathbf{m}^\mathrm{d}$, *i.e.*, $\Sigma_{\mathbf{m}^\mathrm{d}}$, comes from $\Sigma_{\boldsymbol{\theta}_\mathrm{pin}}$. The third coordinate is fixed, thus the uncertainty only affects the first two, and therefore only the 2x2 sub-matrix of $\Sigma_{\mathbf{m}^\mathrm{d}}$ contains relevant information:

$$\Sigma_{\mathbf{m}^\mathrm{d}} = \begin{bmatrix} \sigma^2_{m_\mathrm{u}^d} & \sigma_{m_\mathrm{u}^d m_\mathrm{v}^d} & 0 \\ \sigma_{m_\mathrm{v}^d m_\mathrm{u}^d} & \sigma^2_{m_\mathrm{v}^d} & 0 \\ 0 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} \sigma^2_{m_\mathrm{u}^d} & \sigma_{m_\mathrm{u}^d m_\mathrm{v}^d} \\ \sigma_{m_\mathrm{v}^d m_\mathrm{u}^d} & \sigma^2_{m_\mathrm{v}^d} \end{bmatrix} \tag{3.18}$$

In order to compute $\Sigma_{\mathbf{m}^\mathrm{d}}$, the input quantity vector $\mathbf{x}_{\mathbf{m}^\mathrm{d}} = [f_\mathrm{u}, f_\mathrm{v}, c_\mathrm{u}, c_\mathrm{v}]^\mathrm{T}$, and the output quantity vector $\mathbf{y}_{\mathbf{m}^\mathrm{d}} = \left[m_\mathrm{u}^d, m_\mathrm{v}^d\right]^\mathrm{T}$ are defined, and $\Sigma_{\boldsymbol{\theta}_\mathrm{pin}}$ propagated using a first order Taylor approximation (2.17):

$$\Sigma_{\mathbf{m}^\mathrm{d}} = \mathbf{J}_{\mathbf{m}^\mathrm{d}} \Sigma_{\boldsymbol{\theta}_\mathrm{pin}} \mathbf{J}_{\mathbf{m}^\mathrm{d}}^\mathrm{T} \tag{3.19}$$

where the sensitivity matrix of the output quantities, $\mathbf{J}_{\mathbf{m}^\mathrm{d}} = \frac{\partial \mathbf{y}_{\mathbf{m}^\mathrm{d}}}{\partial \mathbf{x}_{\mathbf{m}^\mathrm{d}}}$, is defined as:

$$\mathbf{J}_{\mathbf{m}^\mathrm{d}} = \begin{bmatrix} (c_\mathrm{u} - m_\mathrm{u})\, f_\mathrm{u}^{-2} & 0 & -f_\mathrm{u}^{-1} & 0 \\ 0 & (c_\mathrm{v} - m_\mathrm{v})\, f_\mathrm{v}^{-2} & 0 & -f_\mathrm{v}^{-1} \end{bmatrix} \tag{3.20}$$

From there, substituting and rearranging terms, one ends up with the following expressions for the variance of the u and v components:

$$\sigma^2_{m_\mathrm{u}^d} = \frac{1}{f_\mathrm{u}^2}\left(d_\mathrm{u}^2\sigma^2_{f_\mathrm{u}} + \sigma^2_{c_\mathrm{u}} - 2d_\mathrm{u}\sigma_{f_\mathrm{u}c_\mathrm{u}}\right) \tag{3.21}$$

and

$$\sigma^2_{m_\mathrm{v}^d} = \frac{1}{f_\mathrm{v}^2}\left(d_\mathrm{v}^2\sigma^2_{f_\mathrm{v}} + \sigma^2_{c_\mathrm{v}} - 2d_\mathrm{v}\sigma_{f_\mathrm{v}c_\mathrm{v}}\right) \tag{3.22}$$

with $d_\mathrm{u} = \frac{c_\mathrm{u} - m_\mathrm{u}}{f_\mathrm{u}}$ and $d_\mathrm{v} = \frac{c_\mathrm{v} - m_\mathrm{v}}{f_\mathrm{v}}$ being the distances from $\mathbf{m}$ to $\mathbf{c}$, respectively in horizontal and vertical direction, projected onto the normalized plane (in mm).

Similarly, the covariances are given by 3.23:

$$\sigma_{m_\mathrm{u}^d m_\mathrm{v}^d} = \sigma_{m_\mathrm{v}^d m_\mathrm{u}^d} = \frac{1}{f_\mathrm{u}f_\mathrm{v}}(\sigma_{c_\mathrm{v}c_\mathrm{u}} + d_\mathrm{u}d_\mathrm{v}\sigma_{f_\mathrm{v}f_\mathrm{u}} - d_\mathrm{u}\sigma_{f_\mathrm{u}c_\mathrm{v}} - d_\mathrm{v}\sigma_{f_\mathrm{v}c_\mathrm{u}}) \tag{3.23}$$

Note that the computation of projected point $\mathbf{m}^{\mathrm{d}}$ only takes a subset of the camera parameters $\boldsymbol{\theta}_{\mathrm{cam}}$ as an input, *i.e.*, the pinhole parameters $\boldsymbol{\theta}_{\mathrm{pin}}$. The other subset, the distortion parameters $\boldsymbol{\theta}_{\mathrm{dis}}$, are used in the distortion removal step. As this second step also takes $\mathbf{m}^{\mathrm{d}}$ as an input, if $\boldsymbol{\theta}_{\mathrm{pin}}$ and $\boldsymbol{\theta}_{\mathrm{dis}}$ are correlated, so will $\mathbf{m}^{\mathrm{d}}$ and $\boldsymbol{\theta}_{\mathrm{dis}}$. In order to take into account these covariances, $\Sigma_{\mathbf{m}^{\mathrm{d}}}$ is augmented to form the extended covariance matrix $\Sigma^{*}_{\mathbf{m}^{\mathrm{d}}}$, as shown in 3.24:

$$\Sigma^{*}_{\mathbf{m}^{\mathrm{d}}} = \begin{bmatrix} \Sigma_{\mathbf{m}^{\mathrm{d}}} & \Sigma_{\mathbf{m}^{\mathrm{d}},\boldsymbol{\theta}_{\mathrm{dis}}} \\ \Sigma_{\boldsymbol{\theta}_{\mathrm{dis}},\mathbf{m}^{\mathrm{d}}} & \Sigma_{\boldsymbol{\theta}_{\mathrm{dis}}} \end{bmatrix} = \left[\begin{array}{cc|cccc} \sigma^2_{m^d_{\mathrm{u}}} & \sigma_{m^d_{\mathrm{u}} m^d_{\mathrm{v}}} & \sigma_{m^d_{\mathrm{u}} k_1} & \sigma_{m^d_{\mathrm{u}} k_2} & \sigma_{m^d_{\mathrm{u}} p_1} & \sigma_{m^d_{\mathrm{u}} p_2} \\ \sigma_{m^d_{\mathrm{v}} m^d_{\mathrm{u}}} & \sigma^2_{m^d_{\mathrm{v}}} & \sigma_{m^d_{\mathrm{v}} k_1} & \sigma_{m^d_{\mathrm{v}} k_2} & \sigma_{m^d_{\mathrm{v}} p_1} & \sigma_{m^d_{\mathrm{v}} p_2} \\ \hline \sigma_{k_1 m^d_{\mathrm{u}}} & \sigma_{k_1 m^d_{\mathrm{v}}} & \sigma^2_{k_1} & \sigma_{k_1 k_2} & \sigma_{k_1 p_1} & \sigma_{k_1 p_2} \\ \sigma_{k_2 m^d_{\mathrm{u}}} & \sigma_{k_2 m^d_{\mathrm{v}}} & \sigma_{k_2 k_1} & \sigma^2_{k_2} & \sigma_{k_2 p_1} & \sigma_{k_2 p_2} \\ \sigma_{p_1 m^d_{\mathrm{u}}} & \sigma_{p_1 m^d_{\mathrm{v}}} & \sigma_{p_1 k_1} & \sigma_{p_1 k_2} & \sigma^2_{p_1} & \sigma_{p_1 p_2} \\ \sigma_{p_2 m^d_{\mathrm{u}}} & \sigma_{p_2 m^d_{\mathrm{v}}} & \sigma_{p_2 k_1} & \sigma_{p_2 k_2} & \sigma_{p_2 p_1} & \sigma^2_{p_2} \end{array}\right] \tag{3.24}$$

where $\Sigma_{\mathbf{m}^{\mathrm{d}}}$ and $\Sigma_{\boldsymbol{\theta}_{\mathrm{dis}}}$ are the covariance matrices representing the uncertainty of the projected point $\mathbf{m}^{\mathrm{d}}$ and the distortion parameters, whereas $\Sigma_{\mathbf{m}^{\mathrm{d}},\boldsymbol{\theta}_{\mathrm{dis}}}$ contains the covariances between them.

The extended covariance matrix $\Sigma^{*}_{\mathbf{m}^{\mathrm{d}}}$, is computed using the extended input quantity vector $\mathbf{x}^{*}_{\mathbf{m}^{\mathrm{d}}}$ and the extended output quantity vector $\mathbf{y}^{*}_{\mathbf{m}^{\mathrm{d}}}$, which are defined as $\mathbf{x}^{*}_{\mathbf{m}^{\mathrm{d}}} = [f_{\mathrm{u}}, f_{\mathrm{v}}, c_{\mathrm{u}}, c_{\mathrm{v}}, k_1, k_2, p_1, p_2]^{\mathrm{T}}$ and $\mathbf{y}^{*}_{\mathbf{m}^{\mathrm{d}}} = \left[m^d_{\mathrm{u}}, m^d_{\mathrm{v}}, k_1, k_2, p_1, p_2\right]^{\mathrm{T}}$, and the uncertainties are propagated using a first order Taylor expansion:

$$\Sigma^{*}_{\mathbf{m}^{\mathrm{d}}} = \mathbf{J}^{*}_{\mathbf{m}^{\mathrm{d}}} \Sigma_{\boldsymbol{\theta}_{\mathrm{cam}}} \mathbf{J}^{*\mathrm{T}}_{\mathbf{m}^{\mathrm{d}}} \tag{3.25}$$

where the $\mathbf{J}^{*}_{\mathbf{m}^{\mathrm{d}}}$ is a simple extension of the sensitivity matrix defined in 3.20:

$$\mathbf{J}^{*}_{\mathbf{m}^{\mathrm{d}}} = \begin{bmatrix} \mathbf{J}_{\mathbf{m}^{\mathrm{d}}} & \mathbf{0}_{2\mathrm{x}4} \\ \mathbf{0}_{4\mathrm{x}4} & \mathbf{I}_{4\mathrm{x}4} \end{bmatrix} \tag{3.26}$$

With $\mathbf{0}$ and $\mathbf{I}$ respectively representing zero vectors and identity matrices.

3.3.2 Lens Distortion Removal

Once pixel coordinates have been mapped to the normalized image plane, the next step in the measurement process is to correct the lens distortions using the iterative algorithm shown in Alg. 1:

$$\mathbf{m}^{\mathrm{u}} = \boldsymbol{\delta}^{-1}\left(\mathbf{m}^{\mathrm{d}}, \boldsymbol{\theta}_{\mathrm{dis}}\right) = \begin{bmatrix} u_{\mathrm{u}} \\ v_{\mathrm{u}} \\ 1 \end{bmatrix} \tag{3.27}$$

where $\mathbf{m}^{\mathrm{u}}$ stands for the undistorted coordinates of $\mathbf{m}$ in the normalized image plane, $\boldsymbol{\delta}^{-1}$ represents the iterative correction algorithm, and the input parameters $\boldsymbol{\theta}_{\mathrm{dis}}$ and $\mathbf{m}^{\mathrm{d}}$ respectively correspond to the distortion parameters and the distorted coordinates of $\mathbf{m}$.

The correction step thus involves uncertainties in both $\mathbf{m}^{\mathrm{d}}$ and $\boldsymbol{\theta}_{\mathrm{dis}}$, which can be correlated, hence the extended covariance matrix (3.24) computed in the previous step.

$$\Sigma_{\mathbf{m}^{\mathrm{d}}_{\mathrm{ext}}} = \begin{bmatrix} \Sigma_{\mathbf{m}^{\mathrm{d}}} & \Sigma_{\mathbf{m}^{\mathrm{d}},\boldsymbol{\theta}_{\mathrm{dis}}} \\ \Sigma_{\boldsymbol{\theta}_{\mathrm{dis}},\mathbf{m}^{\mathrm{d}}} & \Sigma_{\boldsymbol{\theta}_{\mathrm{dis}}} \end{bmatrix} \tag{3.28}$$

In order to compute the uncertainty of the undistorted $\mathbf{m}^{\mathrm{u}}$, the input quantity vector $\mathbf{x}_{\mathbf{m}^{\mathrm{u}}} = [u_{\mathrm{d}}, v_{\mathrm{d}}, k_1, k_2, p_1, p_2]^{\mathrm{T}}$, and the output quantity vector $\mathbf{y}_{\mathbf{m}^{\mathrm{u}}} = [u_{\mathrm{u}}, v_{\mathrm{u}}]^{\mathrm{T}}$are defined, and $\Sigma^{*}_{\mathbf{m}^{\mathrm{d}}}$ is propagated using 2.17:

$$\Sigma_{\mathbf{m}^{\mathrm{u}}} = \mathbf{J}_{\mathbf{m}^{\mathrm{u}}} \Sigma^{*}_{\mathbf{m}^{\mathrm{d}}} \mathbf{J}^{\mathrm{T}}_{\mathbf{m}^{\mathrm{u}}} \tag{3.29}$$

where the Jacobian $\mathbf{J}_{\mathbf{m}^{\mathrm{u}}} = \frac{\partial \mathbf{y}_{\mathbf{m}^{\mathrm{u}}}}{\partial \mathbf{x}_{\mathbf{m}^{\mathrm{u}}}}$ describes the sensitivity of the output parameters, and $\Sigma_{\mathbf{m}^{\mathrm{u}}}$ characterizes the uncertainty in the undistorted point:

$$\Sigma_{\mathbf{m}^{\mathrm{u}}} = \begin{bmatrix} \sigma^2_{m^u_{\mathrm{u}}} & \sigma_{m^d_{\mathrm{u}} m^d_{\mathrm{v}}} & 0 \\ \sigma_{m^u_{\mathrm{v}} m^u_{\mathrm{u}}} & \sigma^2_{m^u_{\mathrm{v}}} & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{3.30}$$

The fact that the undistortion operation is performed using a numerical method does not allow the derivatives to be directly obtained as defined in 2.18, so the Jacobian is computed using the implicit approach described in 2.19. Using the distortion model in 3.7, the implicit equation $\mathbf{\Phi}_{\mathrm{u}}$ is defined as:

$$\mathbf{\Phi_u} = \mathbf{m}^{\mathrm{d}} - \mathbf{m}^{\mathrm{u}} k_{\mathrm{r}}(u_{\mathrm{u}}, v_{\mathrm{u}}) - \boldsymbol{\delta}_{\mathrm{t}}(u_{\mathrm{u}}, v_{\mathrm{u}}) = \mathbf{0} \tag{3.31}$$

with $k_{\mathrm{r}} = 1 + \boldsymbol{\delta}_{\mathrm{r}}/\mathbf{m}^{\mathrm{u}}$, and $\boldsymbol{\delta}_{\mathrm{r}}$, and $\boldsymbol{\delta}_{\mathrm{t}}$ are the radial and tangential distortion coefficients.

Thus, the jacobian can be computed as:

$$\mathbf{J}_{\mathbf{m}^{\mathrm{u}}} = -\left(\frac{\partial \mathbf{\Phi}_{\mathrm{u}}}{\partial \mathbf{y}_{\mathbf{m}^{\mathrm{u}}}}\right)^{-1} \frac{\partial \mathbf{\Phi}_{\mathrm{u}}}{\partial \mathbf{x}_{\mathbf{m}^{\mathrm{u}}}} \tag{3.32}$$

Note that following the chain rule $\Sigma_{\mathbf{m}^{\mathrm{u}}}$ can be computed directly as:

$$\Sigma_{\mathbf{m}^{\mathrm{u}}} = \mathbf{J}_{\mathbf{m}^{\mathrm{u}}} \mathbf{J}^{*}_{\mathbf{m}^{\mathrm{d}}} \Sigma_{\boldsymbol{\theta}_{\mathrm{cam}}} \mathbf{J}^{*\mathrm{T}}_{\mathbf{m}^{\mathrm{d}}} \mathbf{J}^{\mathrm{T}}_{\mathbf{m}^{\mathrm{u}}} \tag{3.33}$$

3.3.3 Camera Ray Direction Computation

The camera ray direction $\hat{\mathbf{i}}$ is defined as the 3D unit vector corresponding to the direction of the line connecting the optical center $\mathbf{O}_{\mathrm{c}}$ and the pixel center back-projected to the normalized image plane $\mathbf{m}^{\mathrm{u}}$, which in camera coordinates reduces to normalizing $\mathbf{m}^{\mathrm{u}}$:

$$\hat{\mathbf{i}} = \frac{\mathbf{m}^{\mathrm{u}} - \mathbf{O}_{\mathrm{c}}}{||\mathbf{m}^{\mathrm{u}} - \mathbf{O}_{\mathrm{c}}||} = \frac{\mathbf{m}^{\mathrm{u}}}{||\mathbf{m}^{\mathrm{u}}||} \tag{3.34}$$

Thus, the uncertainty of $\hat{\mathbf{i}}$ only depends on the uncertainty of $\mathbf{m}^{\mathrm{u}}$, and is characterized by the 3x3 uncertainty matrix $\Sigma_{\hat{\mathbf{i}}}$, which can be obtained by propagating $\Sigma_{\mathbf{m}^{\mathrm{u}}}$ through Taylor approximation using 2.17, with the input quantity vector $\mathbf{x}_{\hat{\mathbf{i}}} = [u_{\mathrm{u}}, v_{\mathrm{u}}]^{\mathrm{T}}$, and the output quantity vector $\mathbf{y}_{\hat{\mathbf{i}}} = \hat{\mathbf{i}} = [i_x, i_y, i_z]^{\mathrm{T}}$:

$$\mathbf{\Sigma}_{\hat{\mathbf{i}}} = \mathbf{J}_{\hat{\mathbf{i}}} \Sigma_{\mathbf{m}^{\mathrm{u}}} \mathbf{J}_{\hat{\mathbf{i}}}^{\mathrm{T}} \tag{3.35}$$

where $\Sigma_{\mathbf{m}^{\mathrm{u}}}$ corresponds to the uncertainty in the back-projected pixel, and $\mathbf{J}_{\hat{\mathbf{i}}}$ represents the Jacobian matrix describing the sensitivity of the system:

$$\mathbf{J}_{\hat{\mathbf{i}}} = \frac{\partial \mathbf{y}_{\hat{\mathbf{i}}}}{\partial \mathbf{x}_{\hat{\mathbf{i}}}} = \begin{bmatrix} \frac{\partial \hat{i}_x}{\partial u_{\mathrm{u}}} & \frac{\partial \hat{i}_x}{\partial v_{\mathrm{u}}} \\ \frac{\partial \hat{i}_y}{\partial u_{\mathrm{u}}} & \frac{\partial \hat{i}_y}{\partial v_{\mathrm{u}}} \\ \frac{\partial \hat{i}_z}{\partial u_{\mathrm{u}}} & \frac{\partial \hat{i}_z}{\partial v_{\mathrm{u}}} \end{bmatrix} \tag{3.36}$$

Note that following the chain rule, the propagation of the uncertainty from the camera parameters to the camera ray direction is reduced to the following equation:

$$\Sigma_{\hat{\mathbf{i}}} = \mathbf{J}_{\hat{\mathbf{i}}}\mathbf{J}_{\mathbf{m}^{\mathrm{u}}}\mathbf{J}^{*}_{\mathbf{m}^{\mathrm{d}}}\Sigma_{\boldsymbol{\theta}_{\mathrm{cam}}}\mathbf{J}^{*\mathrm{T}}_{\mathbf{m}^{\mathrm{d}}}\mathbf{J}^{\mathrm{T}}_{\mathbf{m}^{\mathrm{u}}}\mathbf{J}_{\hat{\mathbf{i}}}^{\mathrm{T}} \tag{3.37}$$

Although $\Sigma_{\hat{\mathbf{i}}}$ fully describes the uncertainty of a camera ray, interpreting such 3x3 matrix is not very intuitive. One approach could be to consider the coverage region, given that the shape of the output distribution is known. By definition, $\hat{\mathbf{i}}$ is a unit vector, thus in the case of small uncertainties, rather than a volume, the coverage region will describe a surface perpendicular to the ray $\hat{\mathbf{i}}$.

A more intuitive measure can be obtained by considering the uncertainty in the angle between $\hat{\mathbf{i}}$ and the optical axis $\mathbf{Z}_{\mathrm{c}}$, as it allows us to interpret the uncertainty as a change in direction, and is represented by a single value. Such angle can be computed as:

$$\alpha_{\hat{\mathbf{i}}} = \mathrm{acos}\left(\hat{\mathbf{i}} \cdot \hat{\mathbf{Z}}_{c}\right) \tag{3.38}$$

From there, the angular uncertainty of the camera ray can be obtained by propagation through Taylor approximation:

$$\Sigma_{\alpha_{\mathrm{i}}} = \mathbf{J}_{\alpha_{\hat{\mathbf{i}}}}\Sigma_{\hat{\mathbf{i}}}\mathbf{J}^{\mathrm{T}}_{\alpha_{\hat{\mathbf{i}}}} \tag{3.39}$$

with $\mathbf{J}_{\alpha_{\hat{\mathbf{i}}}}$

$$\mathbf{J}_{\alpha_{\hat{\mathbf{i}}}} = \begin{bmatrix} \frac{\partial \alpha_{\hat{\mathbf{i}}}}{\partial \hat{i}_x} \\ \frac{\partial \alpha_{\hat{\mathbf{i}}}}{\partial \hat{i}_y} \\ \frac{\partial \alpha_{\hat{\mathbf{i}}}}{\partial \hat{i}_z} \end{bmatrix} \tag{3.40}$$

Note that the cosine is a clearly non-linear function with high local curvatures, thus, if accuracy were to be required, higher order terms should be considered in the Taylor approximation. In this thesis, this measure will only be used to interpret the results, and it will not be used in any additional computation, restricting to $\Sigma_{\hat{\mathbf{i}}}$ for propagating the uncertainties to further stages of the measurements. For this reason, only the first order approximation of $\alpha_{\hat{\mathbf{i}}}$ is considered.

3.4 Effects on Measurements

In this section, an approach to analyze the influence of the camera ray uncertainty $\Sigma_{\hat{\mathbf{i}}}$ on the measurements is described, considering all system parameters except for the camera parameters to be exactly known. Note that these measurements are not only dependent on the camera parameters, as in the previous section, but also by the whole setup geometry. The following figure shows a simplified sketch of the propagation of the camera uncertainty to the measurements.

3.4.1 Normal Vector Uncertainty

The measurement of a normal $\hat{\mathbf{n}} = [n_{\mathrm{x}}, n_{\mathrm{y}}, n_{\mathrm{z}}]^{\mathrm{T}}$ observed in a single image pixel $\mathbf{m} = [u, v]^{\mathrm{T}}$ is defined as:

$$\hat{\mathbf{n}} = \frac{\hat{\mathbf{r}} - \hat{\mathbf{i}}}{||\hat{\mathbf{r}} - \hat{\mathbf{i}}||} \tag{3.41}$$

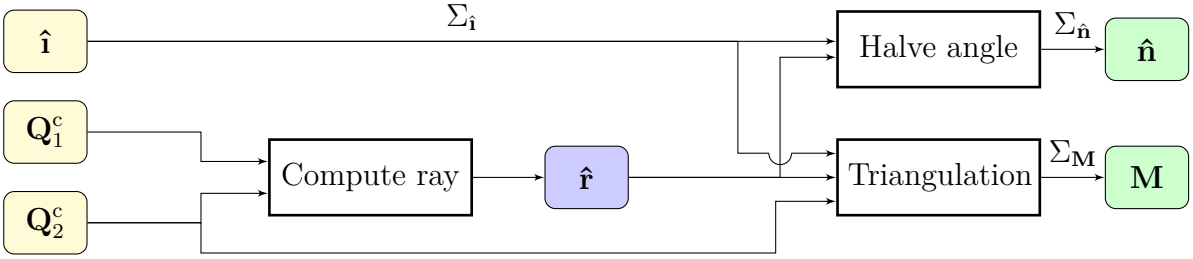

Figure 3.6: Camera ray uncertainty: effects on the measurement

where $\hat{\mathbf{r}} = [r_x, r_y, r_z]^T$ and $\hat{\mathbf{i}}$ are the unit vectors corresponding to the camera and reflected rays, as shown in Fig. 2.8.

In this first approach, $\hat{\mathbf{r}}$ is assumed to be perfectly known, and thus the uncertainty comes from the computation of the camera ray $\hat{\mathbf{i}}$, such that the previously computed $\Sigma_{\hat{\mathbf{i}}}$ can be propagated using 2.18:

$$\mathbf{\Sigma}_{\hat{\mathbf{n}}} = \mathbf{J}_{\hat{\mathbf{n}}} \Sigma_{\hat{\mathbf{i}}} \mathbf{J}_{\hat{\mathbf{n}}}^T \tag{3.42}$$

where

$$\mathbf{J}_{\hat{\mathbf{n}}} = \begin{bmatrix} \frac{\partial n_x}{\partial \hat{i}_x} & \frac{\partial n_x}{\partial \hat{i}_y} & \frac{\partial n_x}{\partial \hat{i}_z} \\ \frac{\partial n_y}{\partial \hat{i}_x} & \frac{\partial n_y}{\partial \hat{i}_y} & \frac{\partial n_y}{\partial \hat{i}_z} \\ \frac{\partial n_z}{\partial \hat{i}_x} & \frac{\partial n_z}{\partial \hat{i}_y} & \frac{\partial n_z}{\partial \hat{i}_z} \end{bmatrix} \tag{3.43}$$

In order to obtain a interpretable representation of this uncertainty matrix, the angular uncertainty is considered, as has been previously done for the camera ray:

$$\alpha_{\hat{\mathbf{n}}} = \text{acos}(\hat{\mathbf{n}} \cdot \hat{\mathbf{i}}) \tag{3.44}$$

From there, the angular uncertainty of the surface normal can be obtained by propagation through Taylor approximation:

$$\Sigma_{\alpha_{\hat{\mathbf{n}}}} = \mathbf{J}_{\alpha_{\hat{\mathbf{n}}}} \Sigma_{\hat{\mathbf{n}}} \mathbf{J}_{\alpha_{\hat{\mathbf{n}}}}^T \tag{3.45}$$

3.4.2 Triangulated Point Uncertainty

Similarly, the triangulated surface point $\mathbf{M} = [M_x, M_y, M_z]^T$ observed in $\mathbf{m}$ is given by the intersection of these two rays, which is computed as the point that minimizes the squared sum of distances to the two rays (d^2):

$$\mathbf{M} = \arg\min_{\mathbf{M}} d^2 = d_r^2 + d_i^2 \tag{3.46}$$

where d_r is the distance from $\mathbf{M}$ to $\mathbf{r}$ and d_i^2 the distance from $\mathbf{M}$ to $\mathbf{i}$, that can be found respectively as:

$$d_r = (\mathbf{M} - \mathbf{Q}_1)^T \cdot (\mathbf{M} - \mathbf{Q}_1) - \left[(\mathbf{M} - \mathbf{Q}_1)^T \cdot \hat{\mathbf{r}}\right] \tag{3.47}$$

and

$$d_i = (\mathbf{M} - \mathbf{O}_c)^T \cdot (\mathbf{M} - \mathbf{O}_c) - \left[(\mathbf{M} - \mathbf{O}_c)^T \cdot \hat{\mathbf{i}}\right] \tag{3.48}$$

Differentiating d^2 with respect to $\mathbf{M}$, and rearranging the terms an equation of the form $\mathbf{SM} = \mathbf{C}$ is obtained, then a closed form solution is given by:

$$\mathbf{M} = \boldsymbol{S}^+ \mathbf{C} \tag{3.49}$$

where $\mathbf{S}$ is defined as:

$$\mathbf{S} = \left(\hat{\mathbf{r}} \cdot \hat{\mathbf{r}}^{\mathrm{T}} - \mathbf{I}\right) + \left(\hat{\mathbf{i}} \cdot \hat{\mathbf{i}}^{\mathrm{T}} - \mathbf{I}\right) \tag{3.50}$$

and $\mathbf{C}$ is defined as:

$$\mathbf{C} = \left(\hat{\mathbf{r}} \cdot \hat{\mathbf{r}}^{\mathrm{T}} - \mathbf{I}\right) \cdot \mathbf{Q}_1 + \left(\hat{\mathbf{i}} \cdot \hat{\mathbf{i}}^{\mathrm{T}} - \mathbf{I}\right) \cdot \mathbf{O}_{\mathrm{c}} \tag{3.51}$$

As only the uncertainty of $\hat{\mathbf{i}}$ is considered, the uncertainty in the triangulated point can be obtained through the following first order Taylor approximation:

$$\mathbf{\Sigma_M} = \mathbf{J_M} \mathbf{\Sigma_{\hat{i}}} \mathbf{J_M}^{\mathrm{T}} \tag{3.52}$$

where

$$\mathbf{J_M} = \begin{bmatrix} \frac{\partial M_{\mathrm{x}}}{\partial i_{\mathrm{x}}} & \frac{\partial M_{\mathrm{x}}}{\partial i_{\mathrm{y}}} & \frac{\partial M_{\mathrm{x}}}{\partial i_{\mathrm{z}}} \\ \frac{\partial M_{\mathrm{y}}}{\partial i_{\mathrm{x}}} & \frac{\partial M_{\mathrm{y}}}{\partial i_{\mathrm{y}}} & \frac{\partial M_{\mathrm{y}}}{\partial i_{\mathrm{z}}} \\ \frac{\partial M_{\mathrm{z}}}{\partial i_{\mathrm{x}}} & \frac{\partial M_{\mathrm{z}}}{\partial i_{\mathrm{y}}} & \frac{\partial M_{\mathrm{z}}}{\partial i_{\mathrm{z}}} \end{bmatrix} \tag{3.53}$$

In order to obtain a more interpretable measure of this uncertainty, it might be interesting to give it in a reference frame different from the camera frame, *e.g.*, the object or world frame. Assuming that the corresponding transformation is known without uncertainty, the uncertainty $\mathbf{\Sigma_M}$ can be propagated to $\Sigma_{\mathbf{M}^{\mathrm{w}}}$

$$\Sigma_{\mathbf{M}^{\mathrm{w}}} = \mathbf{J}_{\mathbf{M}^{\mathrm{w}}} \Sigma_{\mathbf{M}} \mathbf{J}_{\mathbf{M}^{\mathrm{w}}}{}^{\mathrm{T}} \tag{3.54}$$

with the sensitivity matrix $\mathbf{J}_{\mathbf{M}^{\mathrm{w}}}$ obtained as:

$$\mathbf{J}_{\mathbf{M}^{\mathrm{w}}} = \frac{\partial \mathbf{M}^{\mathrm{w}}}{\partial \mathbf{M}^{\mathrm{c}}} = {}_{\mathrm{w}}\mathbf{R}_{\mathrm{c}} \tag{3.55}$$

where the frame transformation is defined as:

$$\mathbf{M}^{\mathrm{w}} = {}_{\mathrm{w}}\mathbf{R}_{\mathrm{c}}\, \mathbf{M}^{\mathrm{c}} + {}_{\mathrm{w}}\mathbf{t}_{\mathrm{c}}^{\mathrm{w}} \tag{3.56}$$

3.5 Experiments

In the previous two sections, the propagation of the uncertainty from an extended pinhole camera calibration through the different stages of a deflectometric measurement process has been modeled following the GUM (Guide for the expression of Uncertainty in Measurements) approach. In this section, such models are used to evaluate a real deflectometric measurement system.

For this purpose different experiments are performed:

- In Sec. 3.5.1 a 5 Mpixel grayscale industrial camera with a f=25mm lens is calibrated using a precision calibration plate with a checkerboard pattern and the Matlab calibration toolbox [Bou05]. The uncertainty associated to the internal parameters is obtained by propagating the residual errors of the calibration. These values will be used in the next experiments.

- In Sec. 3.5.2 the influence of the camera calibration uncertainty on the uncertainty of the estimated camera ray direction is tested by propagating the uncertainties from the first experiment.

- In Sec. 3.5.3 the influence of the camera ray direction uncertainty on the final measurements is analyzed, considering the effects on both, the normal vector and the triangulated point.

Note that in all the models developed in the previous sections, a generic pixel was considered, the uncertainty of variable x being defined as (σ_x). In these experiments the models will be applied to every pixel in a sensor of size WxH, where W and H are the sensor width and height in pixels. Thus the uncertainty σ_x depends on the sensor pixel $[u, v]$, resulting in a new variable $\mathbf{u}_x(u, v)$ that represents the standard uncertainty in x over the sensor. The mean and standard deviation of $\mathbf{u}_x(u, v)$ will be defined as $\mu_{\mathbf{u}_x}$ and $\sigma_{\mathbf{u}_x}$. In addition, several of the considered variables consist of multiple components, *e.g.*, the direction vector $\hat{\mathbf{i}}$ has the x, y, z components or the position vector $\mathbf{m}^{\mathrm{d}}$ has the u, v components. The combined effects of these components are difficult to interpret from the covariance matrix, so other equivalent variables will be defined that are easier to analyze, *e.g.*, the angle of $\hat{\mathbf{i}}$ w.r.t. Z_c direction.

Finally, the GUM approach was applied to several non-linear operations, with assumptions on their linearity in small regions. MCM (Monte Carlo) simulations are used to validate these assumptions. In order to do so, the propagation of some particular pixels is analyzed in depth to check the validity of the first order Taylor approximations, as well as the assumptions on the shape of the output distributions characterizing the uncertainty. For this purpose, the measurements corresponding to each selected pixel are repeatedly computed using camera parameters that have been randomly perturbed, by sampling a multivariate normal distribution with a mean, and covariance equal to the $\boldsymbol{\theta}_{\mathrm{cam}}$ and $\Sigma_{\boldsymbol{\theta}_{\mathrm{cam}}}$ estimated in Sec. 3.5.1. In our tests 1.0e8 repetitions were enough to ensure the covariance matrix of the MCM data resembled the one describing the uncertainty of the camera parameters.

3.5.1 Camera Characterization

In order to calibrate the camera, a series of 30 images of the calibration pattern are acquired carefully selecting the viewpoints, such that the field of view and the measurement volume are fully covered with different viewing angles and plate tilts. These images are used to calibrate the camera using three variations of the distortion model: The first one ($R2\&T$) considers two radial and two tangential coefficients, the second one ($R2$) considers two radial coefficients, and the third one ($R1$), only a single radial coefficient. The resulting parameters and the corresponding 3σ coverage intervals are shown in Table 3.1:

Model	Reprojection error			Focal Length		Principal point		Distortion coefficients			
	er_{u}(pix)	er_{v}(pix)		f_{u}(pix)	f_{v}(pix)	c_{u}(pix)	c_{v}(pix)	k_1	k_2	p_1	p_2
R2&T	0.040	0.038	$\boldsymbol{\theta}_{\mathrm{cam}}$	5130.8	5131.0	1268.0	1011.5	-0.3795	0.169	2.1e-4	5.9e-4
			CI	2.9	2.9	2.3	2.3	0.0014	0.013	0.8e-4	0.8e-4
R2	0.042	0.040	$\boldsymbol{\theta}_{\mathrm{cam}}$	5125.8	5125.9	1276.4	1013.2	-0.3775	0.168	-	-
			CI	3.0	2.9	2.2	2.1	0.0015	0.014	-	-
R1	0.049	0.044	$\boldsymbol{\theta}_{\mathrm{cam}}$	5122.8	5123.3	1276.3	1014.1	-0.3618	-	-	-
			CI	3.3	3.3	2.4	2.4	0.0008	-	-	-

Table 3.1: Camera calibration results: Re-projection errors, estimated parameters ($\boldsymbol{\theta}_{\mathrm{cam}}$), and 3σ coverage intervals (CI) for each distortion model.

As can be seen in Table 3.1, the re-projection errors and the parameter's uncertainties are similar for the first and second model, with the first resulting in slightly better re-projection errors and smaller uncertainties in the focal and distortion coefficients, but higher uncertainties in the principal point. This indicates that both models are capable of fitting the distortions up to a similar extent, which makes sense considering that for the first model the distortion is clearly dominated by the radial distortion component, as the tangential coefficients p_1 and p_2 are close to zero. This can also be seen in Fig. 3.7, where the contour lines show that the contribution of the tangential distortion component is 50 times smaller than that of the radial one. The third model gives higher re-projection errors and uncertainties suggesting that the simple model is unable to fit the lens distortions as well as the previous two.

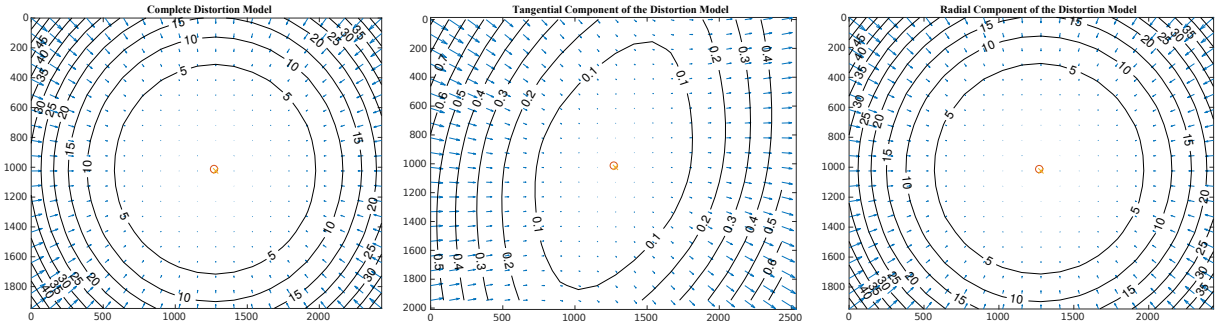

Figure 3.7: Lens distortion induced displacements for each image pixel: The cross represents the image center, the circle indicates the principal point, and each arrow illustrates the effective displacement of the corresponding pixel, which is also represented by the isoline contours. From left to right: (a) Impact of complete distortion model (radial + tangential). (b) Impact of tangential distortion model. (c) Impact of radial distortion model.

In all three cases the re-projection errors were approximately normally distributed, as shown in Fig. 3.8.

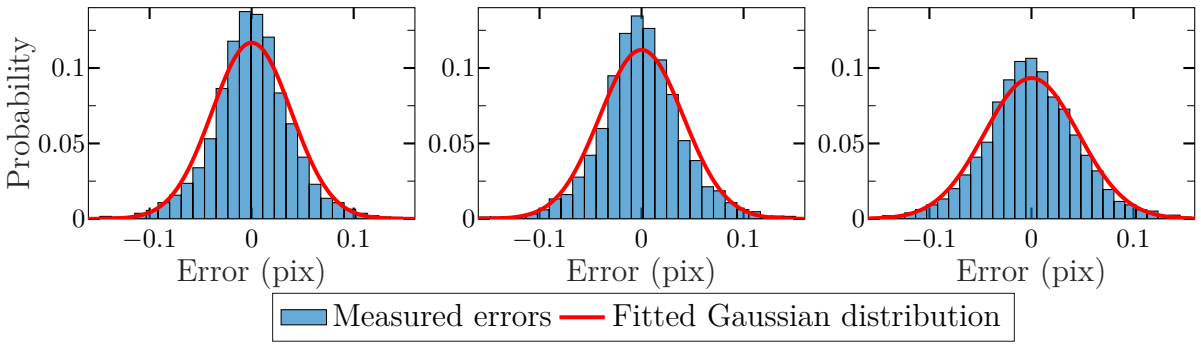

Figure 3.8: Re-projection error distribution for calibration using model (a) R2&T, (b) R2, (c) R1.

Figures 3.9 and 3.10 show the covariance and correlation matrices of the camera parameters for the three models. The former are the uncertainty matrices that will be used in the remaining of the experiments, the latter show correlations between camera parameters. In Fig. 3.10 (a) it can be seen that in model (R2&T), the principal point and the tangential parameters are highly correlated. This can affect the minimization step of the

calibration, which might be the reason for the higher uncertainty of the principal point compared to that obtained in (R2).

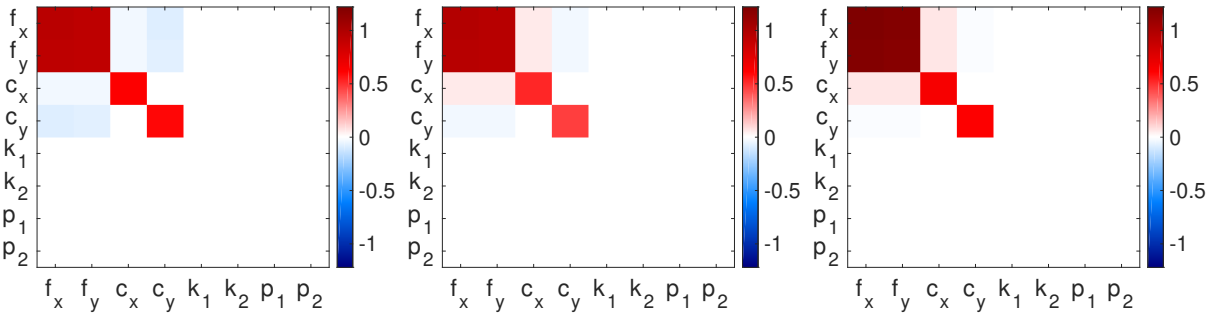

Figure 3.9: Covariance matrix $\boldsymbol{\Sigma}_{\theta_{\text{cam}}}$ corresponding to camera parameters obtained using distortion model (a) R2&T, (b) R2, (c) R1.

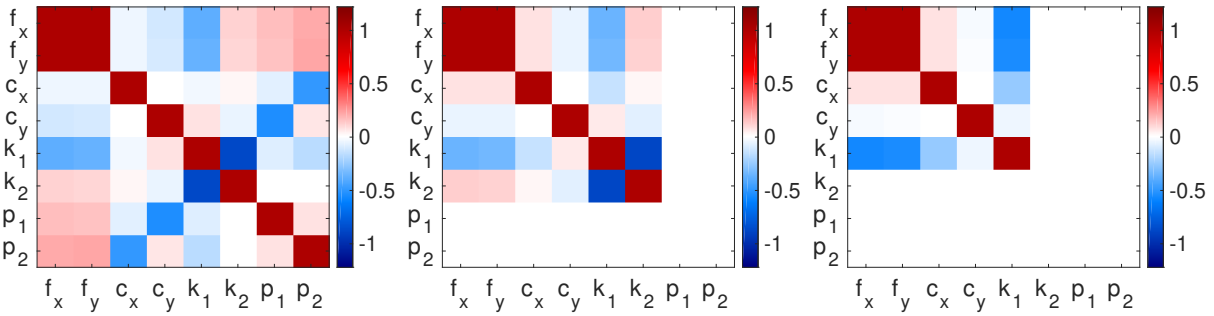

Figure 3.10: Correlation matrix $\boldsymbol{\rho}_{\theta_{cam}}$ corresponding to camera parameters obtained using distortion model (a) R2&T, (b) R2, (c) R1.

3.5.2 Uncertainty in the Camera Ray

The goal of this experiment is to demonstrate how the propagation model defined in Sec. 3.3 is used to estimate the uncertainties along the camera ray computation procedure, by applying it to an industrial camera as the one employed in the setup.

An analysis is made of how the uncertainty in the camera parameters is propagated from the calibration to the camera ray direction corresponding to each back-projected pixel. For this purpose, the camera parameters and uncertainties estimated in the previous experiment (Sec. 3.5.1) are taken, and the propagated uncertainties, the resulting coverage intervals, and their variation afterward is analyzed. The obtained results will be used later in the experiments of Sec. 3.5.3, where the influence of the camera ray on the measurement will be considered.

Note that the camera has been calibrated with three different variations of the distortion model (see Table 3.1), using the same images and the same future points, and thus three sets of input parameters $\{\boldsymbol{\theta}_{\text{cam}}, \Sigma_{\boldsymbol{\theta}_{\text{cam}}}\}$ are obtained. The uncertainties in the camera ray directions resulting from such sets are compared in order to select the distortion model that will be employed in the rest of this work. In this way, the most appropriate distortion model for the employed camera/lens configuration will be selected.

As the uncertainties in $\mathbf{m}^{\text{d}}$ and $\mathbf{m}^{\text{u}}$ define the uncertainty in the camera ray, the contributions of the pinhole and distortion parameters can be analyzed in an easier way

in the normalized image plane ($Z_c = 1\,\text{mm}$). The normalized image plane is particularly suitable for this analysis, as the uncertainties before and after the lens distortion correction represent a variation of a pixel back-projected to the normalized plane measured in the same units (mm). This will be done in Sec. 3.5.2.1. Finally, the camera ray uncertainty is analyzed in Sec. 3.5.2.2.

3.5.2.1 Uncertainty in the Normalized Plane: Effects of Camera Parameters

In this test, the propagation models defined in Sec. 3.3 are used to estimate the uncertainties of the pixel back-projected to the normalized image plane. Taking the same camera from Sec. 3.5.1 as an example, the contributions of the uncertainty of pinhole and distortion parameters are analyzed. For this purpose, the variables $\mathbf{u}_{\mathbf{m}^{\text{d}}}$ and $\mathbf{u}_{\mathbf{m}^{\text{u}}}$ are considered, which represent the standard uncertainties of the sensor pixels back-projected to the normalized image plane, respectively before and after the lens distortions have been removed. Table 3.2 gives a summary of the results, indicating average value and the standard deviation of these uncertainties for each axis (U,V).

Model	Variable	U axis (µm)		V axis (µm)	
		avg.	std	avg.	std
$R2\&T$	$\mathbf{u}_{\mathbf{m}^{\text{d}}}$	0.155	0.002	0.152	0.002
	$\mathbf{u}_{\mathbf{m}^{\text{u}}}$	0.158	0.005	0.154	0.004
$R2$	$\mathbf{u}_{\mathbf{m}^{\text{d}}}$	0.143	0.003	0.135	0.002
	$\mathbf{u}_{\mathbf{m}^{\text{u}}}$	0.147	0.006	0.138	0.004
$R1$	$\mathbf{u}_{\mathbf{m}^{\text{d}}}$	0.159	0.003	0.155	0.002
	$\mathbf{u}_{\mathbf{m}^{\text{u}}}$	0.164	0.007	0.159	0.004

Table 3.2: Uncertainty in the normalized image plane: standard uncertainty

From Table 3.2 it appears clear that in this case, the order of magnitude of the standard uncertainties is similar for all three distortion models, with values below 0.2µm over the whole sensor. As a reference, note that the pixel size projected to the normalized image plane is approximately 2 µm, thus these uncertainties represent around 10% of a pixel size projected to the normalized image plane. The distortion model with 2 radial coefficients (R2) results in the lowest uncertainties over the whole sensor, both before and after correcting the distortions. In addition, it can be seen that $R2\&T$ distortion model performs slightly better than $R1$. In any case, it should be noted that the differences in the average standard uncertainty between all three models are around 0.01-0.02 µm, which barely represents a 1% of the pixel size.

It is interesting to represent graphically the spatial distribution of these uncertainties. These curves are shown in Fig. 3.11, corresponding to (a) the values of $\sigma_{m_{\text{u}}^d}$ and $\sigma_{m_{\text{u}}^u}$ over the pixels in a row, and (b) $\sigma_{m_{\text{v}}^d}$ and $\sigma_{m_{\text{v}}^u}$ for the pixels in a column.

With the aim of understanding the joint effects of the uncertainties in each axis, their corresponding standard coverage regions can be considered. Following the procedure explained in Sec. 2.4.1, the uncertainty matrices are used to obtain the standard coverage ellipses for every back-projected pixel, computing the corresponding halve-length of the major (r_1) and minor (r_2) axes for each of them, before and after the correction of the

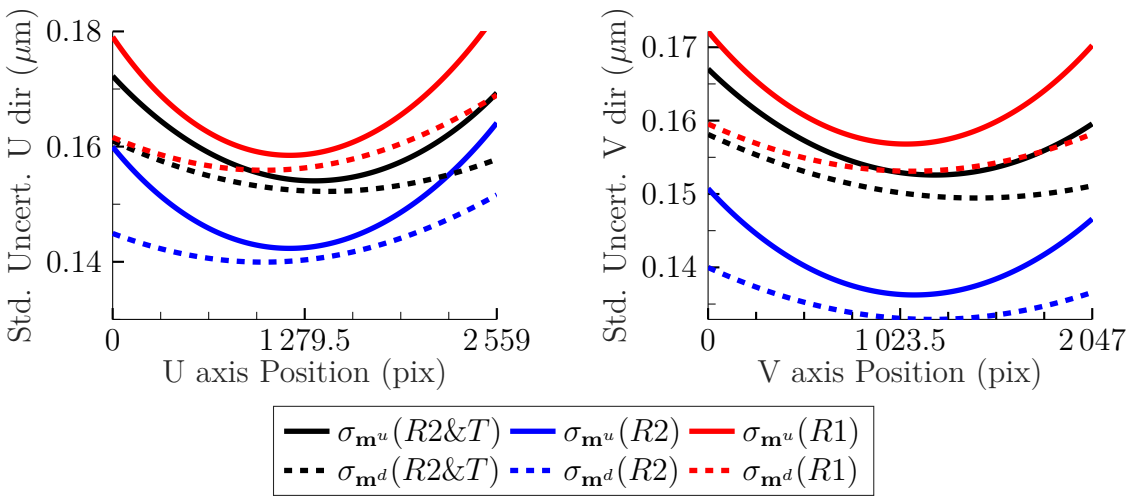

Figure 3.11: Standard uncertainties of the pixels back-projected to the normalized image plane: (a) standard uncertainties in the u coordinate. (b) standard uncertainties in the v coordinate. The dashed lines correspond to the uncertainty before the distortion correction, respectively $\sigma_{m_{\mathrm{u}}^d}$ and $\sigma_{m_{\mathrm{v}}^d}$, and the solid lines correspond to the uncertainty after the lens distortion compensation, respectively $\sigma_{m_{\mathrm{u}}^u}$ and $\sigma_{m_{\mathrm{v}}^u}$. The color of the line represents the employed model.

distortions. In order to describe these coverage regions with a single number, the radius of the corresponding circle with equivalent area ($r_m = \sqrt{r_1 r_2}$) is considered in Table 3.3.

	R2 & T		R2		R1	
Variable	avg.	std	avg.	std	avg.	std
$\mathrm{r_m}^{\mathrm{d}}$(µm)	0.232	0.002	0.210	0.003	0.238	0.003
$\mathrm{r_m}^{\mathrm{u}}$(µm)	0.237	0.005	0.216	0.006	0.244	0.007

Table 3.3: Summary of the standard coverage regions for $\mathbf{m}^{\mathrm{d}}$ and $\mathbf{m}^{\mathrm{u}}$ along the sensor, measured as the radius of circle with area equal to the standard uncertainty ellipse (r_{m}). For every distortion model, the average radius and its range of variation are computed over the whole sensor area, before ($r_{\mathrm{m}}^{\mathrm{d}}$) and after ($r_{\mathrm{m}}^{\mathrm{u}}$) the distortion correction.

As can be seen in Table 3.3, for all distortion models the average radius of the standard coverage region is larger than the standard uncertainty in the individual coordinates: the standard uncertainties for each coordinate where around 0.14-0.16 µm, and the average of the mean radius of the coverage ellipse is around 0.21-0.24 µm. The range of the variation over the sensor is also incremented similarly. This shows the effects of the joint uncertainty of both coordinates. In any case, the analysis of the coverage regions shows similar findings to those of the individual coordinates: **the model corresponding to two radial distortion components (R2) exhibits the narrowest coverage regions, and the uncertainty is dominated by the pinhole parameters**.

Fig. 3.12 shows the spatial variation of the average coverage interval radius r_m for both distorted and undistorted coordinates, for each distortion model. The color code represents the size of the coverage region, with brighter colors meaning larger regions. Note that in order to highlight the structure of the spatial variation, the color scales

have been normalized for each map, thus in order to compare different maps one needs to carefully pay attention to the corresponding colorbars. As can be seen, the coverage regions are expanded as they get further away from a minimum uncertainty point around the principal point (indicated by the green cross). This location of minimum uncertainty is different for each model, and in all cases it is shifted towards the principal point after compensating distortions, which makes sense considering that the distortions are lowest at this location.

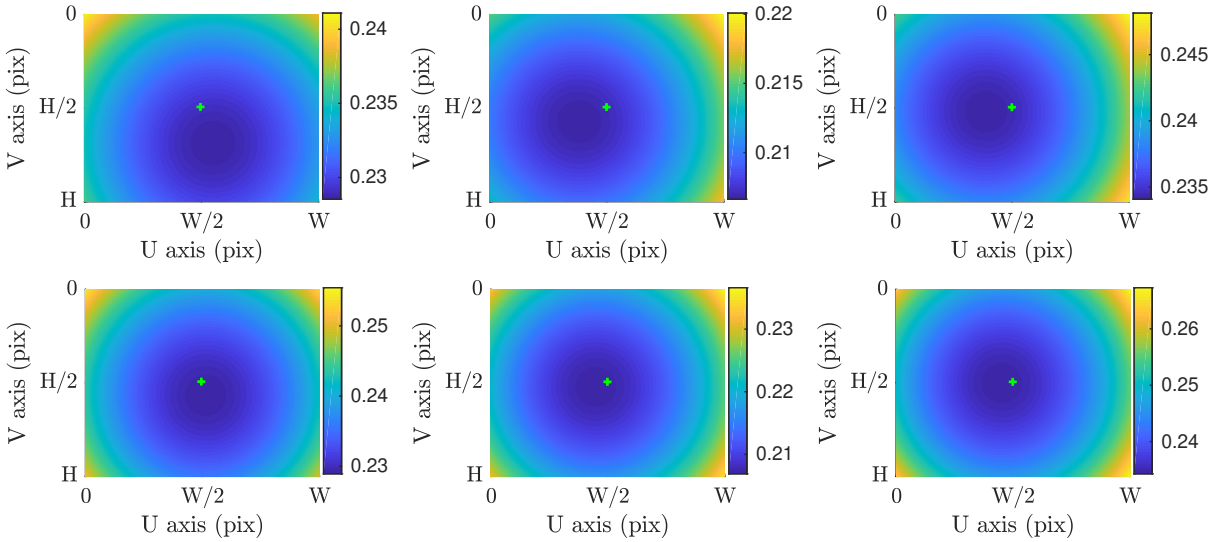

Figure 3.12: Average radius (r_m) of the standard coverage ellipses corresponding to pixels back-projected to the normalized image plane. The first row corresponds to the uncertainties in the distorted coordinates, the second one to the undistorted ones. From left to right, the results correspond to the distortion model with (a) 2 Radial and Tangential terms. (b) 2 Radial terms. (c) 1 Radial term. The color code represents the size of the coverage region measured in µm, with brighter colors meaning larger regions (scales have been normalized for each map). In each figure, the principal point location is indicated with a green cross.

Monte Carlo simulations are used to validate the obtained results following the procedure described in Sec. 3.5. The distribution of the coordinates resulting from the back-propagation perfectly matched the one described by the Taylor approximation (by several orders of magnitude beyond the significant digits) for all tested pixels with every distortion model. The results for the MCM test for a pixel with maximum uncertainty (upper right corner) are shown in Fig. 3.13 and Fig. 3.14, the other results were similar.

3.5.2.2 Uncertainty in Camera Ray Direction

The goal of this experiment is to demonstrate how the propagation model defined in Sec. 3.3 is used to estimate the uncertainties in the camera ray $\hat{\mathbf{i}}$. The uncertainty matrix $\Sigma_{\hat{\mathbf{i}}}$ is computed using 3.37 for every camera pixel, obtaining the variable $\mathbf{u}_{\hat{\mathbf{i}}}$ representing the standard uncertainties of the camera ray over the sensor. The results are summarized in Table 3.4, where the average and the standard uncertainty of $\mathbf{u}_{\hat{\mathbf{i}}}$ are shown for every component of $\hat{\mathbf{i}}$.

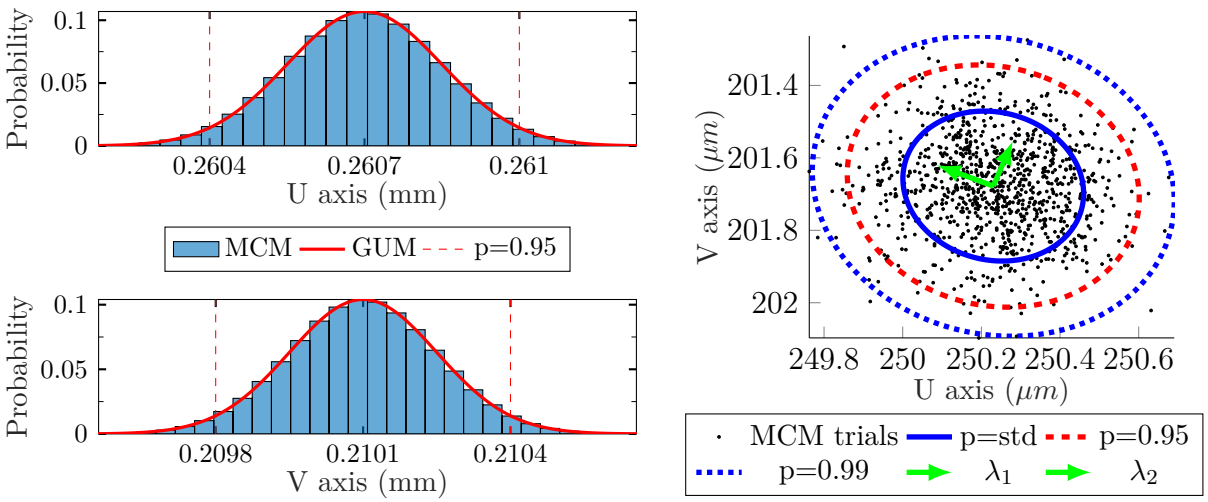

Figure 3.13: Validation of uncertainties in the normalized image plane: Monte Carlo trial results for the back-projected pixel without distortion correction. The histograms show that the distribution of the independent coordinates predicted using GUM matches the one obtained from the MCM test. The ellipsoid plot shows that the standard, p=0.95, and p=0.99 coverage regions arising from the combination of these uncertainties also agree.

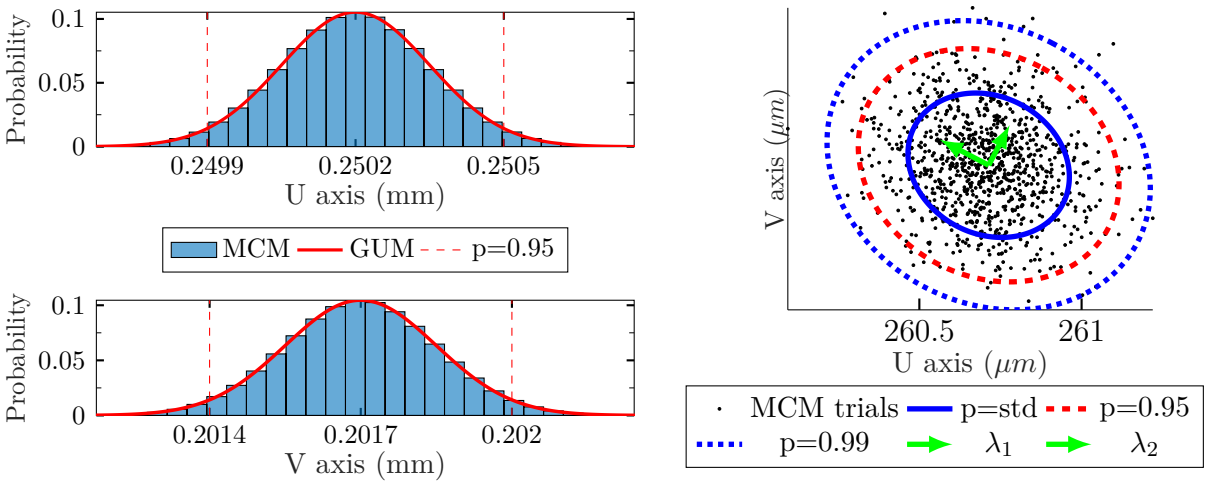

Figure 3.14: Validation of uncertainties in the normalized image plane: Monte Carlo trial results for the back-projected pixel after distortion correction also validate the GUM results.

Model	$\mu_{\mathbf{u}_{\hat{\mathbf{i}}_x}}(\mu\text{m})$	$\sigma_{\mathbf{u}_{\hat{\mathbf{i}}_x}}(\mu\text{m})$	$\mu_{\mathbf{u}_{\hat{\mathbf{i}}_y}}(\mu\text{m})$	$\sigma_{\mathbf{u}_{\hat{\mathbf{i}}_y}}(\mu\text{m})$	$\mu_{\mathbf{u}_{\hat{\mathbf{i}}_z}}(\mu\text{m})$	$\sigma_{\mathbf{u}_{\hat{\mathbf{i}}_z}}(\mu\text{m})$
R2 & T	0.152	0.005	0.150	0.009	0.027	0.052
R2	0.142	0.007	0.134	0.006	0.025	0.049
R1	0.158	0.009	0.154	0.005	0.028	0.055

Table 3.4: Spatial distribution of the uncertainty in the camera ray $\hat{\mathbf{i}}$ over the sensor.

As can be seen in Table 3.4, using distortion model that considers two radial terms (R2) results in the lowest uncertainties, which is logical considering that it resulted in the

smallest uncertainties in the back-projected pixel (recall Table 3.3).

A further simplification can be achieved by considering the uncertainty in the angle between the camera ray and the optical axis, so as to evaluate the uncertainty with a single value (3.39). Table 3.5 shows the results for every distortion model:

Model	$\mu_{u_{\alpha_i}}$(mrad)	$\sigma_{u_{\alpha_i}}$(mrad)
R2&T	0.154	0.003
R2	0.142	0.005
R1	0.160	0.004

Table 3.5: Spatial distribution of the angular uncertainty $\mathbf{u}_{\alpha_i}$ of the camera ray $\hat{\mathbf{i}}$ over the sensor.

The following figures show the spatial distribution of the angular uncertainty over the sensor. As previously shown, the minimum uncertainty is found towards the center of the image, and increases radially to the corners, and it is smaller for the $R2$ model.

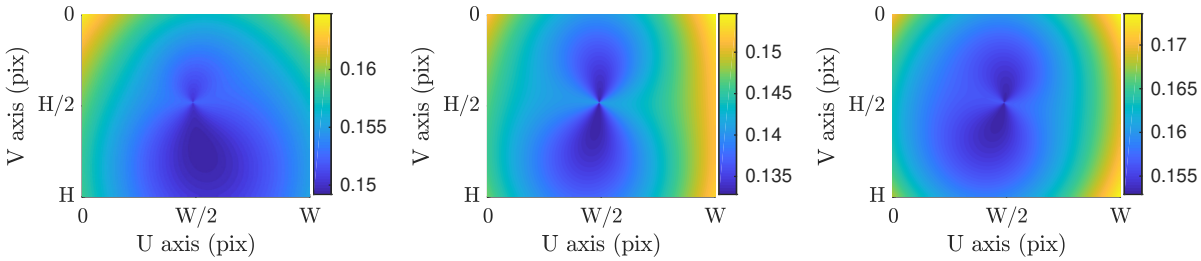

Figure 3.15: Standard angular uncertainties $\Sigma_{\alpha_\mathbf{i}}$ measured in mrad, for the camera rays over the sensor of size WxH. From left to right, the maps corresponding to models $R2\&T$, $R2$, and $R1$.

Similar to the procedure followed in the normalized image plane, the propagation for a series of pixels is analyzed in depth using Monte Carlo simulations. For every employed distortion model, the obtained distribution perfectly matches the one described by the Taylor approximation. Fig. 3.16 shows the results for the pixel with maximum uncertainty (right upper corner), the others being similar.

The distribution of the results of the Monte Carlo trial show that, under the assumption that the uncertainty in the pinhole parameters follows a multivariate normal distribution defined by $\Sigma_{\boldsymbol{\theta}_{\text{pin}}}$, the camera ray vector also follows a multivariate normal distribution. Furthermore, the standard deviation of such distribution matches the standard uncertainty obtained by propagating uncertainties through Taylor, validating the first order approximation under this uncertainty level.

3.5.3 Uncertainty in the Measurements

In this section, the influence of the camera parameter uncertainty on the measurements is analyzed. First the uncertainty of the normal is calculated, and then the uncertainty of the triangulated point. At this point, the camera description is not enough to determine these uncertainties, as they also depend on the reflected ray direction that is defined by the geometrical arrangement of the setup. For this reason, all the experiments in

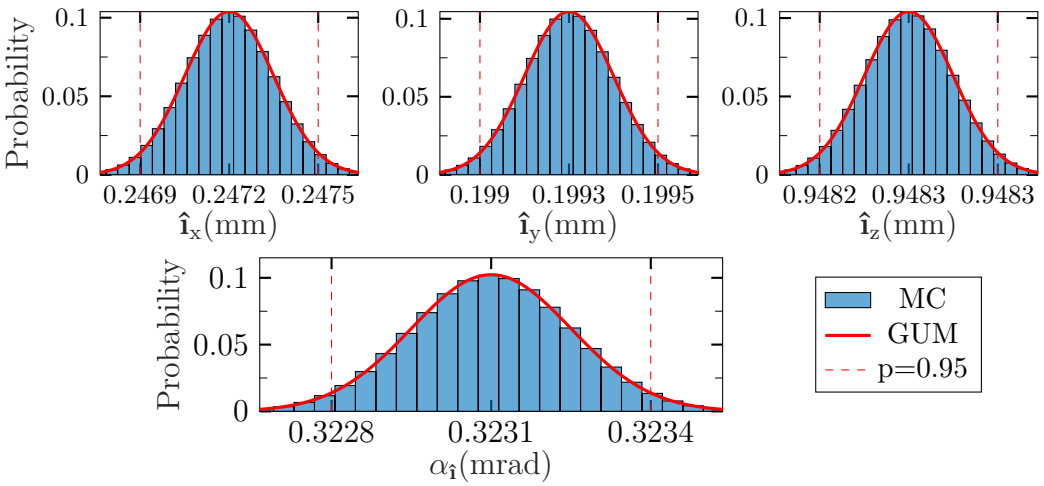

Figure 3.16: Result of the Monte Carlo trials and theoretical error propagation in the components of $\hat{\mathbf{i}}$ $(\sigma_{i_x}, \sigma_{i_y}, \sigma_{i_z})$ for the maximum uncertainty pixel show agreement in the uncertainty of the independent coordinates and the angle with respect to the optical axis.

this section are based on the setup available at our laboratory, taking into account only camera parameter uncertainties, *i.e.*, assuming all the rest of parameters of the setup are perfectly known. In this way, the observed LCD pattern points and corresponding reflected rays ($\hat{\mathbf{r}}$) can be obtained from the geometry of the setup for each camera pixel, and the camera ray uncertainties from Sec. 3.5.2 are used in order to compute the final measurement uncertainties.

3.5.3.1 Measured Normal Uncertainty

In order to calculate the normal uncertainty, equation 3.42 is evaluated at every camera pixel, and the obtained uncertainties $\mathbf{\Sigma}_{\hat{\mathbf{n}}}$ are propagated to the corresponding angular uncertainty of the normal using 3.44. The following figures show the distribution of the uncertainty in the estimated surface normal vector for each camera pixel:

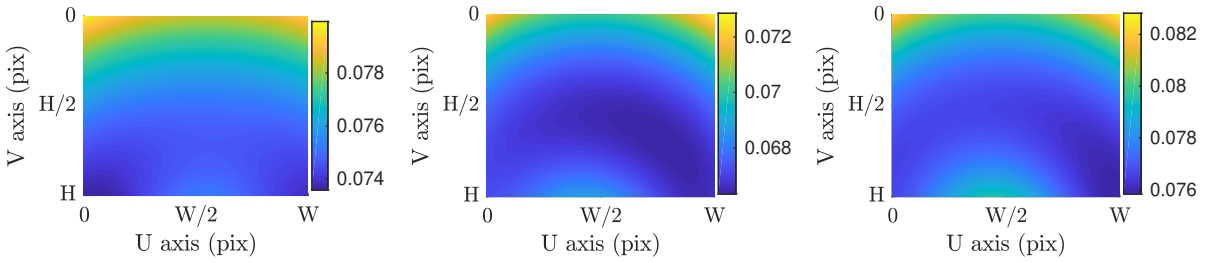

Figure 3.17: Standard angular uncertainties $\Sigma_{\alpha_{\hat{\mathbf{n}}}}$ measured in mrad for the normal vectors estimated over the sensor of size WxH. From left to right, the maps corresponding to models $R2\&T$, $R2$, and $R1$.

The following table summarizes the results shown in the previous figures:

Model	$\mu_{u_{\alpha n}}$(mrad)	$\sigma_{u_{\alpha n}}$(mrad)
R2&T	0.0757	0.0014
R2	0.0677	0.0013
R1	0.0776	0.0013

Table 3.6: Summary of uncertainty tests for $\hat{\mathbf{n}}$: the angular uncertainty of the estimated normal vectors

These results were validated using MCM simulations showing an agreement with the uncertainties propagated using GUM, as shown in 3.18. The results confirm that $R2$ is the model with the least uncertainty in the measured normals, which are smallest around the image center and increase radially to the corners of the image.

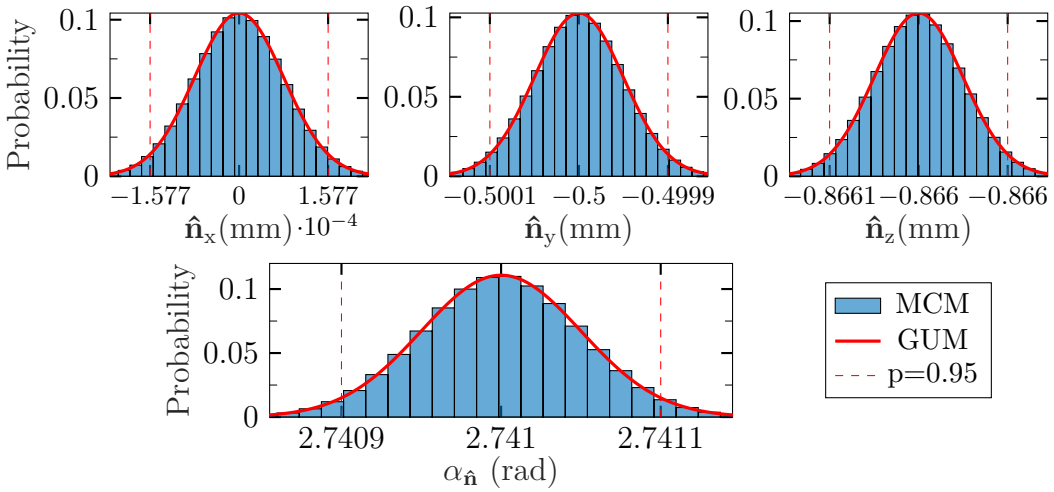

Figure 3.18: Result of the Monte Carlo trials and theoretical error propagation for the normal vector $\hat{\mathbf{n}}$ measured in mrad corresponding to the pixel with maximum uncertainty. The histograms show the agreement of the GUM results with the MCM simulations in the uncertainty of the coordinates of $\hat{\mathbf{n}}$ and the corresponding angle ($\alpha_{\hat{\mathbf{n}}}$) with respect to the camera ray.

3.5.3.2 Triangulated Point Uncertainty

In order to calculate the uncertainty of the triangulated point, equation 3.52 is evaluated at every camera pixel, obtaining the corresponding covariance matrices $\Sigma_{\mathbf{M}^w}$. Table 3.7 summarizes the results, showing the average and standard deviation of the standard uncertainty in each coordinate over the sensor.

Model	μ_{u_x}(µm)	σ_{u_x}(µm)	μ_{u_y}(µm)	σ_{u_y}(µm)	μ_{u_z}(µm)	σ_{u_z}(µm)
R2&T	28	2	32	5	54	7
R2	26	2	28	4	48	6
R1	29	2	32	5	56	7

Table 3.7: Summary of uncertainty tests along the sensor for $\mathbf{M}^w$, measured as the mean($\mu_{\mathbf{u}}$) and standard deviation ($\sigma_{\mathbf{u}}$) of the standard uncertainty of the triangulated position coordinates ($\sigma_{i_x}, \sigma_{i_y}, \sigma_{i_z}$), with $\mathbf{u}_x = \sigma_{M_x^w}(u,v)$, $\mathbf{u}_y = \sigma_{M_y^w}(u,v)$, and $\mathbf{u}_z = \sigma_{M_z^w}(u,v)$.

In order to facilitate the analysis, the coverage intervals are computed. Assuming that the uncertainty follows a multi-variate Gaussian distribution, the coverage region is described by an ellipsoid, whose principal axes ($\mathbf{v}_{M1}, \mathbf{v}_{M2}, \mathbf{v}_{M3}$) correspond to the eigen-vectors of $\Sigma_{\mathbf{M}}$ and the length of the axes are given by the corresponding eigen-values (λ_{M1},λ_{M2},λ_{M3}). In this way, the 3x3 matrix can be reduced to three parameters: $u_{M1} = \sqrt{\lambda_{M1}}$, $u_{M2} = \sqrt{\lambda_{M2}}$, and $u_{M3} = \sqrt{\lambda_{M3}}$, that are the standard uncertainties of the triangulated point in such directions. In our experiments, the smallest eigenvalue was always close to zero, meaning that the uncertainty is described by a plane almost perpendicular to the direction of the camera ray $\hat{\mathbf{i}}$, as shown in Fig. 3.19. This allows a further simplification, as the radius of the corresponding circle with equivalent area ($r_m = \sqrt{r_1 r_2}$) can be used to describe the coverage region with a single number. Figs. 3.20(a)-(c) show the radius of the coverage region for the surface point observed in each camera pixel.

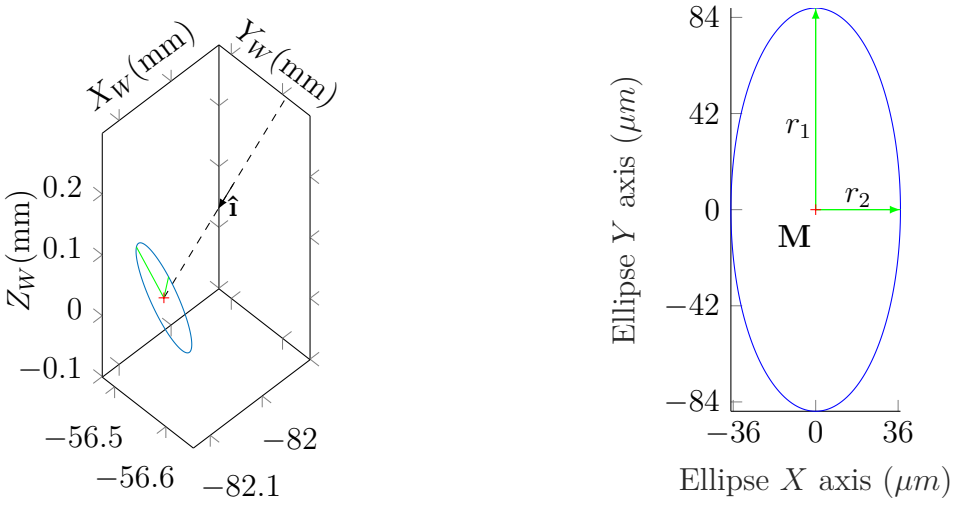

Figure 3.19: Standard coverage region for the triangulated point **M** corresponding to a pixel close to the lower left corner of the image. The left side shows the oblique elliptical surface describing the coverage region arising from the uncertainty in the camera ray direction $\hat{\mathbf{i}}$. The right side depicts the plane of the ellipse. In both drawings the red cross corresponds to **M** and the green lines correspond to the major (r_1) and minor (r_2) axes of the ellipse.

The following table summarizes the results shown in the previous figures:

Model	μ_{r_1}(µm)	σ_{r_1}(µm)	μ_{r_2}(µm)	σ_{r_2}(µm)	μ_{r_m}(µm)	σ_{r_m}(µm)
R2&T	96	6	41	3	63	2
R2	86	6	38	3	57	2
R1	99	7	43	3	65	2

Table 3.8: Summary of uncertainty tests for $\mathbf{M}^w$: the standard coverage region of the triangulated point is described by a elliptical plane with major and minor axes r_1 and r_2. r_m corresponds to the radius of the circle with equal area, allowing us to compare the size of the coverage regions with a single variable.

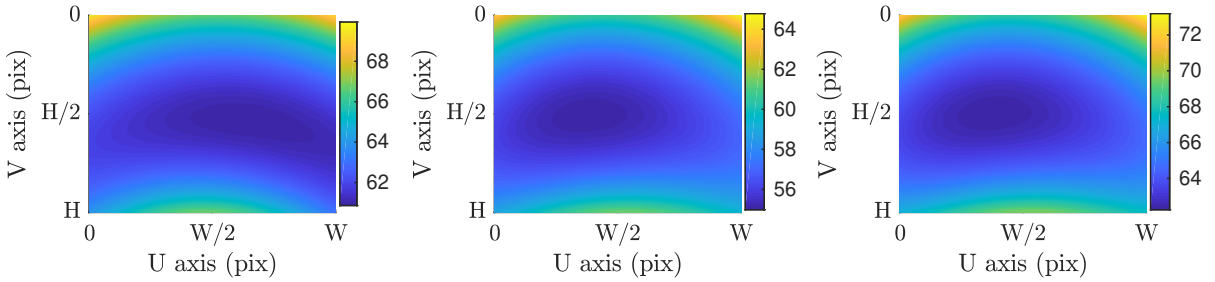

Figure 3.20: Average radius (r_m) of the standard coverage ellipses corresponding to the triangulated points $\mathbf{M}^w$ for every image pixel. The color code represents the radius of the coverage region measured in µm, with brighter colors meaning larger regions (scales have been normalized for each map). In each figure, the principal point location is indicated with a green cross. From left to right, the results correspond to the distortion model with (a) 2 Radial and Tangential terms. (b) 2 Radial terms. (c) 1 Radial term.

The results were validated using MCM simulations, which agreed with those obtained through GUM. The results for the MCM test for a pixel with maximum uncertainty (right upper corner) are shown in Fig. 3.22 and Fig. 3.21, the other results were similar.

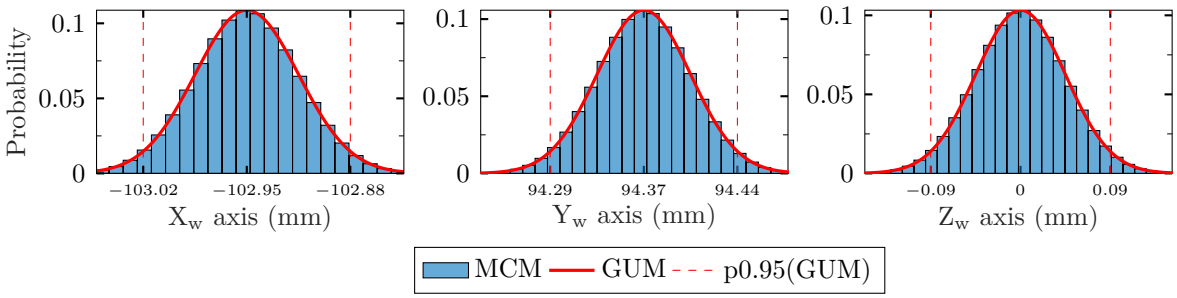

Figure 3.21: Monte Carlo results: histograms corresponding to the coordinates of the triangulated point $\mathbf{M}^w$ observed after 1.0e8 MCM iterations. In these graphs the y axis represents the pdf (probability of the random variable falling within a particular range of values), and the x axis shows the corresponding coordinate, indicating the mean μ (best estimate) and the edges of the symmetric 95% coverage interval μ-2σ and μ+2σ estimated from the observations. As can be seen in the figures, both distributions agree.

These results confirm that $R2$ is the model with the least uncertainty in the triangulated points, which is smallest around the image center and increases radially to the corners of the image.

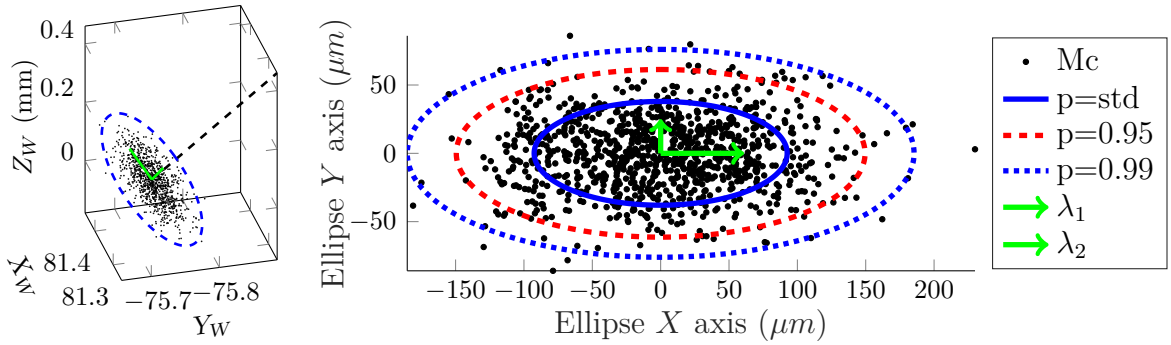

Figure 3.22: Monte Carlo results confirm the std, 95%, and 99% elliptical coverage regions arising from the combination of the individual coordinate uncertainties also agree.

3.6 Chapter Conclusions

This chapter has analyzed the influence of the camera calibration uncertainty on deflectometric measurements using the extended pinhole camera model.

First, we model the propagation of the uncertainty from the calibrated camera parameters to the camera rays, considering the uncertainties in each intermediate operation. These uncertainties can be used to identify the parameters that most affect the measurements and also to select the distortion model. The uncertainty model can also be used to obtain an estimate of how these uncertainties affect the measured normal vector field and the triangulated surface.

Afterwards, several experiments are performed using the uncertainty model in the setup in our laboratory. Three different variations of the distortion model are considered, which correspond to a radial distortion with a single coefficient ($R1$), radial distortion with two coefficients ($R2$), and radial distortion with two coefficients plus two tangential distortion coefficients ($R2\&T$). The following list outlines the main findings:

- The $R2$ model results in smaller uncertainties in the camera rays, although it should be noted that they are of the same order of magnitude for the three models ($0.14 - 0.16$ mrad). The resulting uncertainties in both the surface normal and triangulated surface also follow this relation, with standard uncertainties close to 0.07 mrad in the normal vectors and 50 μm in the z coordinate of the triangulated points, for an approximate field of view of 200 x 140 mm.

- The variation of these uncertainties over the sensor is relatively small, with a standard deviation one order of magnitude smaller than the average uncertainty.

- The analysis of uncertainties in the normalized image plane shows that the pinhole parameters have a higher influence than the distortion parameters. Among the pinhole parameters, the principal point has a higher influence than the focal distance.

Regarding future lines, several lines of work can be proposed:

- All the experiments have been performed with a particular camera and lens. In order to generalize the drawn conclusions, it would be necessary to repeat the experiments with a wider of range of camera and lens combinations.

- Although the analysis of the camera ray uncertainty only depends on the employed camera, the uncertainties in the final measurements also depend on the setup geometry. It would be interesting to extend this work to also include the setup parameters, or at least to consider several variations of the setup.
- Environmental disturbances typically present in industrial scenarios have not been taken into account. Further works should consider the effects of temperature variations, changing lightning conditions, and mechanical vibrations of the setup, among others.
- This work has focused on the extended camera pinhole, it would be interesting to do the same for other camera models for comparison purposes.

Chapter 4

Influence of the Screen Model on Measurement errors

4.1 Introduction

In the previous chapter, the uncertainties associated to the camera rays and their effects on deflectometric measurements have been investigated by considering the pinhole model calibration uncertainties. This chapter concentrates on the reference structures observed by these rays, focusing on the measurement errors arising from inaccurate assumptions on the LCD screen model.

In deflectometric measurements, the camera observes a reference structure displaying spatially coded patterns through their reflection on the surface under inspection (SUT). The spatial coding provides correspondences between image locations and observed pattern locations that constrain the shape of the reflecting surface. Although different deflectometric techniques exploit these correspondences in different ways, they are all affected by errors of the observed feature point locations up to some extent.

The most commonly used pattern coding strategy is based on the observation of multiple sequences of phase-shifted sinusoidal fringes, which has been called Phase Measuring Deflectometry (PMD) [KKH04]. In this approach, the phase of sinusoidal patterns provides a dense encoding of the reference plane, with the advantage that if the period of the sinusoids is chosen correctly, their phase is insensitive to the blurring caused by defocussing. This is very useful considering that the camera can not be usually focused on the surface and on the pattern simultaneously [HRW13].

Early work by Knauer et al. [KKH04] analyzed the physical limitations of PMD by considering an ideal deflectometry setup operating at the diffraction limit. They show that there is an uncertainty relation that couples the uncertainty of the slope measurement with the uncertainty of the measured position on the surface and the uncertainty of the decoded phase. However, in practical implementations with LCD screens there are other issues that also need to be taken into account [PFT13].

LCDs are considered suitable for generating accurate reference structures because they are manufactured to very high precision using lithographic techniques. Based on this premise, many works consider an LCD as an ideal device that generates 2D light patterns with characteristics close to a Lambert radiator. As shown in Fig. 4.1, an LCD screen is a complex structure composed of several transparent layers on top of the light source, including a liquid crystal layer, glass covers, polarizers, color filters, or diffusers

[YW14], and there are a series of factors that lead to deviations from an ideal pattern generator.

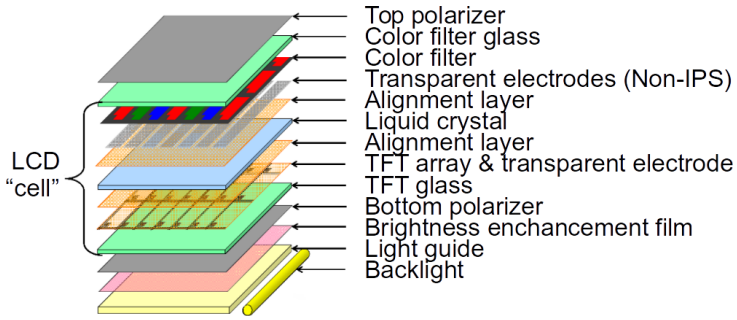

Figure 4.1: Sketch of an LCD display device [Wal14].

On the one hand, there are several photometric distortions and noise sources that contribute to the phase uncertainty introducing high-frequency errors in the measurements. These effects can originate from the discrete nature of the LCD panel structure itself, *e.g.*, it has been shown that the fill factor of the screen pixels [PT05] or their size [XGJ18] influences the measurement uncertainty. The influence of non-linear brightness curve of the screen due to gamma correction has also been studied by several researchers [FPT10, Rap12, PFT13]. These works show that the use of phase shifting algorithms can alleviate the effects of these non-linear errors [Rap12], particularly for LCDs with IPS technology [FPT10]. However, if color displays are used to generate grayscale patterns, the different luminance of the three color sub-pixels of the screen and the spectral sensitivity of the camera can still cause considerable phase errors [PFT13]. The noise introduced by the camera and the microstructure of the observed surface are another source of high-frequency errors [HRW13, FPT16, PBKB19].

On the other hand, several authors have pointed out that low-frequency errors are introduced by two common simplifications in the employed screen model [PT05, PFT13, BNP+18]. The first one consists in assuming a planar screen. The structure of the screen is not a rigid body, as it possesses a certain degree of flexibility that results in deformations in the order of 1 mm. Furthermore, Bartch et al. showed that such deformations are influenced by temperature as well as gravity and physical stress [BNP+18]. If these deviations from planarity are not taken into account, the conversion from pixel coordinates to metric 3D coordinates will introduce systematic errors in the measurements. The second one consists in ignoring the refraction generated in the transparent layers. The transparent layers generate a shift in the observed pattern, as the light-path is bent because of the refraction that the light suffers at the boundary between different materials. If these shifts are not taken into account, and one assumes that the incoming light-path follows a straight line, there will be additional systematic errors in the 3D coordinates of the observed reference point.

Fortunately, the phase uncertainty arising from high-frequency errors can be mitigated with an appropriate selection of the pattern coding strategy [Rap12, HRW13, WYY+15] and careful setting of the screen and camera adjustments [FPT10]. In contrast, there are still open questions regarding how to compensate for the low-frequency errors and the effects of not compensating. Note that these errors are independent from the decoding uncertainty and influence both the calibration procedure and the measurement procedure. Still, and despite the calibration being commonly considered as the current limitation

of deflectometric techniques [HFOE13, HIZA18], only few works have attempted to include these LCD model errors into the calibration procedures. Bartch et al. published a method that accounted for the non-planar display shape [BKB19], showing that a polynomial model for the display shape significantly reduced systematic measurement errors. However, the deflectometric setup employed in this work includes a confocal sensor, which would considerably increase the price of the measurement system. Other works have been proposed with simpler setups, *e.g.*, [OH14, PT05], but they do not fully detail the employed procedures.

The aim of this chapter is to analyze the influence of these low-frequency errors on the measurements performed on an Active Reflection Grating Photogrammetry setup [PT05]. Instead of approaching this analysis using uncertainty propagations as in Ch. 3, a simulation based approach is followed to investigate their effects on the final measurements, because the errors are systematic and depend strongly on the employed setup. The simulations will be based on the setup available in our laboratory.

The rest of the chapter is divided as follows. In Sec. 4.2, the measurement model is described considering the different measurement errors. In Sec. 4.3 the screen employed in the experimental measurements of Sec. 4.4.1 is characterized. In Sec. 4.4.2 several simulations are performed considering each error source independently as well as in combination with the others. Afterwards, the simulations will be compared to real measurements. Finally, the chapter ends with the conclusions and future lines in Sec. 4.5.

4.2 Measurement Model

Active Reflection Grating Photogrammetry[PT05] is a deflectometric technique proposed by Petz et al. [PT05]. In this approach, the height-angle ambiguity (see section 2.5) is solved by moving the LCD to at least two positions while maintaining the camera and the SUT steady, as shown in Fig. 4.2. In this way, for an image pixel $\mathbf{m}$ observing the surface point $\mathbf{M}$, the direction of the reflected ray $\hat{\mathbf{r}}$ is defined by the two decoded screen points $\mathbf{Q}_1$ and $\mathbf{Q}_2$. From there, the surface normal vector $\mathbf{n}$ direction can be estimated as the half vector between the estimated $\hat{\mathbf{r}}$ and the camera ray $\hat{\mathbf{i}}$, and the 3D location can also be estimated using triangulation.

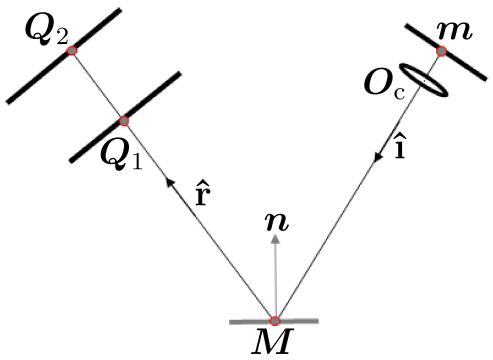

Figure 4.2: Reflection Grating Photogrammetry measuring principle: A surface point $\mathbf{M}$ and its normal vector $\mathbf{n}$ are defined by the intersection of the incident and reflected rays.

In the remainder of this section, the measurement procedure is described together with the errors introduced in each step.

4.2.1 Light-map computation

In this section we describe how the light-map is estimated, *i.e.*, the correspondences between the camera pixels and the observed LCD locations. The first subsection deals with the sinusoidal encoding of the screen positions is described and the second focuses on the algorithms employed for their decoding.

4.2.1.1 Spatial Encoding Using Sinusoidal Patterns

Sinusoidal patterns encode the coordinates of each screen pixel $p_l = [u_l, v_l]^T$ in their phase. Usually two phase maps ϕ_u and ϕ_v are employed for the absolute spatial coding of the pattern:

$$\phi_u(u_l, v_l) = \frac{2\pi}{L} u_l; \qquad \phi_v(u_l, v_l) = \frac{2\pi}{L} v_l \tag{4.1}$$

where L is the pattern period.

The intensity I_l of a displayed pattern is:

$$I_{l\xi}(u_l, v_l) = A\,(1 + cos\,(P_\xi\,(u_l, v_l))) \tag{4.2}$$

where $I_{l\xi}$ is the intensity of the displayed pattern corresponding to a particular direction $\xi \in [u, v]$, A is half the peak-to-valley amplitude and P_ξ is the phase value that encodes the screen location.

4.2.1.2 Phase Map Decoding

In order to recover a phase map, phase shifting methods require displaying a series of N patterns of the same wavelength but shifted by $2\pi/N$ [SB06]. The intensity I of the captured images is described by:

$$I_{\xi i}(u_c, v_c) = I_{\text{avg}} + I_{\text{pv}} \cos(\phi_\xi + \delta_i) \tag{4.3}$$

where $I_{\xi i}$ represents the ith image intensity at camera pixel $p_c = [u_c, v_c]^T$, when a pattern encoding the coordinate ξ is displayed on the LCD screen. I_{avg} is the average intensity, and I_{pv} is half the peak-to-valley intensity modulation, δ_i is the phase shift, and ϕ_ξ is the phase of the observed pattern point, defined as:

$$\phi_\xi(u_c, v_c) = \frac{2\pi}{L} l_r(u_c, v_c)\,; \tag{4.4}$$

where L is the period of the sinusoid on the LCD and l_r is the mapping from the camera to the screen.

The phase is extracted from the images:

$$\phi_\xi(u_c, v_c) = \tan^{-1}\left(\frac{-\sum_{i=1}^{N} I_i \sin(\delta_i)}{\sum_{i=1}^{N} I_i \cos(\delta_i)}\right) \tag{4.5}$$

It has been found that uncertainty of the decoded phase can be sufficiently approximated by a normal distribution, which depends on the number of employed phase shifts (N), the the average intensity I_{avg} and contrast I_{pv} of the captured patterns, and the

signal to noise ratio of the camera [HRW13]. In order to reduce the uncertainty of the decoded screen locations, it is desirable to employ patterns with the smallest possible period that provide "enough" contrast in a particular type of surface [KKH04, HRW13]. In most cases, such an optimal pattern period is smaller than the screen length, and the decoding process will result in ambiguities due to the 2π modulus operation inherent to sinusoidal functions. This is known as the phase-wrap problem, and it can be solved by using an unwrapping method to find the fringe order corresponding to each phase value [GP98]. The resulting phase-map will extend the original $0-2\pi$ range, such that there is no ambiguity.

In this work we employ a hierarchical multi-frequency unwrapping method based on [Zha09], which consist in acquiring a sequence of K phase-maps with different periods in decreasing order (following an exponential sequence). The coarsest one ϕ_1 must be at least as wide as the LCD screen, so that no phase-wraps are present in it. The finest one, ϕ_K, will be the one that will actually be used for the measurements. The first unwrapped phase Φ_1 is obtained as:

$$\Phi_1\left(u_c, v_c\right) = \phi_1\left(u_c, v_c\right) \tag{4.6}$$

The rest of the phase-maps Φ_k are successively unwrapped using the information from the previous one as shown in eq. 4.7, until the phase-map with finest period is decoded:

$$\Phi_k\left(u_c, v_c\right) = \phi_k\left(u_c, v_c\right) + 2\pi m_k\left(u_c, v_c\right) \tag{4.7}$$

where the fringe order m_k is computed using the previous unwrapped phase map Φ_{k-1} and the actual wrapped phase map ϕ_k

$$m_k\left(u_c, v_c\right) = \left[\frac{M\Phi_{k-1}\left(u_c, v_c\right)}{2\pi} - \frac{\phi_k\left(u_c, v_c\right)}{2\pi}\right] \tag{4.8}$$

where $[\ \ ]$ represents the *Round* function, M is the ratio between the period of two consecutive phase maps.

From the unwrapped phase-map a one to one mapping $\mathbf{l}_{\text{px}} : \mathbb{N}^2 \rightarrow \mathbb{R}^2$ is obtained from every camera pixel $\mathbf{m}$ to the center of the observed screen area observed in it $\mathbf{q}^p = [q_u^p, q_v^p]^{\text{T}}$. This mapping is called a light-map and is given by:

$$\mathbf{l}_{\text{px}}(u_c, v_c) = \left[\frac{L}{2\pi}\Phi_u(u_c, v_c), \frac{L}{2\pi}\Phi_v(u_c, v_c)\right] \tag{4.9}$$

Each decoded pattern location $\mathbf{q}^p$ will be affected by the high-frequency errors caused by radiometric noise, non linearity of screen and camera, fill factor and color sub-pixels of the screen pixels, etc. However, modeling these errors is out of the scope of this work and only a general uncertainty will be considered, assuming a zero-mean normal distribution independent for each pixel, such that:

$$\mathbf{q}^p = \bar{\mathbf{q}}^p + \delta_{\mathbf{q}^p}, \qquad \delta_{\mathbf{q}^p} \approx \mathcal{N}(0, \sigma_{\mathbf{q}^p}) \tag{4.10}$$

where $\bar{\mathbf{q}}^p$ is the true location of the observed LCD location.

4.2.2 Pattern to 3D Location

Each decoded point $\mathbf{q}^p$ is a 2D location in the pattern frame given in pixels. In this step, a conversion to the corresponding 3D location on the LCD frame in mm ($\mathbf{Q}^s$) is performed.

If a planar screen model is assumed, with square pixels of size s_{lcd}, then the mapping function becomes trivial, denoted ${}_{\pi}f_p$ in 4.11:

$$\mathbf{Q}^s = {}_{\pi}f_p(\mathbf{q}^p) = \begin{bmatrix} q_u^p \\ q_v^p \\ 0 \end{bmatrix} \cdot s_{\text{lcd}} \tag{4.11}$$

Many researchers have pointed out that an LCD screen is actually not perfectly flat, a fact that is also corroborated in our experiments in Sec. 4.3.1. Thus, using ${}_{\pi}f_p$ will introduce a systematic error δ_{shape} on $\mathbf{Q}^s$. Besides this, the error due to the refraction ($\delta_{refract}$) has to be taken into account. The refractive layers of the LCD screen generate a shift in the observed pattern points due to the refraction that the light suffers at the boundary between different materials. Therefore, a decoded pattern point $\mathbf{Q}^s$ will not lie on the reflected ray we want to estimate. These errors can be modeled as:

$$\mathbf{Q}^s = \bar{\mathbf{Q}}^s + \delta_{shape} + \delta_{refract} \tag{4.12}$$

where $\bar{\mathbf{Q}}^s$ is the true location of a point on the reflected ray $\mathbf{r}$.

4.2.3 Computing the Reflected Ray

The reflected ray direction is obtained from the line connecting the points obtained from the screen in two positions, namely $\mathbf{Q}_1^{s1}$ and $\mathbf{Q}_2^{s2}$. These points have to be transformed to a common frame, the camera frame. For this purpose the geometric relation between each screen and the camera frame need to be calibrated in advance. This is referred to as extrinsic calibration and will be explained in the following section.

$$\hat{\mathbf{r}} = \frac{\mathbf{Q}_2^c - \mathbf{Q}_1^c}{||\mathbf{Q}_2^c - \mathbf{Q}_1^c||} \tag{4.13}$$

$$\mathbf{Q}_{\text{i}}^c = {}_{\text{c}}\mathbf{R}_{\text{s}_i}\mathbf{Q}_i^{s_i} + {}_{\text{c}}\mathbf{t}_{\text{s}_i}^{\text{c}}, \qquad i = 1, 2 \tag{4.14}$$

where ${}_{\text{c}}\mathbf{R}_{\text{s}_i}$ and ${}_{\text{c}}\mathbf{t}_{\text{s}_i}^{\text{c}}$ are the rotation and translation between the camera and the LCD in the ith position.

Note that the points $\mathbf{Q}_1^{s1}$ and $\mathbf{Q}_2^{s2}$ contain errors caused by both the high-frequency errors of the decoded light-map as well as the low-frequency errors corresponding to the non-planar shape and refraction. In addition, the errors and uncertainty of the extrinsic parameters and the positioning uncertainty of the linear stage will also contribute to the errors in these points.

4.2.4 Extrinsic Calibration

The concept is similar to camera calibration: a known pattern is displayed on the screen and captured by the camera, obtaining a set of 2D-3D correspondences that allows the position and orientation of the LCD screen relative to the camera to be estimated by fitting a homography between these correspondences and the previously calibrated internal camera parameters $\boldsymbol{\theta}_{\text{cam}}$. However, the camera does not usually have a direct view of the screen, and a first surface mirror is used to view it indirectly [KKH04]. Therefore, the obtained extrinsic parameters ${}_{\text{c}}\mathbf{R}_{\text{vs}_i}$ and ${}_{\text{c}}\mathbf{t}_{\text{vs}_i}^{\text{c}}$ correspond to a virtual reflected screen.

In order to recover the pose of the "real screen" it is thus necessary to take into account the transformation corresponding to the reflection through the mirror. For this purpose, it is necessary to find the reflection plane $\pi_r = ax + by + cz + d = 0$. The pose of the mirror with respect to the camera (${}_{\mathrm{c}}\mathbf{t}_{\mathrm{m}}^{\mathrm{c}}$ and ${}_{\mathrm{c}}\mathbf{R}_{\mathrm{m}}$) can be obtained using marks embedded on the mirror [KKH04]. Alternatively, if the mirror has no embedded marks, the reflection plane can be estimated using a calibration plate placed on top of the mirror [FY07]. If the calibration plate and the mirror are assumed to be perfectly flat, and thus parallel planes, only the plate thickness needs to be compensated for. In any case, the obtained pose defines the reflection plane, with its normal vector $\mathbf{n}_{\mathrm{r}} = [a, b, c]^{\mathrm{T}}$ being equal to the Z axis:

$$\mathbf{n}_{\mathrm{r}} = \mathbf{Z}_{\mathrm{m}}^{c} = {}_{\mathrm{c}}\mathbf{R}_{\mathrm{m}} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{4.15}$$

and the distance to the origin d is computed as:

$$d = -{}_{\mathrm{c}}\mathbf{t}_{\mathrm{m}}^{\mathrm{c}} \cdot \mathbf{n}_{\mathrm{r}} \tag{4.16}$$

Once the reflection plane is known, the real pose of the screen can be recovered using the Householder transformation [Str05]:

$${}_{\mathrm{c}}\mathbf{R}_{\mathrm{s}_i} = \left(\mathbf{I} - 2\mathbf{n}_{\mathrm{r}}\mathbf{n}_{\mathrm{r}}{}^{\mathrm{T}}\right) {}_{\mathrm{c}}\mathbf{R}_{\mathrm{vs}_i} \tag{4.17}$$

$${}_{\mathrm{c}}\mathbf{t}_{\mathrm{s}_i}^{\mathrm{c}} = \left(\mathbf{I} - 2\mathbf{n}_{\mathrm{r}}\mathbf{n}_{\mathrm{r}}{}^{\mathrm{T}}\right) {}_{\mathrm{c}}\mathbf{t}_{\mathrm{vs}_i}^{\mathrm{c}} - 2d\mathbf{n}_{\mathrm{r}} \tag{4.18}$$

Note that the estimated pose will contain errors originated from the non-planar shape and refractive layers of the screen, and also from the uncertainty of the decoded pattern, the reflection plane estimation, and the camera calibration.

4.2.5 Normal Vector and Triangulated Point Computation

Finally, the normal vector $\hat{\mathbf{n}}$ and the triangulated point $\mathbf{M}$ are calculated as explained in Sec. 3.4:

$$\hat{\mathbf{n}} = \frac{\hat{\mathbf{r}} - \hat{\mathbf{i}}}{||\hat{\mathbf{r}} - \hat{\mathbf{i}}||} \tag{4.19}$$

$$\mathbf{M} = \boldsymbol{S}^{+}\mathbf{C} \tag{4.20}$$

where $\mathbf{S}$ and $\mathbf{C}$ are the variables defined previously in 3.50 and 3.51 as

$$\mathbf{S} = \left(\hat{\mathbf{r}} \cdot \hat{\mathbf{r}}^{\mathrm{T}} - \mathbf{I}\right) + \left(\hat{\mathbf{i}} \cdot \hat{\mathbf{i}}^{\mathrm{T}} - \mathbf{I}\right) \tag{4.21}$$

and

$$\mathbf{C} = \left(\hat{\mathbf{r}} \cdot \hat{\mathbf{r}}^{\mathrm{T}} - \mathbf{I}\right) \cdot \mathbf{Q}_1 + \left(\hat{\mathbf{i}} \cdot \hat{\mathbf{i}}^{\mathrm{T}} - \mathbf{I}\right) \cdot \mathbf{O}_{\mathrm{c}} \tag{4.22}$$

4.3 Screen Characterization

In order to simulate the effects of the low-frequency errors due to the refraction and non-planar shape of the screen, first it is necessary to quantify the model errors from a real LCD screen. The experiments performed for such purpose are described in this section.

4.3.1 Screen Shape

In order to analyze the effects of the non-planarity of the screen, the shape of the front surface of the LCD screen was measured using a coordinate measuring machine (CMM). The LCD was unmounted from the deflectometry setup and set lying on the CMM table with its Z axis pointing upwards. A grid of 3D points was acquired with a sampling rate of 1mm. The measurements are performed in a local coordinate system, which we choose to align to the screen frame: the origin $\mathbf{O}_\mathrm{s}$ is located at the center of the screen, with the X axis being parallel to the horizontal edge of the screen and positive towards the right, the Y axis being parallel to the vertical edge of the screen and pointing downwards, and the Z axis being perpendicular to the XY plane pointing outwards from the front screen surface. The resulting surface can be seen in Fig. 4.3

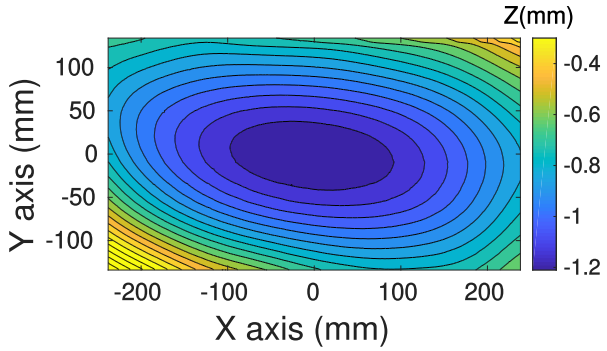

Figure 4.3: Shape of the LCD screen measured using a coordinate measuring machine.

The deformation of the screen is assumed to be smooth, enabling the front surface to be characterized with a continuous function that is approximated locally by fitting the points to a piecewise bi-cubic B-spline:

$$z = S_\mathrm{f}(x, y) = \sum_{i=0}^{3} \sum_{j=0}^{3} c_{i,j} \phi(x) \psi(y) \tag{4.23}$$

where c is the coefficient array (knot sequence) and $\phi(x)$ and $\psi(y)$ are the basis functions of the splines.

Note that the shape of the LCD screen will change once it is mounted back in the deflectometry setup due to gravity and physical stress [BNP+18]. Therefore, the measured shape is only an approximation of its real deformation, and although it is useful for simulation experiments, it should not be used to correct the measurements.

4.3.2 Refractive Layer

Petz et al. [PT05] proposed that the layer structure of an LCD screen can be modeled as a single transparent layer, as most of the refraction occurs in the glass substrate (the other layers are of negligible thickness). In [Pet06] the author presented a method to compensate for the refraction effect and showed that the single layer refraction model can improve the measurement accuracy, but the thickness and refraction index need to be known; these parameters are not usually available from the manufacturer. In [PFT13] the authors mentioned that it is possible to estimate both parameters experimentally using a pattern composed of nine circular markers, eight of which are attached to the

front surface of the screen and an additional one displayed on it. They used a camera to observe the pattern from different angles and obtain the unknown parameters from the displacement of the displayed marker with respect to the attached ones. As the article is not focused on this method, they do not go into detail: they do not explicitly state the involved equations, nor how they solve them, and they do not show any analysis of the results either. For this reason, we proposed a method [MIA17] to estimate the thickness and index of refraction of the single layer model corresponding to an LCD screen.

Inspired by [PFT13], our method relies on displaying a series of feature points on the screen and observing them from different directions, but instead of analyzing the relative shift, the light-path corresponding to each observation is considered. For this purpose, first the LCD screen is measured with a CMM as explained in the previous section, obtaining the shape of its front surface. Afterwards, without moving the screen, a calibration plate is attached to the LCD to provide a visual reference between the measured front surface of the screen and the scene observed by the camera. Then, a series of feature points are generated by displaying a chessboard pattern on the screen, and a camera observes them from different directions at an approximate distance of 600 mm. The calibration plate and the displayed pattern can be seen respectively in the red and blue rectangles in Fig. 4.4 (a).

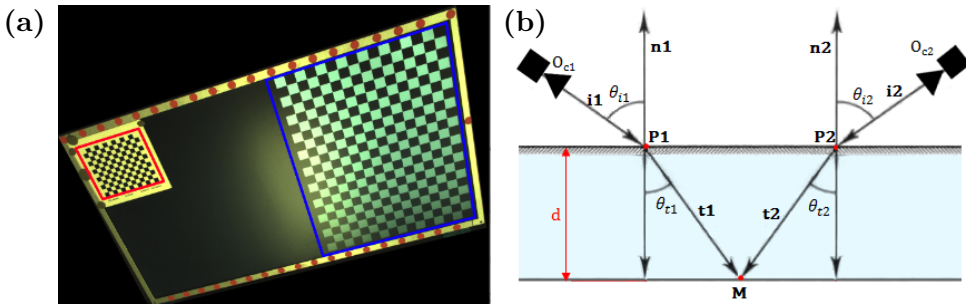

Figure 4.4: (a) Front and back surface feature points. (b) Sketch of the method to estimate the refractive layer parameters.

Note that each of the displayed feature points is observed multiple times through different light-paths, as shown in Fig. 4.4 (b). The light path corresponding to each observation of a feature point $\mathbf{M}$ is composed of two line segments: the incident ray $\mathbf{i}_i$ connecting the optical center of the camera with a point on the front surface $\mathbf{P}_i$, and the transmitted ray $\mathbf{t}_i$, which connects $\mathbf{P}_i$ with $\mathbf{M}$. The relative pose between the camera and the LCD screen is obtained using the calibration plate, and the location of $\mathbf{P}_i$ is calculated as the intersection of the observation ray $\mathbf{i}_i$ with the front surface model. Since $\mathbf{i}_i$, $\mathbf{P}_i$, and the surface normal $\mathbf{n}_i$ are known, the location of a feature point $\mathbf{M}$ only depends on the unknown model parameters of the refractive layer, the index of refraction η and thickness d, which are estimated by minimizing the error of the estimated location $\mathbf{M}$ for each observed feature point.

Several simulation experiments were performed showing that the method converged to the right values. However, in real experiments minimization errors were too large and the algorithm got stuck in local minimums dependent on the initial values of η and d. We suspect that this might be related to errors in the pose estimation of the cameras, as the calibration pattern might have not been aligned to the XY plane of the LCD frame due to the its non-planar shape. For this reason, typical values for screen cover glass ($\eta = 1.52$ and $d = 1mm$) are adopted for the simulations.

4.4 Measurements

In this section, we want to investigate the effects of the employed mathematical model of the LCD screen on the measurements. For this purpose, we will consider the measurement of a planar mirror using the setup in our laboratory, comparing the measurement errors in real measurements with simulations of different error sources, mainly focusing on the low-frequency errors introduced by the non-planar screen shape and refraction. Additionally, some tests will also consider high-frequency errors in the decoding process.

The test surface consists in a planar first surface mirror with a flatness of 4-6 λ, and it is used for both system calibration and measurements. In order to analyze the measurement errors, the angular error of the normals (angle of $\hat{\mathbf{n}}$ with respect to Z_w) and the distance of the triangulated points to the Z=0 plane will be considered.

The employed setup consists of a camera (the one used in the previous chapter), an LCD screen (the one characterized in the previous section), and a linear stage to move the screen to two positions. Their geometric arrangement is shown in Fig. 4.5. The camera observes a planar mirror at a distance of 400 mm and an angle of 17.25° with respect to the mirror's Z axis (principal ray), and the screen is placed approximately symmetrical with respect to the mirror's Z axis at distances of 256 mm and 379 mm respectively.

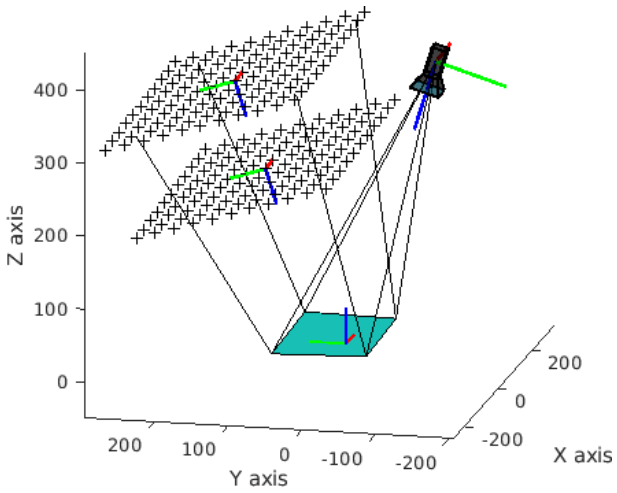

Figure 4.5: Scheme of the measurement setup. The involved reference frames are represented by the colored arrows, with red, blue, and green arrows indicating respectively the X, Y and Z axes.

With this setup, not all the camera pixels observe the pattern reflected on the surface, resulting in a effective field of view of 2560x1600 pixels, which translates into roughly 135 mm in the Y axis, and between 220 mm and 200 mm in the X axis (respectively at the top and bottom of the image). The employed pattern coding strategy consisted of horizontal and vertical sinusoidal pattern sequences with K=7 different spatial periods for the unwrapping, and N=5 phase-shifted patterns in each sequence. The exposure time of the camera was set to maximize its dynamic range while keeping a safety margin to avoid sensor saturation. The patterns are displayed on the LCD using the same intensity value for all color sub-pixels, no gamma correction was applied. This configuration is not optimal for avoiding high-frequency errors, but it provides a reference of the effects that one might encounter with a simple implementation.

4.4.1 Real Measurements

The measurements are performed following the procedures in Sec. 4.2. Two error maps are computed from the results, one for the angular error in the surface normals ($\epsilon_{\hat{\mathbf{n}}}$),*i.e.*, the angle with respect to the reflection plane, and the other one for the triangulation error ($\epsilon_{\mathbf{M}}$), *i.e.*, the error in $\mathbf{Z}_{\mathrm{w}}$ axis. These measures are defined respectively in Eqs. 4.24 and 4.25:

$$\boldsymbol{\epsilon}_{\hat{\mathbf{n}}}(u_{\mathrm{c}}, v_{\mathrm{c}}) = \mathrm{atan2}(||\hat{\mathbf{n}}(u_{\mathrm{c}}, v_{\mathrm{c}}) \times \mathbf{Z}_{\mathrm{w}}||, \hat{\mathbf{n}}(u_{\mathrm{c}}, v_{\mathrm{c}}) \cdot \mathbf{Z}_{\mathrm{w}}) \tag{4.24}$$

$$\boldsymbol{\epsilon}_{\mathbf{M}}(u_{\mathrm{c}}, v_{\mathrm{c}}) = M_z(u_{\mathrm{c}}, v_{\mathrm{c}}) \tag{4.25}$$

where $\hat{\mathbf{n}}(u_{\mathrm{c}}, v_{\mathrm{c}})$ and M_z represent the normal vector and the z component of triangulated point $\mathbf{M}$ measured at pixel$[u_{\mathrm{c}}, v_{\mathrm{c}}]$, and $\mathbf{Z}_{\mathrm{w}}$ corresponds to the Z axis of the reflection plane estimated in calibration, *i.e.*, mirror's normal.

The computed error maps $\epsilon_{\hat{\mathbf{n}}}$ and $\epsilon_{\mathbf{M}}$ exhibit a considerable amount of high-frequency variations, as can be seen in Figs. 4.6(a) and 4.6(b). These variations are most likely originated from the decoding process, and should be removed in order to analyze the low-frequency errors introduced by the non-planar shape and refractive layers of the screen.

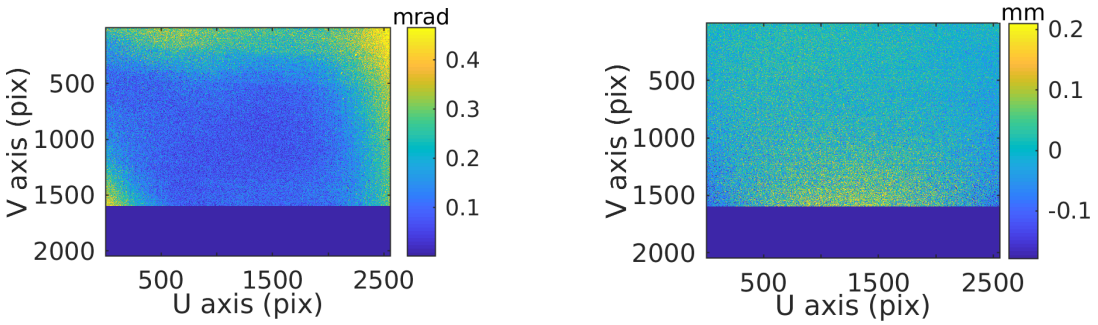

Figure 4.6: Error maps: (a) $\epsilon_{\hat{\mathbf{n}}}$ - Normal vector errors (b) $\epsilon_{\mathbf{M}}$ - Triangulated point errors.

In order to analyze high-frequency (ϵ^{h}) and low-frequency (ϵ^{l}) errors independently, a simple additive noise model is adopted, assuming that (i) low-frequency errors vary smoothly over the measured area, (ii) high-frequency errors are locally normally distributed, and (iii) the variation of their distribution is also smooth, such that

$$\boldsymbol{\epsilon}(u_{\mathrm{c}}, v_{\mathrm{c}}) = \epsilon^{\mathrm{l}}(u_{\mathrm{c}}, v_{\mathrm{c}}) + \epsilon^{\mathrm{h}}(u_{\mathrm{c}}, v_{\mathrm{c}}) \tag{4.26}$$

where $\epsilon^{\mathrm{h}} \approx \mathcal{N}(0, \sigma_\epsilon)$ corresponds to the high-frequency errors and $\sigma_\epsilon = \sigma_\epsilon(u_{\mathrm{c}}, v_{\mathrm{c}})$ represents its locally varying standard deviation.

Note that the purpose of this model is to provide an approximate estimation of the order of magnitude of high-frequency errors, a more detailed error model would be required to analyze the source of these errors. In this way, low-frequency errors $\epsilon^{\mathrm{l}}_{\hat{\mathbf{n}}}$ and $\epsilon^{\mathrm{l}}_{\mathbf{M}}$ can be isolated by applying a low pass Gaussian filter to each error map such that

$$\epsilon^{\mathrm{l}}_{\hat{\mathbf{n}}} = \mathbf{G} * \epsilon_{\hat{\mathbf{n}}} \qquad \text{and} \qquad \epsilon^{\mathrm{l}}_{\mathbf{M}} = \mathbf{G} * \epsilon_{\mathbf{M}} \tag{4.27}$$

where $\mathbf{G}*$ represents a convolution with a 9x9 Gaussian kernel with $\sigma = 2$.

As can be seen in Fig. 4.7, these filtered error maps show a smooth spatial structure that illustrates the variation of the low-frequency errors over the measured area. Figure

Fig. 4.7(a) shows that the estimated normal vector errors are around 0.1 mrad in the central part and get gradually larger at the borders, reaching up to 0.4 mrad in the upper right corner. Figure Fig. 4.7(b) shows that the triangulation errors are around $\pm 10\mu m$ in central part, with larger errors occurring at the bottom of the measured area, reaching >50 μm in the center and $<$-50 μm in the lateral edges.

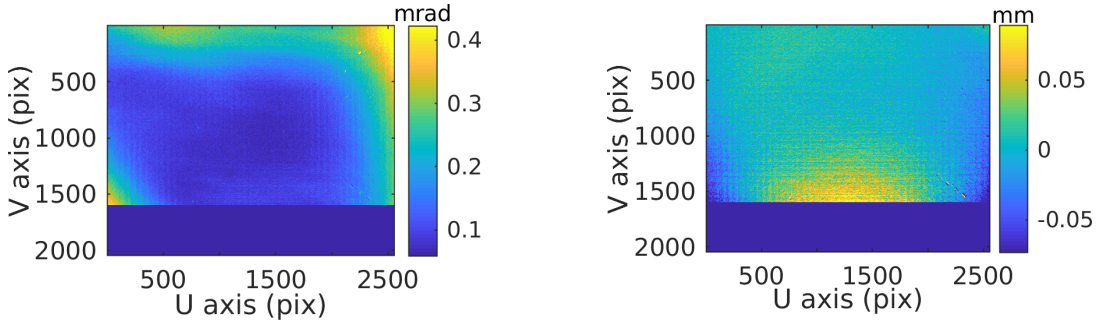

Figure 4.7: Error maps: (a) $\epsilon^{\mathrm{l}}_{\hat{\mathbf{n}}}$ - Normal vector errors (b) $\epsilon^{\mathrm{l}}_{\mathbf{M}}$ - Triangulated point errors.

In the absence of independent measurements, high-frequency errors can be estimated by subtracting the filtered error map to the original one:

$$\epsilon^{\mathrm{h}}_{\hat{\mathbf{n}}} = \epsilon_{\hat{\mathbf{n}}} - \epsilon^{\mathrm{l}}_{\hat{\mathbf{n}}} \qquad \text{and} \qquad \epsilon^{\mathrm{h}}_{\mathbf{M}} = \epsilon_{\mathbf{M}} - \epsilon^{\mathrm{l}}_{\mathbf{M}} \tag{4.28}$$

The local distribution of high-frequency errors is computed by fitting a normal distribution in 31x31 pixel neighborhoods, as shown in Fig. 4.8. The standard deviation is used

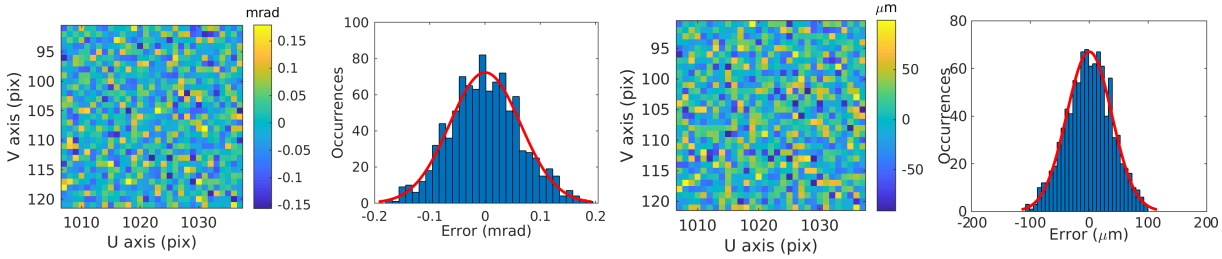

Figure 4.8: High-frequency errors estimated in a local 31x31 pixel patch: (a) normal vector errors $\epsilon^{\mathrm{h}}_{\hat{\mathbf{n}}}$ (b) local distribution of $\epsilon^{\mathrm{h}}_{\hat{\mathbf{n}}}$ (c) triangulated point errors $\epsilon^{\mathrm{h}}_{\mathbf{M}}$ (d) local distribution of $\epsilon^{\mathrm{h}}_{\mathbf{M}}$.

to characterize the magnitude and spatial variation of these errors, similar to the standard uncertainties used in the previous chapter. Fig. 4.7(a) shows that high-frequency errors of the normal vectors estimated in the center of the measured area are below 0.05 mrad, with the errors gradually increasing outwards, peaking around 0.08 mrad at the top edges. Similarly, Fig. 4.7(b) shows that the largest triangulation errors occur at the lower part of the measured area ($100\mu m$) and gradually decrease towards the top ($< 40\mu m$).

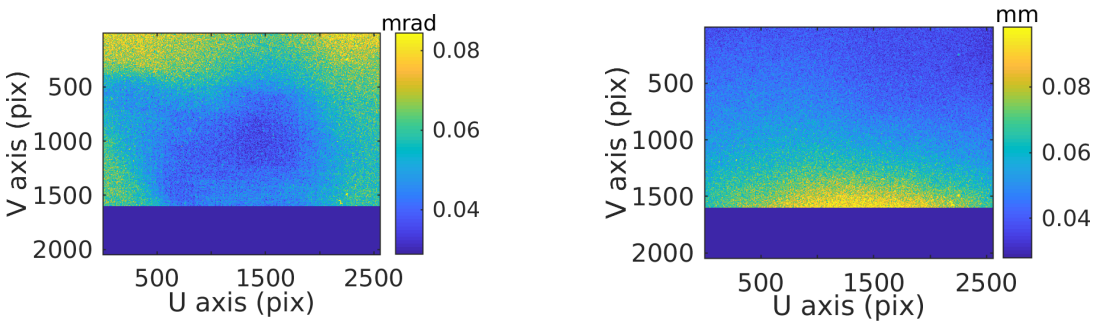

Figure 4.9: High-frequency error distribution: (a) normal vector errors $\sigma_{\epsilon_{\hat{\mathbf{n}}}}$ and (b) triangulated point errors $\sigma_{\epsilon_{\mathbf{M}}}$.

The results are summarized in Table 4.1, showing the mean and standard deviation of the estimated errors over the measured area.

Model	$\hat{\mathbf{n}}$ error (mrad)		Z error (μm)	
	mean	st. dev	mean	st. dev
Measured (ϵ)	0.143	0.087	8.7	58.9
Low-freq. (ϵ^1)	0.143	0.068	8.7	21.5
High-freq (σ_ϵ)	0.053	0.011	52.0	14.1

Table 4.1: Normal vector field and triangulated surface errors.

From the table one can conclude that in the case of the normal vector errors, low-frequency errors are higher and have larger variations over the sensor than high-frequency errors. In contrast, for triangulated points high-frequency errors are dominant.

4.4.2 Simulations

In order to investigate the source of the errors encountered in the measurements, a series of simulations are carried out for both the low and high-frequency errors. The simulations are based on ray tracing using the parameters of the real setup, with the measured non-planar screen shape, and assuming a single layer refractive model with 1 mm thickness and a refraction index of 1.52. The simulations include both, the calibration and the measurement process.

4.4.2.1 Simulated Low-Frequency Errors

Three models are considered in order to have an estimate of the magnitude and spatial distribution of the low-frequency errors introduced by the screen model: The first one only considers the errors generated by the non planar shape of the screen ($\delta_{refract} = 0$), the second one only the ones arising from the refractive layer ($\delta_{shape} = 0$), and the last one combines both. In order to isolate the low-frequency errors from high-frequency errors, all the simulations are performed without any noise. The resulting error maps, $\epsilon_{\hat{\mathbf{n}}}$ and $\epsilon_{\mathbf{M}}$, are shown respectively in Figs. 4.11 and 4.10

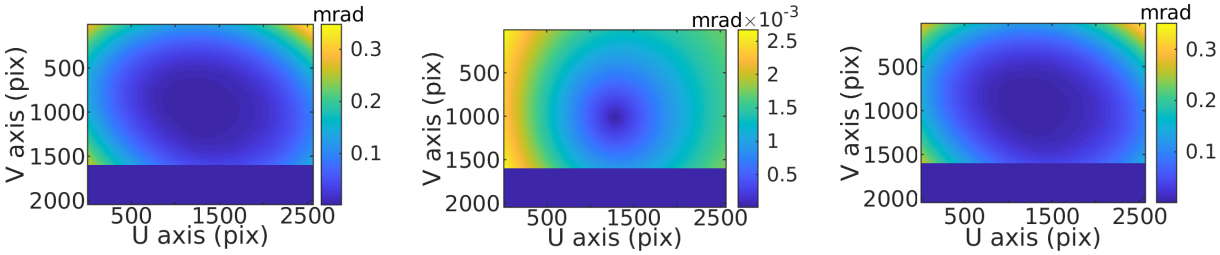

Figure 4.10: Simulated low-frequency error in normal vectors: (a) effects of non-planar screen shape, (b) effects of refractive layers in triangulation, (c) combined effects.

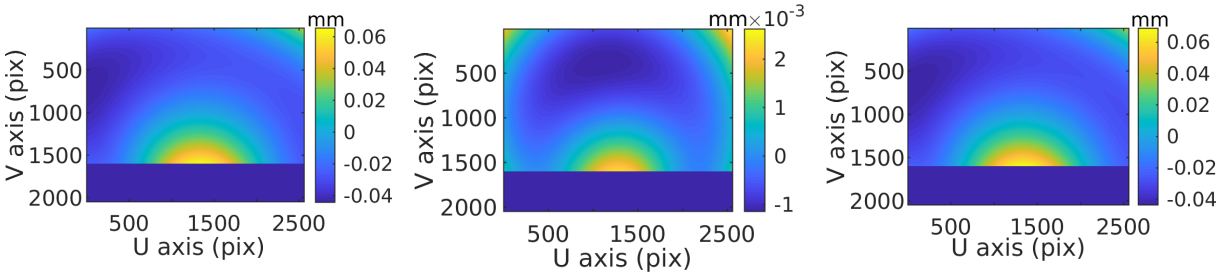

Figure 4.11: Simulated low-frequency error in triangulated points: (a) effects of non-planar screen shape, (b) effects of refractive layers in triangulation, (c) combined effects.

As can be seen in Table 4.2, the simulations indicate that the errors originated by the non-planarity of the screen are substantially larger than the ones due to the refractive layers.

Model	$\hat{\mathbf{n}}$ error (mrad)		Z error (μm)	
	$\mu_{\hat{\mathbf{n}}}$	$\sigma_{\hat{\mathbf{n}}}$	μ_z	σ_z
Screen shape error	0.069	0.060	-10.35	18.09
Screen refraction error	0.001	0.000	0.03	0.72
Screen shape & refraction error	0.070	0.061	-9.99	19.30

Table 4.2: Simulation of low-frequency errors arising from the non-planar shape and refractive layers of the LCD screen.

In order to compare the results from the simulations and the real measurements, the obtained statistics are summarized in Table 4.3, which also shows the uncertainties arising from the camera calibration obtained in the previous chapter.

- The average normal vector error in the real measurements doubles that obtained in the simulations of the screen errors. However, the errors obtained in the simulations are in the same order of magnitude as the uncertainties resulting from the camera calibration, and their sum is close to the error in the real measurements. This suggests that it is the combination of the screen and camera errors that causes the low-frequency errors.

- The average error in the z coordinate of the triangulated points has a similar magnitude in the simulations of the screen errors and in the real measurements, but

they differ in the sign. In this case, the uncertainties resulting from the camera calibration are much higher than the errors obtained in the simulations, and their sum does not agree with the errors obtained in the real measurements.

- The standard deviation of the measurement errors shows that their variation over the measured area is similar in the real measurements and the simulations of the screen errors, for both the normal vector field and the triangulated surface. The structure arising from the spatial distribution of the errors also seems to be in agreement, as can be seen by comparing Figs. 4.10(c) and 4.11(c) with Figs. 4.7 (a)-(b). Note that this variation was much smaller for the uncertainties arising from the camera calibration, which indicates that the variations of the errors over the sensor are mainly caused by the screen errors.

Model	$\hat{\mathbf{n}}$ error (mrad)		Z error (μm)	
	$\mu_{\hat{\mathbf{n}}}$	$\sigma_{\hat{\mathbf{n}}}$	μ_z	σ_z
Low freq. error in real measurement	0.143	0.068	8.7	21.5
Simulated screen shape & refraction error	0.070	0.061	-10.0	19.3
Camera calibration (standard uncertainty)	0.068	0.001	48.0	6.0

Table 4.3: Comparison of low-frequency errors

Thus, although the low-frequency errors in the real measurements seem to coincide reasonably with those obtained in the simulations, there are some differences that we could not explain. These disagreements can be attributed to several factors such as the non-perfectly planar mirror used for calibration or the dependence of the screen shape on mechanical stress and gravity, as its position during the measurements was different from the one corresponding to the characterization.

4.4.2.2 Simulated High-Frequency Errors

In order to investigate the source of high-frequency errors, the decoding noise is considered. For this purpose, a Monte Carlo approach is followed by repeating the simulation corresponding to the complete model (shape + refraction) with noise added to the screen coordinates obtained from the ray tracing. This allows us to study the combined effects of high and low-frequency errors by observing the variations of the measurements in each camera pixel.

The decoding noise is assumed to be statistically independent, zero mean, and normally distributed, such that every observed screen location $\bar{\mathbf{q}}^p$ is corrupted with an error $\delta_{\mathbf{q}^p} \approx \mathcal{N}(0, \sigma_{\mathbf{q}^p})$. Different noise levels are simulated with $\sigma_{\mathbf{q}^p}$ ranging from 0.05 to 0.40 pixels. The calibration and measurement procedure is repeated 1000 times for each noise level, computing the error maps $\epsilon_{\hat{\mathbf{n}}i}$ and $\epsilon_{\mathbf{M}i}$ at each iteration. From there, the measurement errors arising from the decoding noise are estimated by considering the standard deviation of the errors for each camera pixel $\mathbf{m}$, such that the spatially variable distribution of the high-frequency errors in the normal vectors and triangulated points is described respectively by $\sigma_{\epsilon_{\hat{\mathbf{n}}}}(\mathbf{m})$ and $\sigma_{\epsilon_{\mathbf{M}}}(\mathbf{m})$, as shown in Fig. 4.12. Figure 4.13 summarizes the results of the simulations, showing the average and standard deviation of $\sigma_{\epsilon_{\hat{\mathbf{n}}}}$ and $\sigma_{\epsilon_{\mathbf{M}}}$ over the whole sensor for each noise level.

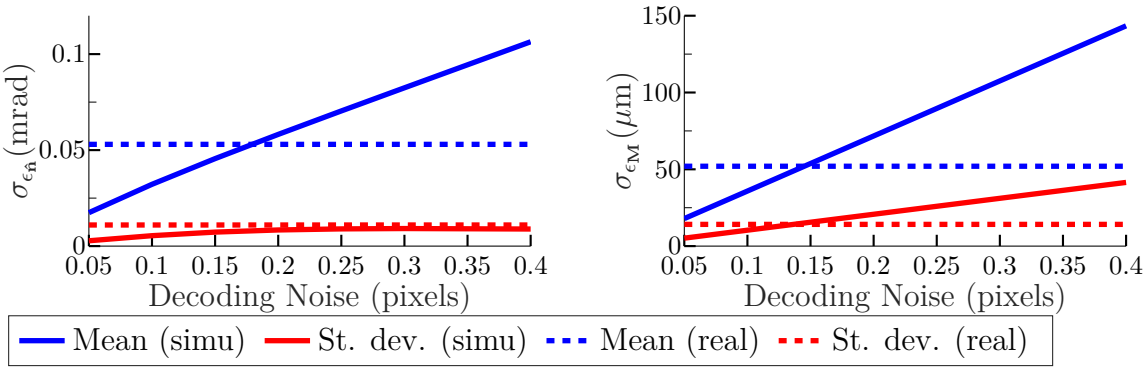

Figure 4.12: Simulated high-frequency errors in (a) normal vectors and (b) triangulated points. The solid lines correspond to the mean and standard deviation of the errors $\sigma_{\epsilon_{\hat{\mathbf{n}}}}$ and $\sigma_{\epsilon_{\mathbf{M}}}$ over the sensor, the dashed ones represent the values estimated from the real measurements.

From the previous graphs, it can be seen that the real measurements exhibit similar levels of high-frequency errors as the simulations corresponding to a decoding noise with a standard deviation around 0.15 pix. The following figure shows the distribution of the errors obtained for this noise level. Note that the spatial structure of these errors reasonably approximates the ones estimated for the real case, as can be seen by comparing Figs. 4.13 (a) and 4.13(b) with Figs. 4.9 (a) and 4.9(b).

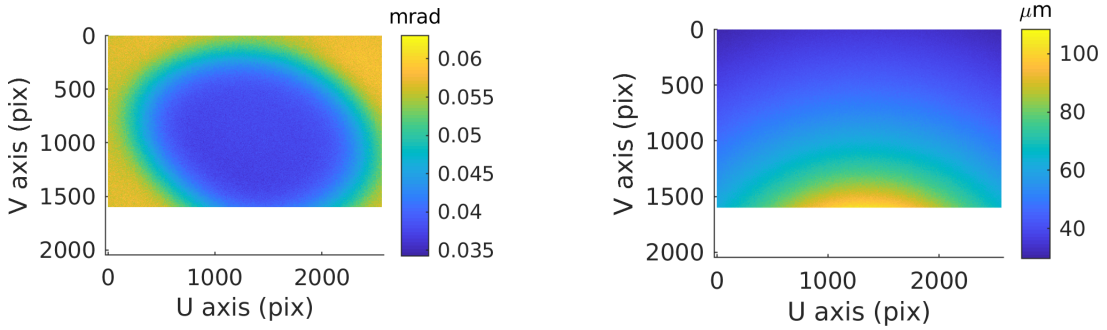

Figure 4.13: Spatial distribution of Simulated high-frequency errors arising from decoding noise of 0.15 pix, respectively in (a) normal vectors, $\sigma_{\epsilon_{\hat{\mathbf{n}}}}$, and (b) triangulated points $\sigma_{\epsilon_{\mathbf{M}}}$.

4.5 Chapter Conclusions

In this chapter, the effects of several non-ideal characteristics of the LCD in deflectometric measurements were considered. The analyzed sources of error have been divided into low-frequency and high-frequency errors. The former included the non-planarity of the LCD screen and the refraction in its translucent layers, while the latter correspond to decoding errors. In order to investigate their effects in the measurements, a real setup was characterized and used to measure the normal vector field as well as the triangulated surface of a planar first surface mirror. Further simulations were performed with this setup in order to isolate the error sources and compare the results.

The experiments show a reasonable agreement between the errors obtained in the real

and simulated measurements, both in the magnitudes and their spatial distribution over the field of view. The following list outlines the main findings:

- Regarding the low-frequency errors, the simulations indicate that the ones originated by the non-planarity of the screen are substantially larger than the ones due to the refractive layers.

- The average normal vector error in the real measurements doubles that obtained in the simulations of the screen errors. However, the errors obtained in the simulations are in the same order of magnitude as the uncertainties resulting from the camera calibration, and their sum is close to the error in the real measurements, suggesting that it is the combination of the screen and camera errors what causes the low-frequency errors.

- The average error in the z coordinate of the triangulated points has a similar magnitude in the simulations of the screen errors and in the real measurements. In this case, the uncertainties resulting from the camera calibration were much higher than the errors obtained in the simulations for the non-planar shape of the LCD.

- The standard deviation of the measurement errors shows that their variation over the measured area is similar in the real measurements and the simulations of the screen errors, for both the normal vector field and the triangulated surface. The structure arising from the spatial distribution of the errors over the field of view also showed a qualitative agreement. Note that the variability was much smaller for the uncertainties arising from the camera calibration, indicating that the spatial variations are mainly caused by the screen errors.

- Regarding the high-frequency errors, the results indicate that the triangulated surface points are more sensitive to these errors than the estimated normals, which shows why most practitioners prefer to obtain the surface shape by integrating the normals instead of using the triangulated points.

Taking into account that all the experiments have been carried out with a planar surface positioned at the calibration position, we expect larger errors when measuring non-planar shaped surfaces or planar but in different position and/or orientation. In order to generalize these findings, further experiments should be performed. In any case, several measures should be taken in order to improve the accuracy of the measurements.

First, it has been shown that the low-frequency errors attributed to the non-planar shape of the LCD are not negligible. Thus, in order to obtain more accurate measurements, the conversion from pattern coordinates in LCD pixels to metric 3D coordinates should consider at least the non-planar shape of the screen, both in the measurements and in the calibration. Alternatively, in measurements with smaller fields of view, the employed screen area could be restricted to the regions with smallest deviations from planarity.

Second, in order to reduce the high-frequency errors, more than 5 phase-shifts should be used in order to reduce the decoding uncertainty. This has already be applied in the experiments performed in the next chapter.

Chapter 5

Surface Imperfection Detection

5.1 Introduction

In many industrial sectors, surface quality is important not only for operative or functional reasons, but also because aesthetic imperfections can affect consumers' perception of the overall quality of a product. Manufacturers from automotive, home appliance or consumer electronic sectors, among others, are required to perform a surface inspection to 100 % of their production to ensure that quality standards are met. This inspection typically entails the detection of local imperfections in the shape and/or texture of the surfaces. The former correspond to geometrical deformations, such as bumps, pits, or dents, whereas the latter include surface finish defects, e.g. scratches, stains, or foreign matter inclusions generated during the manufacturing process.

Local imperfections in surfaces with a glossy or shiny finish are particularly susceptible to being noticed by a human observer. These surfaces reflect at least part of the light in a specular way, which results in a mirror-like appearance where the environment is superimposed on the observed surface. In some cases, the defects themselves are not actually seen, but they are noticed because of optical effects produced by reflections of the surrounding scene, for example, in the form of highlights or *spikes* resulting from small surface deformations. In other cases, the surface finish is altered, resulting in variations of reflectance that change the apparent texture of the surface, drawing the attention of the observer. It has been found that even micrometer scale defects can be perceived by the naked eye [PPD+13, Zie15].

Previous chapters have shown the capabilities of deflectometric techniques for quantitative shape measurement of specular surfaces. This approach is appropriate for dimensional control of precision surfaces such as solar concentrators [BLH+13], telescope mirrors [BLH+13, HFOE13], or lenses [HFOE13]. However, absolute measurements are not always the most appropriate source of information for detecting local surface imperfections. In more general reflective surfaces, where manufacturing precision is not as high, local defects might meet overall dimensional tolerances, and reducing these tolerances can result in objects with acceptable shape variations being rejected. Furthermore, it has been shown that the achievable uncertainty of quantitative measurements is limited by the acquisition system calibration [HFOE13]. For these reasons, more qualitative measurements that exploit the local sensitivity of deflectometric recordings have been proposed for local imperfection detection: Local curvatures can be directly obtained from the deflectometric mapping [KL08, Bur16], providing valuable information for local imperfection detection.

In addition, it has been demonstrated that photometric information from the captured patterns can also be used to identify texture defects [PLK06, NFK13].

Independent of the employed approach, exploiting deflectometric data for automatic defect detection requires further processing that can involve a considerable engineering effort. Furthermore, aesthetic defect specifications are difficult to define because they are related to what a human can perceive [Zie15], and are therefore inherently subjective, which increases the complexity of the involved algorithms. For these reasons, pattern recognition algorithms based on shallow learning techniques have been typically employed for automatic defect classification, *e.g.*, Support Vector Machines or Naive Bayes classifiers [Cau10, Zie14, KSM17]. These algorithms allow defect specifications/thresholds to be learned by example, but they need to be fed with features that allow defects to be discriminated. These features have to be extracted from deflectometric recordings, requiring a time-consuming trial and error process for finding effective/meaningful descriptors and tuning the involved preprocessing operations. Moreover, this feature engineering work is usually domain-specific, and *intuition, creativity, and "black art" are as important as the technical stuff* [Dom12], which in many cases results in ad hoc solutions that are not easily adapted to different objects.

Deep learning (DL) methods leverage the feature engineering complexity, by learning the feature extraction itself during the training process [GBC16]. This approach has been used for multiple purposes such as natural language processing [CW08], image classification [KSH12], speech recognition [HDY+12], or automatic image captioning [XBK+15]. Among the DL approaches, Convolutional Neural Networks (CNN) are particularly popular in computer vision applications, due to their ability to automatically learn meaningful features from raw image data. CNN's notoriety was boosted by their remarkable results in classification competitions [KSH12], and have shown to work well in different domains, *e.g.*, pedestrian detection [AKV+15], human pose estimation [CSWS17], or medical image analysis [CGGS12, Vás17]. The application of these methods to surface inspection has been studied by many researchers, *e.g.*, [MMC+12, SHM14, FRHN+16, WSRS16, YWG17, DCB+18, RHT18, YNDP18, TZM+18, PAK19, MXZ19, LZZ19], but it has not yet been used for defect detection in deflectometric recordings.

In this chapter, we aim to explore the use of a deep learning based approach for the automated defect identification in deflectometric data. The rest of the chapter is organized as follows: A brief review of related works is performed in Sec. 5.2, afterward the contributions of the chapter are stated in Sec. 5.3. Sec. 5.4 focuses on the data acquisition, whereas Sec. 5.5 and Sec. 5.6 are devoted respectively to the detection and segmentation of surface defects. The chapter is ended with the conclusions in Sec. 5.7.

5.2 Related Works

With the aim of putting the contributions of the chapter into context, this section briefly reviews several works related to the detection of surface imperfections using deflectometry (Sec. 5.2.1), and deep learning approaches employed with non-deflectometric inspection techniques (Sec. 5.2.2).

5.2.1 Detection of Surface Imperfections Using Deflectometry

Early work by Seulin et al. [SMG01] used qualitative information obtained from a lighting tunnel displaying binary patterns for specular surface inspection. The lighting structure is moved along the main object axis to capture a sequence of images such that defects appear highlighted in the mean image of the sequence. Although the system enables a fast detection of geometric surface defects in planar surfaces, it will not suffice for more complex surfaces (non-flat, holes, edges...). This has been shown in Refs. [ATHA11, MSAT17], where a similar lighting tunnel idea was used for the inspection of painted car bodies. This application required an ad hoc designed system, with the location and movements of multiple cameras and lighting sources optimized for the particular application, and more elaborated processing of the images, including image fusion techniques and post-processing algorithms to identify the defective regions.

Afterward, most approaches have relied on setups composed of a single camera and an LCD screen, avoiding the displacement of the light source by using phase-shifted sinusoidal patterns. In particular, several works have shown that shape information extracted from the captured patterns can be exploited in applications addressing geometric defects.

Surrel et al. [Sur04] proposed to use phase modulations induced by surface deformations originated in bending tests for material characterization. In their method, repeated measurements are performed while the surface is under load stress, such that the relative phase displacements of the acquired patterns are directly related to changes in surface slope, thereby enabling the estimation of curvatures in quasi-plane surfaces. Kammel et al. [KL08] showed that local curvatures can be directly estimated from the spatial variations in the decoded lightmap, proposing a defect detection strategy based on a comparison against the local curvatures of a reference surface. Ziebarth et al. [Zie14] worked on the identification of dents and pimples in lacquered metal sheets. They tackled a multi-scale analysis of height-maps obtained from a deflectometric setup, considering several methods to extract geometric features and different shallow learning algorithms to obtain a pixel-wise classification of the measurements. The feature extraction based on biorthogonal wavelet filter banks designed to match specific defect classes gave them good results [LZGH16]. Their approach facilitates the segmentation of defective regions without using reference surfaces. However, its adaptation to more complex surfaces and a wider range of defects has not been published. Furthermore, obtaining accurate absolute measurements requires a precise system calibration, and for nonplanar surfaces, also more complex setups[BLH$^+$13, HFOE13]. Kofler et al. [KSM17] also employed local curvatures, in this case for the detection of geometric imperfections in planar silicon wafers. They explored the use of different machine learning algorithms to evaluate defect candidate regions from a feature vector containing several statistical properties of the local curvatures. This method requires defect candidate regions to be segmented in advance, which in their experiments was done by simple thresholding, but more complex surfaces will require more elaborated segmentation strategies.

In addition, other works dealing with textural defects have made use of photometric information extracted from sinusoidal patterns captured with these simple setups, *e.g.*, Puente León et al. [PLK06] and Nagato et al. [NFK13] showed that the contrast between black and white fringes is particularly useful. Similarly, Petz et al. [PFT17] showed that local illumination and visibility of the modulated light source can be used to detect defects smaller than the resolution of the employed camera system.

When both shape and texture defects need to be considered, using only geometric

or photometric data might not suffice, and other works have considered mixing features from both the shape of the surface and photometric information from the captured patterns. Caulier et al.[Cau10] worked on the inspection of metallic surfaces considering a series of 3D defects (surface shape deformations) as well as 2D defects (surface roughness changes). In their method, different features were extracted from raw images of the reflected patterns, showing that the combination of spectral, structural, and brightness features enabled the detection of a wide range of defects. An important part of their work relies on an optimal feature selection and classification approach for the specific task.

Many of these works deal with rather simple surfaces, which have a continuous and many times flat shape. These conditions will not be met in many industrial inspection applications, requiring further processing of the measurements to cope with edges, holes, varying curvatures, etc. Deep learning algorithms seem an interesting approach to cope with these challenges, but to the best of our knowledge, this approach has not been proposed for defect detection in deflectometric measurements, hence the motivation of this chapter. Furthermore, the combination of geometric and photometric data obtained with a simple deflectometric setup seems a good starting point for a generic specular surface inspection system based on deep learning methods.

5.2.2 Deep Learning for Detection of Surface Imperfections

Regarding the surface inspection scenario with non-deflectometric imaging methods, various researchers have applied convolutional neural network (CNN) based approaches for automatic defect detection.

The basic approach consists in using CNN architectures to extract features from raw pixels in order to perform a sample-wise classification, *i.e.*, predicting if any defect is present on the processed image or image patch. Some implementations have been carried out using exclusively deep learning techniques, by using standard fully connected (FC) Multi-Layer Perceptron (MLP) to approximate the posterior class probabilities, while others have proposed to combine them with other techniques for the final classification, *e.g.*, one or more linear SVM classifiers [MANC17, NHVC17, TSU+17].

Examples of defect detection methods based exclusively on CNN's can be found in different domains such as defect classification in steel surfaces using grayscale images [MMC+12, ZCZ+17], detection of rail surface defects using photometric stereo images [SHM14] or grayscale images [FRHN+16], classification of defective photovoltaic cells using electroluminescence images [DCB+18], or welding defects in engine transmission using RGB images [PAK19]. Other examples of industrial CNN implementations use grayscale images for surface inspection of storm-water pipe systems [THTB18], inspection of resin molded surfaces [NTO+18], or blister detection in polymer lithium-ion batteries [MXZ19].

Many of these methods were compared to shallow learning techniques using classical features, showing that CNN architectures result in higher inspection accuracies *e.g.*, [MMC+12, WSRS16, MXZ19]. However, they do not provide any information on the location, shape or size of the defects. In order to overcome this shortcoming, different strategies have been proposed to perform a defect segmentation with these architectures. The most common one consists in dividing the input images into patches using a sliding window, such that each patch is individually classified, and merging the results in a confidence-map or heatmap that is afterward processed using classical segmentation

methods. This approach has been proposed for defect segmentation in different applications, such as textured surfaces [WSRS16, WCQS18] or solar cells [BEAA19], among others. Ren et al. [RHT18] demonstrated the re-usability of this approach, showing that the same CNN architecture could be successfully applied for defect segmentation in different applications including grayscale images of hot-rolled steel strips, grayscale images of wooden surfaces, X-ray images of weld defects, and microscope images of titanium fan blades.

In recent years, several researchers have focused on the semantic segmentation of images [GLGL18] in other fields that typically require localization information, and more advanced neural network architectures have been proposed in order to enable a pixel-wise classification of images. Specifically, Fully Convolutional Networks (FCN) [LSD15] and similar architectures have gained popularity in fields such as medical imaging [RFB15] or autonomous driving [BKC17]. Different implementations of FCNs have been used to perform semantic segmentation in industrial inspection applications: Yu et al. [YWG17] combined two FCNs for defect candidate segmentation and detection in textured surfaces. Li et al. [LZZ19] used an FCN for structural damage detection in concrete structures and Yudin et al. [YNDP18] used it for roof defects recognition on aerial photographs. Stavroulakis et al. [SDS+18] used Segnet [BKC17] to identify different materials in an image such that specular surfaces and scattering surfaces could be identified and afterward separately processed. Tao et al. [TZM+18] proposed a cascaded autoencoder architecture (CASAE) to segment defect candidates that were afterward classified using a compact CNN.

In short, most works related to surface inspection using deep learning in machine vision applications can be divided into classification oriented approaches and segmentation based approaches, and to the author's knowledge, none of them has been tested for defect detection or segmentation in deflectometric data.

5.3 Contributions

The goal of this chapter is to investigate if these DL concepts can be adapted for industrial inspection of reflective surfaces. The main hypothesis is that DL architectures will learn relevant features from a combination of photometric and geometric information derived from deflectometric recordings, enabling the detection of a wide range of shape and texture defects with a reduced feature engineering effort. For this purpose two different approaches to exploit the deflectometric recordings of industrial surfaces are explored:

- The first approach consists in a method for automated defect classification, where a CNN architecture is used to extract relevant features in order to directly classify defective samples (a whole measurement or a measurement patch). The idea was presented at the 13th International Conference on Soft Computing Models in Industrial and Environmental Applications [MWBEAA19a].

- The second approach consists in a method for automated segmentation of surface defects, where an FCN architecture is used to perform a pixel-wise classification of the measurements. The idea was presented at the 15th international biannual conference on Quality Control by Artificial Vision [MWBEAA19b], receiving the Best Paper Award.

The conceptual diagram shown in Fig. 5.1 overviews the three main blocks involved in this chapter. The first one tackles the deflectometric registration procedure (Sec. 5.4), including the acquisition and processing of the fringe patterns, as well as the pre-processing operations required to feed these measurements to a DL architecture. The other two blocks correspond to the proposed classification and segmentation approaches, which are undertaken respectively in Sec. 5.5 and Sec. 5.6, where each of them is described and afterward tested in an industrial case study.

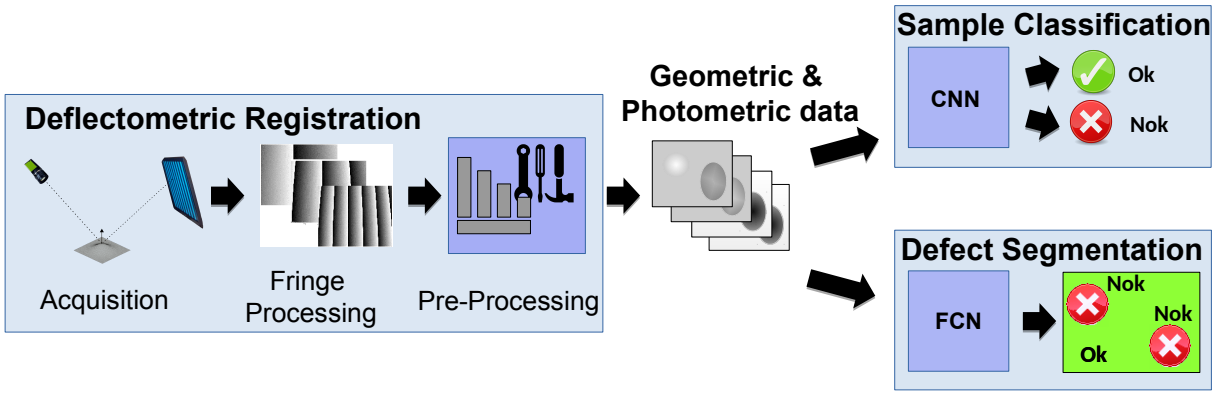

Figure 5.1: Overview of the proposed approaches

5.4 Deflectometric Registration

This section describes the theoretical basis of the deflectometric registration process, including the acquisition and processing of the fringe patterns and the pre-processing of the obtained measurement. The resulting geometric and photometric information will be used to detect/segment shape and texture defects on the surface under test (SUT) later in Sec. 5.5 and Sec. 5.6.

5.4.1 Data Acquisition

The data acquisition system employed in this work is based on a simple deflectometric setup composed of an LCD screen and a single grayscale camera, without any moving parts or external measurement sources. The camera is focused on the SUT, and the LCD is placed such that the SUT reflects the patterns towards the camera, as shown in Fig. 5.2(a).

As explained in Sec. 4.2.1, the location of every screen point is encoded in the phase of two sequences of sinusoidal patterns: vertical fringes encoding the u axis, and horizontal ones for the v axis. Each sequence is composed of $N \geq 4$ phase-shifted patterns that are sequentially displayed to form a time series coding. The intensity measured at the camera pixel $\mathbf{p}_c = [u_c, v_c]$ observing a point of the n^{th} pattern in the sequence ξ is described by [SB06]:

$$I_n(\mathbf{p}_c) = I_{avg}(\mathbf{p}_c) + I_{pv}(\mathbf{p}_c)\cos[\phi_\xi + n\delta_\Phi], \qquad \delta_\phi = 2\pi/N, \qquad n \in \{0, 1, .., N-1\} \quad (5.1)$$

where I_n represents the image intensity at $\mathbf{p}_c$, and I_{avg} and I_{pv} correspond to the average intensity and half the peak-to-valley intensity modulation at $\mathbf{p}_c$ over the sequence, as

shown in Fig. 5.2(b). ϕ_ξ corresponds to the phase of the observed pattern point encoding the coordinate $\xi \in \{u, v\}$, and $n\delta_\Phi$ is the phase-shift for the n^{th} pattern.

Thus, the imaging function is described by three variables that depend on various properties of the surface: ϕ_ξ, I_{avg}, and I_{pv}. As will be explained in the next subsection, ϕ_ξ can be used to extract geometric information, while I_{avg} and I_{pv} provide photometric information related to the reflectance properties of the surface.

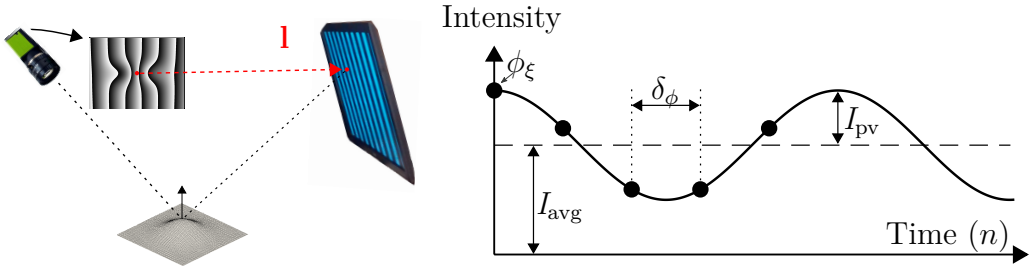

Figure 5.2: (a) Deflectometric recording setup: The camera observes patterns through their reflection on the surface, obtaining a mapping $\mathbf{l}$ from every camera pixel to the observed screen location. (b) Intensity variation for camera pixel $\mathbf{p}_\text{c}$, with black dots representing the recorded intensities for the n^{th} pattern in a sequence of $N = 5$.

5.4.2 Fringe Processing

For each camera pixel, the phases (ϕ_u,ϕ_v) encoding the observed LCD coordinates are extracted from the captured sequences using a phase-shifting [SB06] algorithm. However, the decoded values will be limited to $\phi_\xi \in \{0, 2\pi\}$; thus, in order to recover the absolute phase, a temporal unwrapping [SH97] approach is adopted using additional pattern sequences. The spatial period of the patterns in the first sequence (L_1) is at least equal to the screen width, such that coarse phase values are obtained without ambiguities. In each subsequent sequence, the period of the patterns is reduced to $L_k = 0.5L_{k-1}$ for the k^{th} sequence, removing the ambiguities using the coarse phases from the previous sequence, and obtaining more precise phase estimates in each iteration.

The unwrapped phases (Φ_u, Φ_v) provide a mapping between each image pixel $\mathbf{p}_\text{c}$, and the corresponding observed LCD screen location $\mathbf{p}_\text{l} = [u_\text{l}, v_\text{l}]$, as shown in eq. 5.2. This mapping is called a light-map ($\mathbf{l}$):

$$\mathbf{l} : \mathbf{p}_\text{c} \rightarrow \mathbf{p}_\text{l}, \quad \mathbf{l}(\mathbf{p}_\text{c}) = \frac{L_K}{2\pi}[\Phi_\text{u}(\mathbf{p}_\text{c}), \Phi_\text{v}(\mathbf{p}_\text{c})] = \mathbf{p}_\text{l} \tag{5.2}$$

The light-map itself is directly related to surface slopes, and therefore, its partial derivatives are proportional to local curvatures in orthogonal directions [KL08, Bur16], respectively $\tilde{C}_\text{x}$ and $\tilde{C}_\text{y}$ shown in 5.3. In practice, their computation boils down to a filtering operation which removes the constant component of the light-map, which can be addressed by subtracting a low-pass filtered light-map, as shown in 5.3.

$$\tilde{C}_\text{x}(\mathbf{p}_\text{c}) = \frac{\partial \mathbf{l}(\mathbf{p}_\text{c})}{\partial u_\text{c}} = \mathbf{l}_\text{x}(\mathbf{p}_\text{c}) - \mathbf{l}^*_\text{x}(\mathbf{p}_\text{c}), \qquad \tilde{C}_\text{y}(\mathbf{p}_\text{c}) = \frac{\partial \mathbf{l}(\mathbf{p}_\text{c})}{\partial v_\text{c}} = \mathbf{l}_\text{y}(\mathbf{p}_\text{c}) - \mathbf{l}^*_\text{y}(\mathbf{p}_\text{c}) \tag{5.3}$$

where $\mathbf{l}_\text{x}$ and $\mathbf{l}_\text{y}$ are the x and y components of the light-map $\mathbf{l}$, and $\mathbf{l}^*_\text{x}$ and $\mathbf{l}^*_\text{y}$ are the corresponding low-pass filtered light-map components.

Note that $\tilde{C}_{\mathrm{x}}$ and $\tilde{C}_{\mathrm{y}}$ are proportional quantities, computing the exact curvatures would require absolute measurements, which would be far more sensitive to calibration errors and uncertainty. In contrast, $\tilde{C}_{\mathrm{x}}$ and $\tilde{C}_{\mathrm{y}}$ are capable of capturing discontinuities in the surface gradient, providing valuable geometric information for local shape defect identification using a simple setup.

However, textural or appearance defects cannot always be detected by analyzing curvature information, because they are more related to the surface finish and the way in which light is reflected, and photometric information of the patterns might contain more meaningful features. Variations in the microstructure and reflectance of the surface will cause blurring and loss of contrast in the recorded patterns, modifying the peak-to-valley intensities (I_{pv}) and average intensities (I_{avg}). Data modulation ($\boldsymbol{\gamma}_{\xi}$) is a measure of the signal-to-noise ratio of the acquired pattern sequence, which can be used to extract features for this kind of defects. It is defined as the ratio between (I_{pv}) and (I_{avg}), and can be computed from the recorded intensity values [SB06] for every pixel $\mathbf{p}_{\mathrm{c}}$:

$$\gamma_{\xi}(\mathbf{p}_{\mathrm{c}}) = \frac{I_{\mathrm{pv}}(\mathbf{p}_{\mathrm{c}})}{I_{\mathrm{avg}}(\mathbf{p}_{\mathrm{c}})} = 2\sqrt{\left(\sum_{i=1}^{N} I_i(\mathbf{p}_{\mathrm{c}})\cos(\delta_i)\right)^2 + \left(\sum_{i=1}^{N} I_i(\mathbf{p}_{\mathrm{c}})\sin(\delta_i)\right)^2} \Bigg/ \left(\sum_{i=1}^{N} I_i(\mathbf{p}_{\mathrm{c}})\right) \tag{5.4}$$

Note that data modulation is computed from the pattern sequence corresponding to a direction $\xi \in \{x, y\}$. When the observed surface patch has directional variations, γ_x and γ_y might differ for a given pixel, but most of the time γ_x and γ_y will be similar. Data modulation modulus γ can be used to reduce redundant data:

$$\gamma(\mathbf{p}_{\mathrm{c}}) = \sqrt{(\gamma_x(\mathbf{p}_{\mathrm{c}}))^2 + (\gamma_y(\mathbf{p}_{\mathrm{c}}))^2} \tag{5.5}$$

Thus, $\tilde{C}_{\mathrm{x}}$ and $\tilde{C}_{\mathrm{y}}$ provide information on the surface geometry, and γ_x and γ_y contain photometric information that depends on surface finish. Our hypothesis is that their combination should provide enough features to identify a wide range of local geometrical imperfections and texture defects, as shown in Fig. 5.3.

5.4.3 Data Pre-Processing

For each measurement, the obtained data matrices can be stacked together to form a multi-channel or N-channel image, *e.g.*, 3-Channels $\left(\tilde{C}_{\mathrm{x}}, \tilde{C}_{\mathrm{y}}, \gamma\right)$ or 4-channels $\left(\tilde{C}_{\mathrm{x}}, \tilde{C}_{\mathrm{y}}, \gamma_x, \gamma_y\right)$, and be directly fed to a DL architecture. However, some preprocessing operations might be required or desirable to enhance the performance of the algorithms, the following ones are the most obvious ones:

- **Normalization:** In this work instead of using 'conventional" grayscale or color images, a combination of measurements of different variables is considered, therefore the channels in such N-channel image will have different scales and ranges, which can affect the training convergence [LBOM98]. For this reason, the measurement data should be normalized such that the distribution of the values of each channel is roughly in the same range. Typically this issue is solved via min-max normalization or standardization (Z-score) [Cho17], however, depending on the nature of the data, other approaches that are more robust statistically might offer better performance, *e.g.*, interquartile range normalization or median normalization. In practice, this operation can be considered as an hyperparameter, selecting the procedure that gives the best empirical results at the training stage.

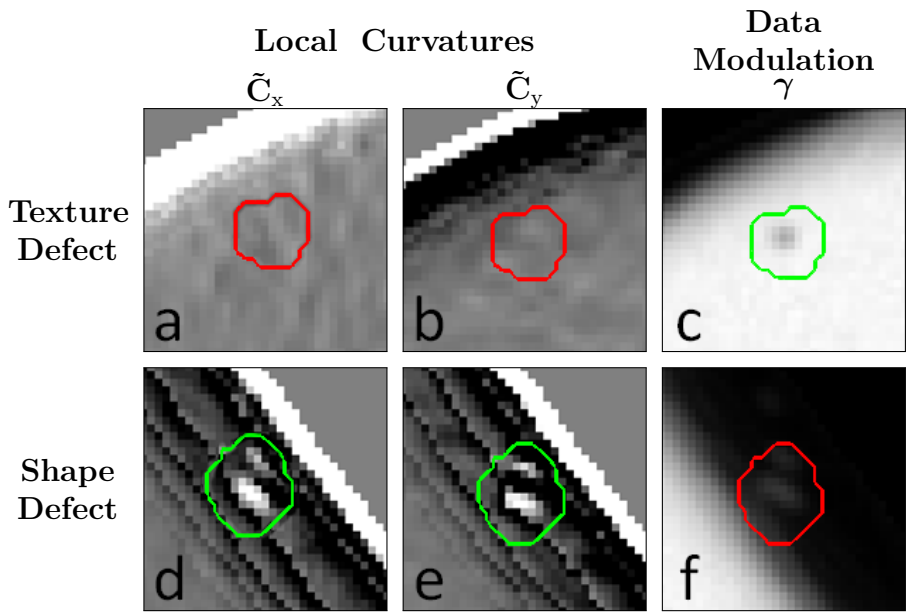

Figure 5.3: Geometry versus photometry: A texture defect (coating imperfection) with an approximate area of 0.14 mm^2 (20 pixels) is not visible in curvatures (figs. a-b), but can be seen in data modulation (fig. c). A shape defect (dent) with an area of 0.7 mm^2 (100 pixels) is visible in curvatures (figs. d-e), but can barely be seen in data modulation (fig. f). Note that dynamic range and zoom have been adjusted to enhance visibility.

- **Distortion correction:** As has been explained in Sec. 3.2.2, lens distortions produce image aberrations that alter the shape, size, and position of the observed objects. In addition, as the camera is not usually positioned at normal incidence (see Fig. 5.2(a)), an oblique view of the scene is acquired, and objects in the distance appear smaller than close objects. In surface inspection applications it is desirable to work with distortion-free images, such that the 2D grid of measurements is spatially consistent and has a homogeneous spatial resolution (uniform sampling of the surface). For this purpose, lens distortions can be compensated for with proper geometric camera calibration, and perspective correction can be applied to approximate an orthonormal view from the oblique projective images. Note that completely removing the perspective distortions would require additional information, *e.g.*, 3D data interpolation. However, if the sample's surface height range is not too large, an approximate compensation can be achieved by applying a transformation that projects the image to a previously calibrated reference plane.

- **Sample size:** Depending on the employed architecture, it might be necessary to adapt the size of measurements to match that required by the input of the neural network. This might involve operations such as cropping or resizing samples, or dividing a large sample into smaller patches, among others.

Fig. 5.4 shows an overview of the operations involved in the data acquisition stage, the resulting N-channel samples are ready to be fed to the deep learning based classification/segmentation architectures employed in the following sections.

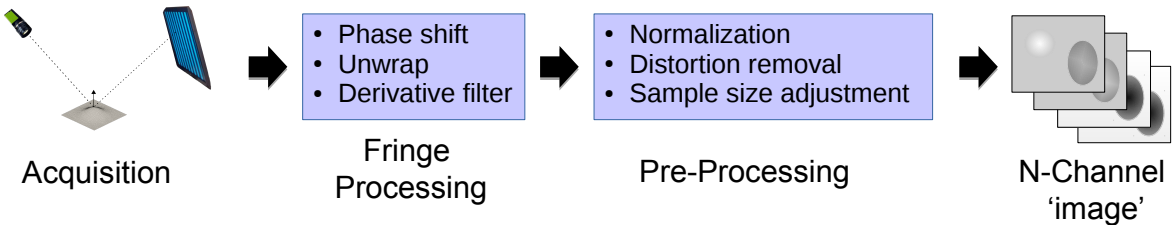

Figure 5.4: Overview of deflectometric registration

5.5 Defective Sample Classification

The first proposal consists in a sample-wise classification approach using a CNN architecture to perform both the feature extraction and classification. These architectures can represent complex functions of the input data, such that an image-to-label mapping is learned from a set of labeled samples during an initial training stage. The idea is to train a model that can be used to classify defective samples directly from the N-channel images containing geometric and photometric information (Sec. 5.4) of the SUT.

The rest of the section will be devoted to this idea: Sec. 5.5.1 describes the employed architecture, Sec. 5.5.2 deals with the experimental setup and the dataset creation. afterward, Sec. 5.5.3 overviews the training of the model, and the final validation results are shown in Sec. 5.5.4 and discussed in Sec. 5.5.5.

5.5.1 Architecture

The architecture of a CNN is composed of multiple layers, each transforming its input data into a (hopefully) more useful representation for the next layer until the sample is assigned to a class at the output. The architecture can be divided into two main parts: a convolutional stage and a dense stage. The first one is in charge of extracting local features from the input data, reducing the 2D-sample into a form easier to process in the dense stage. This convolutional stage is composed of a succession of convolutional and pooling layers. A convolutional layer consists of a set of N linear filters that extract local features to form N-channel feature maps, whereas the pooling layers perform a downsampling operation over the spatial dimensions (width, height) of these feature maps, concentrating dominant features from small areas to form smaller feature maps. The second stage is in charge of performing the classification of the sample based on the previously extracted local features. For this purpose, several fully connected layers are used to create what is known as a multi-layer perceptron (MLP) or a feed-forward neural network [GBC16].

The capacity of the model is defined by the number and configuration of layers, thus, in order to avoid over-fitting or under-fitting problems, the architecture might need to be adapted to the complexity of the problem and the amount of available data. The configuration shown in Fig. 5.5 was the one that worked best in the experiments (Sec. 5.5.4), the arrangement is similar to the ones employed by [FRHN+16] or [MMC+12].

The different elements of the architecture shown in Sec. 5.5.4 will be briefly described, the interested reader can refer to [GBC16] for a detailed explanation of the involved concepts.

Input The input layer takes a 4 channel image of size 256x256 per analyzed sample, as

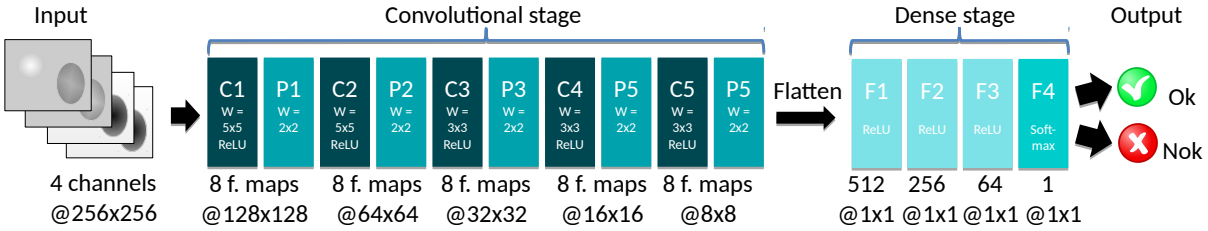

Figure 5.5: Employed CNN Architecture

provided by the deflectometry pre-processing stage (Sec. 5.4.3).

Convolutional stage This stage contains five convolutional layers (C1-C5 in Fig. 5.5), with 8 filters in each. The filters in the first two layers use 5x5 masks per channel, the rest use 3x3 masks per channel. In order to allow the representation non-linear features, a rectified linear unit (ReLU) activation function [NH10] is applied at the output of each filter. A max-pooling layer (P_i) with a filter size of 2x2 is used after each convolutional layer (C_i). At the end of this stage, the feature maps resulting from P_5 are flattened to a feature vector of size 512, as required for the dense stage.

Dense stage This stage is composed of 4 fully connected layers that successively combine the input features transforming them into a class score: the first three (F1-F3), *the hidden layers*, are composed of 512, 256, and 64 nodes (features) respectively, and also use ReLU as an activation function to allow non-linear combinations of features to be learned. The last one (F4), has a single unit that represents the probability of the sample being defective, a Softmax activation function is used to keep values in the (0,1) interval [GBC16]. Note that the model can be trivially extended to a multi-class problem changing this last dense layer.

Output Finally, the output layer provides the binary classification, i.e. it labels each part as *ok* or *nok*. For this purpose, the probability of the sample being defective (the output of F4) is compared to a threshold value (t_{nok}) to decide if the sample is accepted or not. This allows tweaking the classification towards precision ($t_{nok} > 0.5$) or recall ($t_{nok} < 0.5$).

5.5.2 Dataset

As there is no public deflectometric recording database for defect detection, all the experiments, including the ones from the next section, have been carried out with a custom-built database of components from an automotive manufacturer. The employed parts will be visible in the interior of an automobile, demanding high surface quality standards. This work is restricted to the front surface, which has a mirror-like reflective finish for aesthetic purposes, thus detecting visually perceptible imperfections is of great concern. This surface has an approximate diameter of 20 mm with a hole in the middle, and its shape is relatively flat, although it has an increasing curvature towards the borders, finishing with sharp edges. Some of the employed samples can be seen in Fig. 5.6.

In total 512 sample parts were available, with 40 of them being labeled as compliant (*ok*), and 472 as defective (*nok*). Some examples are shown later in Fig. 5.12. As this amount of samples is a rather small for training a CNN model, the dataset was artificially increased by recording each sample eight times, manually re-positioning it between the

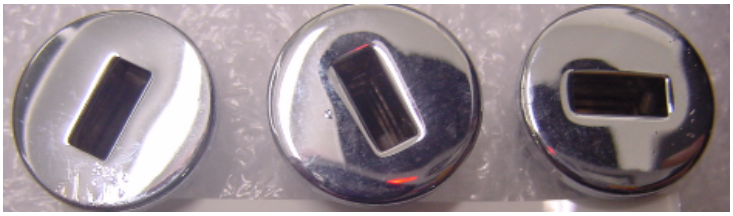

Figure 5.6: Employed test objects

measurements, such that it was rotated approximately 45 degrees in each repetition. This data augmentation strategy should also encourage rotation and positioning invariance during the training of the model. The resulting dataset is composed of 4096 samples containing 15 types of surface imperfections defined by the manufacturer regarding their standards of quality control. The dataset was divided into three sets: Out of each class, 20% percent of the samples are reserved for the final test (the test set), and the remaining are divided into a training set (%80) and a validation set used to monitor the training progress (%20). All the samples corresponding to each part were assigned to the same set to avoid possible overfitting problems.

The employed deflectometric setup consists of an off-the-shelf 24" LCD screen and a 5 Mpixel gray-scale industrial camera with an f=25mm lens. The geometrical arrangement can be seen in Fig. 5.7(a). The setup was designed to perform the simultaneous observation of a batch of 32 parts in its original packaging with an approximate lateral resolution of 0.1 mm/pixel, as shown in Fig. 5.7(b). The processing of the deflectometric recordings was done in Matlab [MAT].

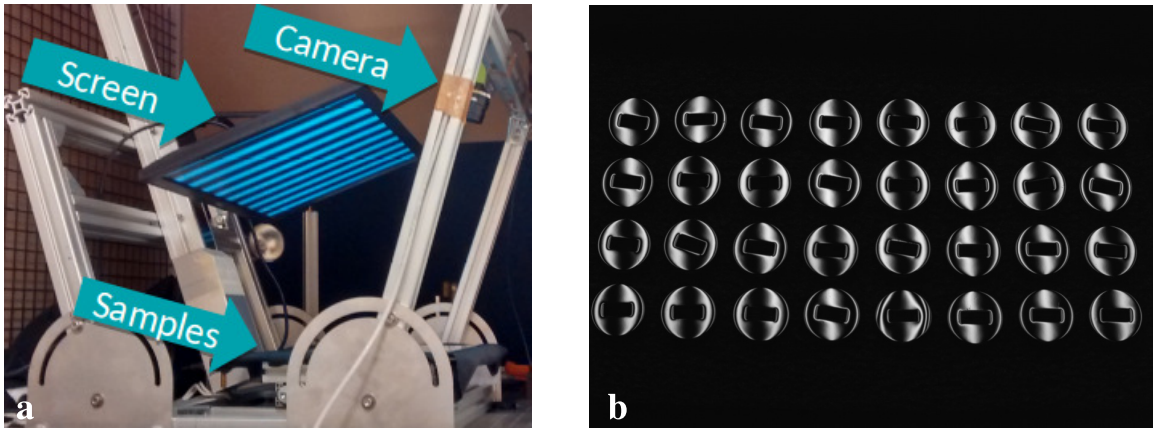

Figure 5.7: (a) Sketch of the employed setup and (b) an image from the measurement of a batch of samples.

The pattern coding strategy was designed to favor quality over speed, using pattern sequences with K=7 different spatial periods for the unwrapping (see Fig. 5.9), and N=11 phase-shifted patterns for each sequence (see Fig. 5.8). This configuration requires 154 images per batch measurement, and it is well suited to minimize the effects of non-linear response and discrete operation of screen and camera. The simultaneous observation of the whole batch results in an acquisition speed slightly over two parts per second, enough at this stage.

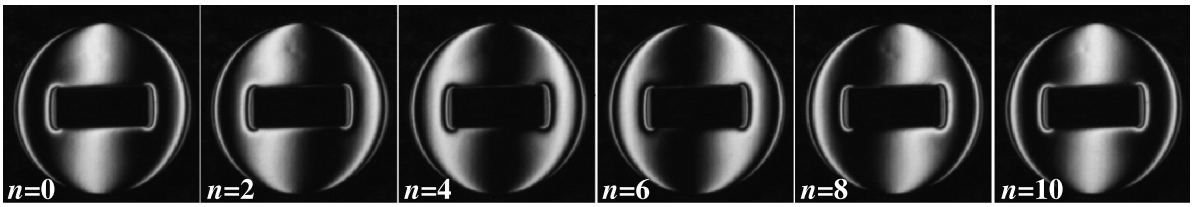

Figure 5.8: Raw recordings from the sequence of horizontal phase-shifted patterns corresponding to $k = 4$, the phase shift order n is indicated in each one.

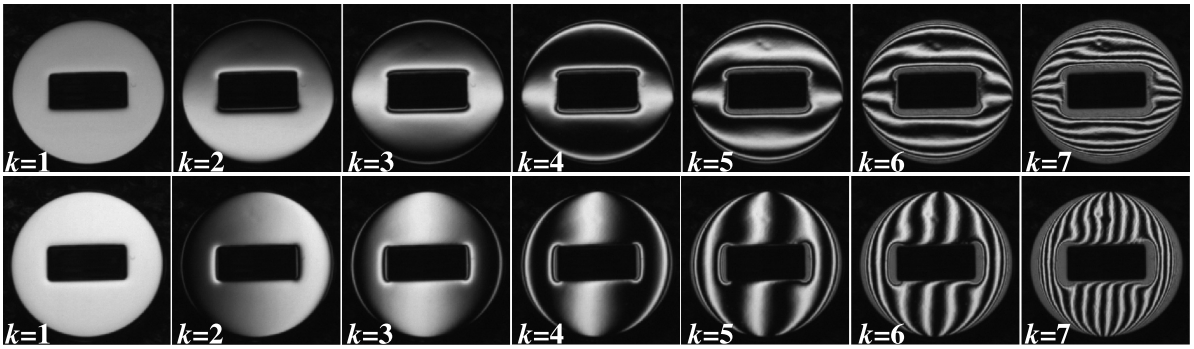

Figure 5.9: Raw recordings of horizontal and vertical fringes with different period L_k, with k specified in each figure. As can be seen, for more curved regions (at the borders) the acquired pattern appears blurred.

The recordings were processed following the procedure detailed in Sec. 5.4. As can be seen in Fig. 5.9, smaller spatial periods (higher k) provide higher sensitivity; note that the defect on the top side of the images is less visible for lower ks. However, in regions with higher surface curvature, *e.g.*, the interior and exterior borders of the sample, the information from patterns with smaller spatial periods is lost, as patterns appear blurred due to the finite camera resolution and the increased bundle spreading. In order to avoid using blurred patterns in the light-map computation, a pixel-wise adaptive depth is adopted, such that the last sequence $k_{max} \leq K$ used in the unwrapping is limited based on a minimum contrast threshold. Figs. 5.10(a)-(b) show the decoded LCD coordinates for every pixel in a measurement, Figs. 5.10(c)-(d) the maximum depth k_{max} employed for the unwrapping during their computation. The resulting local curvatures and data modulation are shown respectively in Figs. 5.11 (c)-(d) and Figs. 5.11 (e)-(f).

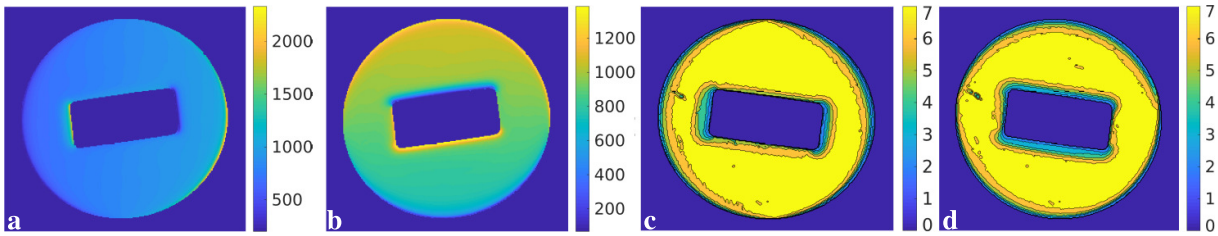

Figure 5.10: (a)–(b) Decoded LCD coordinates for every pixel in a measurement, respectively $\mathbf{l}_x$ and $\mathbf{l}_y$ (c)–(d) maximum depth k_{max} employed for the unwrapping of $\mathbf{\Phi}_u$ and $\mathbf{\Phi}_v$.

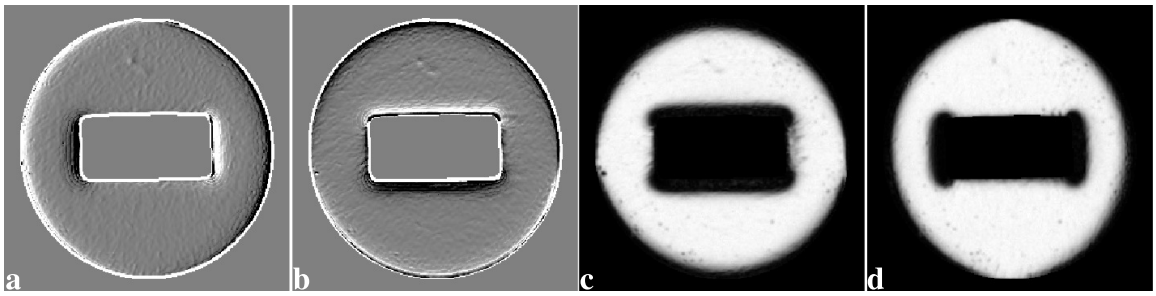

Figure 5.11: Measured values: (a)–(b) Local curvatures, respectively $\tilde{C}_{\mathrm{x}}$ and $\tilde{C}_{\mathrm{y}}$; (c)–(d) data modulation, respectively γ_x and γ_y.

Regarding the employed pre-processing operations, both lens and perspective distortions were corrected. For this purpose, the camera was calibrated using Bouget's Matlab toolbox [Bou05], assuming an extended pinhole with 2 radial distortion coefficients, and both curvature and data modulation maps were rectified. The perspective correction is done according to a plane located and oriented approximately as the measurement plane. This plane was calibrated according to a planar checkerboard calibration plate during camera calibration. Finally, the measurements were cropped to 256x256 pixel samples each containing a single part. The normalization of the data was left as a hyperparameter to tune during the training process. Fig. 5.12 shows some examples of defective samples.

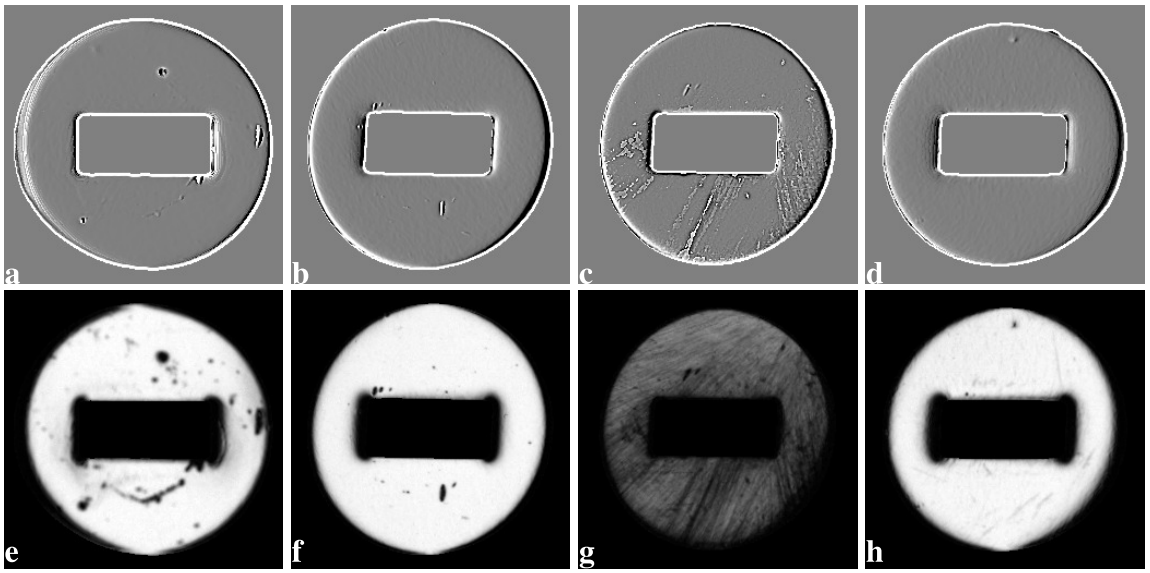

Figure 5.12: Geometrical and textural defects can be observed in $\tilde{C}_{\mathrm{x}}$ (top row) and γ_x (bottom row). Figures (a) and (e) show a part with a stain, scratches, and multiple geometric defects; (b) and (f) show a part with several dents; (c) and (g) show a part with polishing errors; (d) and (h) show a part with a smooth slit, a small pit, and multiple scratches.

5.5.3 Training

In order to validate the proposed classification approach, the architecture described in Sec. 5.5.1 was implemented using *Keras* [C+15] with *Tensorflow* [AAB+] as a backend.

Several model (hyper-)parameters were adjusted during initial runs with the training set, including the network architecture itself (mainly depth and number of filters), and also the employed normalization and regularization functions, optimizer and learning rates. The validation set was used to guide this initial design process, taking accuracy as the evaluation a metric, *i.e.*, the correctly classified sample ratio, as shown in 5.6. The best results were obtained by the architecture shown in Fig. 5.5, and independently normalizing each channel using max-min clipping with outlier removal. Several measures were taken in order to improve the training performance: oversampling was used for the *ok* class to alleviate the effects of the class unbalance in the dataset, and in order to mitigate overfitting, Batch Normalization [IS15] was applied after each convolutional layer ($C_1 - C_5$) and zero-centered Gaussian noise [Bis95, RGBV11] was added after each max-pooling layer ($P_1 - P_4$). Additionally, 50% dropout [SHK+14] was used in dense layers F1 and F3 to help avoid overfitting by breaking up non-relevant coincidental patterns.

$$Accuracy = \frac{\text{Correctly classified samples}}{\text{All samples}} \tag{5.6}$$

Once the model configuration was set, it was trained for 30 epoch, using the validation set to monitor the training progress with previously unseen data after each epoch. The evolution of the accuracy over the training period is shown in Fig. 5.13: The blue line represents the accuracy classifying the training samples, whereas the red one corresponds to the validation samples. As can be seen, the model performs better with the training data, achieving a maximum accuracy of 98.36% at epoch 30, whereas the validation accuracy reaches its maximum accuracy at epoch 28 (88.91%).

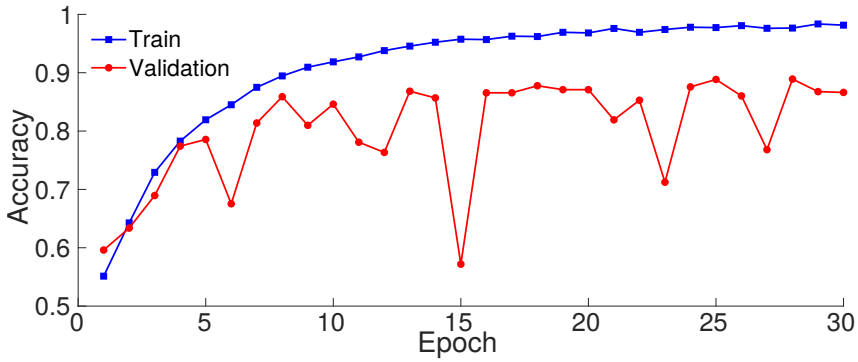

Figure 5.13: Training performance: evolution of the accuracy of the model over the training period.

5.5.4 Results

Finally, the network performance was tested using the test dataset, obtaining an accuracy of 89.57%. In order to further analyze the results, recall and precision metrics are also considered. The recall measures the ratio of defective samples that were classified as defective -*how many of the defects were detected?*-, and the precision measures the ratio of samples classified as defective that were really defective -*how many detected defects corresponded to real defects?*. For this purpose, the assumed hypothesis is that the sample is defective, and thus, samples correctly classified as defective are considered true positives (TP). Similarly, samples correctly classified as non-defective are considered true

negatives (TN), whereas incorrectly classified ones are counted as true positives (TP) or false negatives (FN). In this way, the recall and precision are computed as defined in 5.7, showing a recall of 97.65% and a precision of 83.33%, which indicates that the classifier tends to favor the *nok* class.

$$Recall = \mathrm{TP}/(\mathrm{TP}+\mathrm{FN})\,, \qquad Precision = \mathrm{TP}/(\mathrm{TP}+\mathrm{FP}) \tag{5.7}$$

5.5.5 Discussion

This section has explored the use of a CNN architecture for automated defective sample classification using a combination of photometric and geometric information derived from deflectometric recordings. The experiments indicate the feasibility of the approach, enabling the detection of a wide range of shape and texture defects, and obtaining classification rates around 89% in a test performed with a batch of parts from an automotive industry manufacturer. This shows that deep learning techniques can be used to exploit deflectometric recordings for reflective/specular surface inspections.

The results exhibited much better recall rates ($> 95\%$) than precision rates ($< 85\%$), which was probably caused by the use of a relatively small and unbalanced class dataset, with many more samples belonging to the *nok* class. In any case, in industrial inspection systems it is critical to assure no defective samples are shipped to a customer, so the recall measure is quite meaningful. Of course, the precision is also important, but the samples classified as *nok* can be revised later by human operators, returning the incorrectly classified ones to the production line. For this reason, these results can be considered good enough at this initial stage, although they should be improved in order to apply the method in a real industrial application.

Compared to shallow learning approaches previously employed in deflectometric defect detection literature, the employed method avoids the handcrafted feature selection work, which in previous experiences of the author took a considerable share of the engineering efforts devoted to the development of a deflectometric surface inspection system. Note, also, that experiments in many related works, *e.g.*, [SMG01, Zie14, KSM17], are based on planar, or at least continuous, test surfaces. The test surface employed in this work presents some challenges (edges, holes, variable curvature...) that were handled seamlessly without any manual intervention. Regarding the empirical results, a comparison is difficult, as each work has employed its own case study, with different surfaces and defects, and thus the comparison would not be fair.

Despite the promising results, the method is still in an early stage of development, and several improvements can be made. The employed dataset is relatively small and unbalanced, which not only appears to hamper the training process, but also restricts the conclusions to the employed case study. Therefore, future research should consider validating the approach with larger amounts of data, and also in different case studies. It would also be logical to investigate the application of techniques to cope with small and unbalanced datasets, as this situation is frequent in industrial applications. In this work, the dataset contained far more defective samples, but the reverse is a fairly common situation, *i.e.*, a dataset with defect-free samples overrepresented. Perhaps an anomaly detection approach could be interesting. Additionally, the application of transfer learning techniques could be investigated, employing pre-trained networks to take advantage of previous knowledge. Finally, instead of approaching the problem as a binary classification that just labels parts as *ok* or *nok*, more elaborate schemes that account for the

identification of the defect types, their size, severity, or location could be tested. Some of these ideas will be addressed in the next section in the context of a defect segmentation approach.

5.6 Defect Segmentation

In the previous section, it has been shown that a wide range of defects can be detected by training a CNN to extract and combine information from curvature and contrast, avoiding the handcrafted definition of features. However, this approach is based on a sample-wise classification and does not provide any information on the location or size of defects.

In this section a second approach is explored, which consists in investigating if the Fully Convolutional Network concept can be extended to identify surface defects in deflectometric recordings, adapting the idea of combining local curvatures and data information used in the previous section to also provide feedback on the location and boundaries of the object and the possible defective areas present.

FCNs are trainable architectures that take a raw image as an input and return a pixel-wise class map as an output. Compared to the previous approach, this one is interesting because instead of classifying the whole sample, it performs a segmentation of the measurement, providing more detailed information on the defects, and also visual feedback on what image areas have triggered its decisions. Furthermore, in this approach more 'expert' information can be transferred to the model, as the ground-truth used for training must specify all the relevant image locations with pixel-wise annotations.

The rest of the section is organized as follows: Sec. 5.6.1 describes the employed architecture, Sec. 5.6.2 deals with the experimental setup and the dataset creation. Afterward, Sec. 5.6.3 overviews the training of the model, and the final validation results are shown in Sec. 5.6.4 and discussed in Sec. 5.6.5.

5.6.1 Architecture

In this work the U-Net [RFB15] architecture is adopted, an FCN that has shown to work well in different biomedical image segmentation tasks that required end-to-end training with low amounts of data, which is also a common case in industrial inspection with non-conventional imaging techniques.

The U-Net network is composed of two parts: an encoder and a decoder. The encoder is trained to extract a deep feature hierarchy using a CNN network without the fully connected layers. The downsampling steps of the encoder facilitate the extraction of meaningful features for classification, but localization information is lost in the process. The decoder part transforms these low-resolution features into a full-resolution segmentation map by successive upsampling steps. In addition, the localization is enhanced by using skip connections that combine features from both parts. Fig. 5.14 shows a sketch of the employed U-Net implementation:

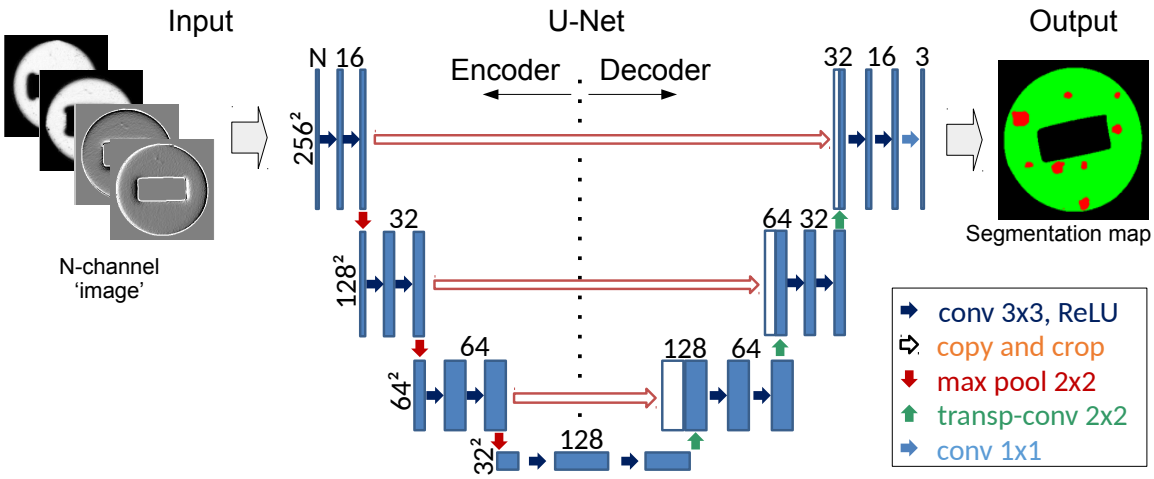

Figure 5.14: Architecture of U-Net [RFB15] Network: The left side corresponds to the encoder, the right side to the decoder. Blue boxes represent multi-channel feature maps, with the number of channels denoted on top of the box, and their size in pix^2 at the side. White boxes correspond to copied feature maps. Arrows denote the performed operations.

The four main parts of the architecture shown in Fig. 5.14 are briefly described in the following; more detailed explanations can be found in the original U-net paper [RFB15].

Input The input layer takes an N-channel image, as provided by the deflectometry preprocessing stage (Sec. 5.4.3). There is no size restriction for the analyzed sample other than the computing resources; the example in Fig. 5.5 is based on 256x256 images.

Encoder This part is similar to the convolutional stage of a conventional CNN, as shown in the architecture in Sec. 5.5.1. As can be seen in Fig. 5.14, the encoder is divided into "steps" or "blocks" composed of two convolutional layers and a max-pooling layer. These layers are configured such that, at each step, the number of filters is doubled and the size of the feature maps (width and height) is halved. The encoder ends with two additional convolutional layers, also called the "bottleneck". All convolutional layers use a rectified linear unit (ReLU) as the activation function and 3x3 masks per channel, whereas the max-pooling layers use a mask size of 2x2. Batch Normalization is applied after each convolutional layer.

Decoder This part is symmetric to the encoder, except that the decoder blocks are composed of a transposed convolution layer [ZKTF10, DV16] and two convolutional layers. In each block, the transposed-convolution increases the resolution of the feature maps, doubling the size of the input feature map, and cuts the number of channels in half. This up-sampled feature map is then stacked with the one coming from the encoder part, and two successive convolutional layers are used to combine these features; the resulting feature map is passed to the following block. These convolutional layers use ReLU as activation function and 3x3 masks per channel, whereas transposed convolution layers use a mask size of 2x2. At the end of the decoder, all the features corresponding to a pixel are transformed into class scores using an additional convolutional layer with a 1x1 mask and one filter for each class in the dataset, such that the resulting feature map channels correspond to class

scores. These class scores are converted into class probabilities using softmax as an activation function.

Output Finally, the output layer provides a pixel-wise classification by selecting the label with the highest probability, *e.g.*, it labels each pixel as belonging to background, surface or defect.

The configuration of the U-Net architecture might need to be adapted for other applications, *e.g.*, adjusting the capacity to the amount of data and the complexity of the problem, however, the concept behind the four main parts of the implementation shown in Fig. 5.14 will remain the same.

5.6.2 Dataset

In order to validate the proposed defect segmentation approach, a dataset with deflectometric recordings with their corresponding pixel-wise annotated ground-truth is required. A deflectometric recording database was already created for the experiments in Sec. 5.5.2, thus, using the same case study, the database creation is "reduced" to labeling the defects. For this purpose, the recordings of 480 sample parts were annotated (*e.g.*, fig. Fig. 5.15(d)) indicating the pixels corresponding to the background (B), surface (OK), and defects that were visible to the human eye (DF_i), where $i \in 1, 2, .., 15$ corresponds to the defect class. Borderline defects, which would not lead to rejection but could be seen, were also labeled.

Recall from Sec. 5.5.2 that each part was recorded eight times, thus, counting the repetitions, 3840 samples containing 15 classes of surface imperfections needed to be labeled pixel-wise. In order to speed up this task, the ground truth corresponding to the original orientation of each part was annotated manually, while the remaining seven were generated automatically using best fitting. This automatic labeling approach introduces some noise in the ground-truth: the positioning variability causes perspective changes in the images, resulting in incorrect labeling of pixels around defective regions. In order to account for these inaccuracies, the regions enclosing defects were annotated including a safety margin. Therefore, the drawn areas were generally larger than the defect itself.

The dataset was divided into three sets: 70% for training, 20% for training-validation, and 10% for final testing. All the samples corresponding to a part (all the repetitions) were always assigned to the same set to avoid possible over-fitting problems.

Additionally, in order to have a more meaningful evaluation of the results, each labeled defect in the test set was classified according to its visual impact: Low impact (LI) for tolerable defects, Medium impact (MI) for defects that could lead to a rejection if several are present, and High impact (HI) for defects that lead directly to the rejection of the part.

5.6.3 Training

The architecture was adjusted during several initial test runs. These short experiments were used to define the capacity of the model, including the number of contracting steps, and the number of channels of the feature maps at each step; the final configuration is shown in Fig. 5.14. Several hyperparameters were also tuned at this stage, including the optimizer (RMSProp) and the data normalization (max-min clipping excluding outliers).

The highly unbalanced class distribution (only 6% of the pixels contain defects) appeared to hamper the training process, so several measures were taken to reduce its effects: on the training side, the focal loss function [LGG+17] gave us the best results. This loss function is a variation of the standard cross-entropy that down-weights the loss assigned to well-classified pixels, reducing the contribution from easy examples and focusing training on hard examples. On the data side, merging all the different defect classes DF_i into one class (NOK) and using additional synthetic data augmentation worked best.

5.6.4 Results

Four models were considered for the final experiments. They all share the same architecture, but each takes different input data: two use either local curvatures (model 2a) or data modulation solely (model 2b), and the other two combine information on both, using curvatures and data modulation modulus (model 3) or curvatures and data modulation in two directions (model 4), as shown in Table 5.1. These models were trained for 600 epochs, using the validation set to select the iteration with the lowest loss value. In order to evaluate the generalization capabilities of the selected models, the final experiments are performed using the test set, which is composed of previously unseen samples containing more than 1500 defects. Figure 5.15 shows one measurement.

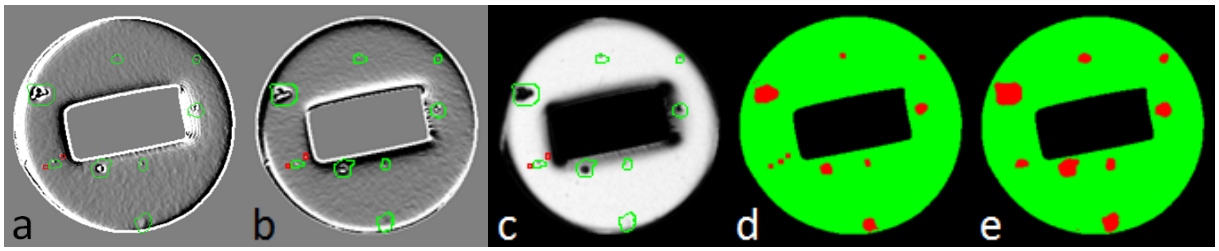

Figure 5.15: (a)-(c) correspond to local curvatures and data modulation ($\tilde{C}_x$, $\tilde{C}_y$, and γ), where contours shown in green indicate segmented defects, the ones in red not segmented ones. (d)-(e) show the corresponding ground truth label and predicted segmentation; the color code indicates the class each pixel belongs to, using green for OK, red for NOK, and black for background B classes.

In general, the results showed that the models correctly classified between 98 to 99.99% of the pixels depending on the measurement. However, given the class unbalanced dataset, these numbers mostly reflect the correct classification of background and OK classes, as can be seen by comparing the ground truth and the segmentation shown in Figs. 5.15 (d) and (e). In order to focus on the defect detection capabilities, pixels corresponding to the same defective area are grouped as independent objects or "blobs", and the defect segmentation task is analyzed as a binary classification problem, taking recall and precision as metrics (as defined in 5.7). For this purpose, defects intersected by at least 50% in the segmentation are considered as true positives (TP), and false negatives (FN) if they are not; blobs incorrectly detected as defective are regarded as false positives (FP).

The results corresponding to the blob analysis are shown in Table 5.1. As can be seen, the combination of curvature and data-modulation (models 3 and 4) results in increased recall rates compared to using just one of them, showing that the combined information leads to better performance.

Model	Channels	Global		Visual Impact Based		
		Recall	**Precision**	**Recall-LI**	**Recall-MI**	**Recall-HI**
2a	$\tilde{C}_x, \tilde{C}_y$	0.49	0.78	0.41	0.39	0.79
2b	γ_x, γ_y	0.65	0.60	0.58	0.56	0.88
3	$\tilde{C}_x, \tilde{C}_y, \gamma$	0.82	0.73	0.67	0.83	0.99
4	$\tilde{C}_x, \tilde{C}_y, \gamma_x, \gamma_y$	0.86	0.66	0.72	0.89	1.00

Table 5.1: Segmentation Results - blob analysis

The achieved global recall and precision rates shown in the first two columns are a bit low for industrial inspection, especially for the first two models. In any case, although no information on the visual impact of the defects was fed to the network during training, it appears to be correlated to recall rates, with high impact (HI) defects achieving the highest rates, and low visual impact ones (LI) the lowest. Models 3 and 4 are both capable of detecting most of the HI-MI defects but struggle with LI ones. Although model 4 results in better recall rates, the balance of model 3 is favored and adopted for the remainder of the experiments.

Several variations of model 3 were considered in an attempt to improve the results: the U-net architecture was modified to increase and decrease its capacity by varying both the number of down-steps and the number of feature maps at each step, and different normalization and loss functions were used to train them. The results are shown in Fig. 5.16:

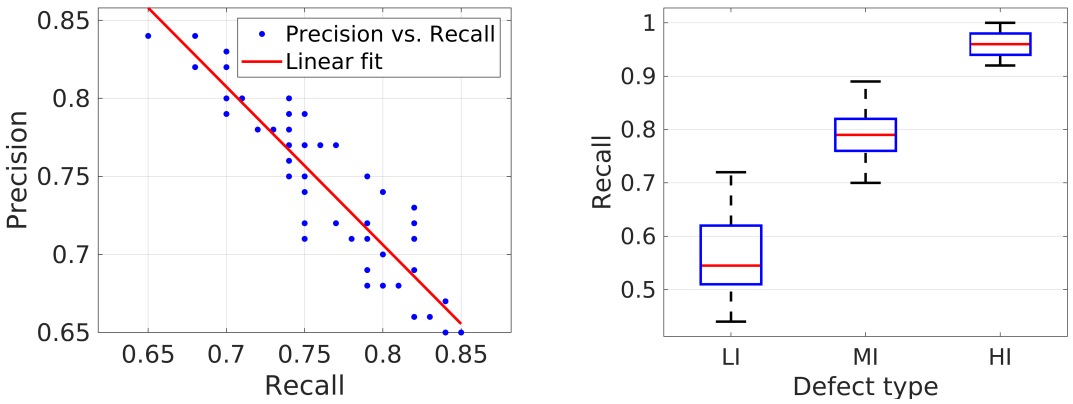

Figure 5.16: Results of variations of model 3: (a) The compromise between recall and precision is represented by the linear fit. (b) The correlation between the visual impact of the defects and recall rates is apparent comparing the box plots corresponding to each type of defect.

As can be seen in Fig. 5.16(a), the models which increased the recall rate reduced the precision rate, suggesting that there was a compromise between the achieved recall and precision rates. The correlation between the visual impact of the defects and recall rates is apparent in the box plot shown in Fig. 5.16(b). On each box, the red mark corresponds to the median, the bottom and top edges of the box represent the 1st and 3rd quartiles, and the whiskers indicate the extreme values. As shown in the box plots, the high impact (HI) defects not only result in the highest recall rates in all models, but their values have

the smallest dispersion. The low visual impact defects (LI) obtain the lowest values and have the highest dispersion, which indicates that the less critical defects are the ones the models have more difficulties learning.

A closer look at the results revealed some interesting findings, both when the segmentation was correct and when it failed. Fig. 5.17 shows several defects that were correctly detected. The effects of the safety margin for the automatic labeling can be noticed as the segmented area is generally larger than the defect itself, which indicates that more precise labeling should be employed in applications where accurate boundaries are critical.

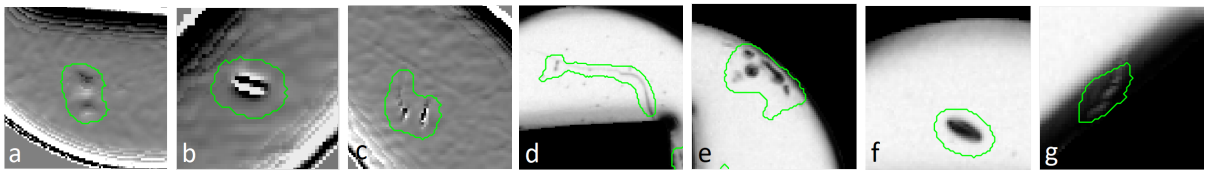

Figure 5.17: Correctly classified defects: Local curvatures showing (a) two smooth indentations (2.3 mm^2), (b) a small deep dent (0.5 mm^2), (c) a deep mark (1.56 mm^2). Data modulation showing (d) a scratch (2 mm^2), (e) pores (3.5 mm^2), (f) a stain (1.25 mm^2), (g) a deep mark at the edge (1.85 mm^2). Dynamic range and zoom have been adjusted to enhance visibility

Incorrectly classified samples also expose some weaknesses of the trained models. Several examples are shown in Fig. 5.18, using red contours to indicate FN, blue ones for FP, and green ones for TP.

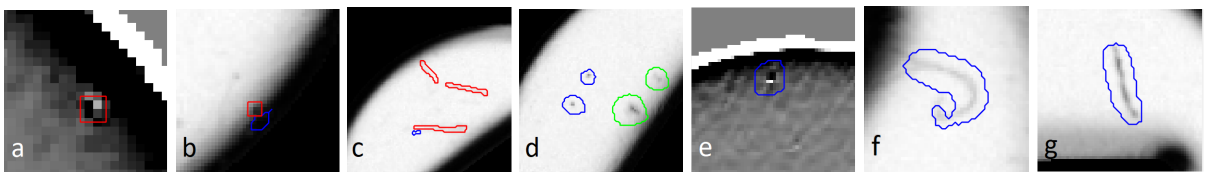

Figure 5.18: Incorrectly classified defects, with false negatives marked in red, and false positives in blue: (a) undetected small dent (0.23 mm^2), (b) dent at the edge (0.11 mm^2) that was only partially detected, (c) three undetected slight scratches (0.35/0.42/0.48 mm^2), (d) four surface finish imperfections were detected, the smaller ones had not been labeled (0.3/0.4 mm^2); (e) inclusion resulting in very small blister that had not been labeled (0.06 mm^2), (f) textile fiber posed on the surface (2.37 mm^2), and (g) non-labeled scratch (2.7 mm^2). Dynamic range and zoom have been adjusted for visibility.

An analysis of the false negatives showed that the majority of the occurrences (55%) were due to non detected type LI defects, *e.g.*, Fig. 5.18 (a). More severe defects at the edges of the surface (16%), *e.g.*, Fig. 5.18 (b), and thin superficial scratches (17%), *e.g.*, Fig. 5.18 (c), also exhibited some systematic failures. Similarly, most false positives were due to unlabeled slight defects, but the majority (77%) could have been labeled as type LI defects. Figures 5.18(d)–(e) show examples of this type of failure. Additionally, there were also several unlabeled elements, such as extraneous foreign matter posed on the surface (5%) or unlabeled defects (3%), *e.g.*, Figs. 5.18 (f) and (d), respectively. In addition, there were defects that were only partially detected (3%), as shown in Fig. 5.18 (b).

These "failures" were consistent over the repetitions of the same part: if the model failed in one it generally failed in all. An analysis of the LI limit defects revealed that distinguishing between some false positives and false negatives was impossible, which indicates that the inclusion criterion for LI defects might have varied over the labeling process that took several days. This also makes sense of the apparent compromise between the recall and precision encountered in the experiments.

5.6.5 Discussion

This section has explored the use of an FCN architecture for automated defect segmentation using a combination of photometric and geometric information derived from deflectometric recordings. The proposed method consists in training a U-Net network to extract and combine features from local curvatures and data modulation, such that it learns how to perform a pixel-wise classification of defect candidates. Its capabilities have been demonstrated in a real industrial problem, where pixels containing geometric or textural imperfections were automatically discriminated by a trained model.

Several experiments were performed using an industrial case study, showing that the combination of geometric and photometric information results in an improved global accuracy compared to just using either of them, showing that more than 98% of the pixels could be correctly classified in all measurements. A defect detection analysis resulted in moderate recall (82%) and precision (0.73%) rates, but showing a sensitivity closely correlated to the visual impact of the defects, with higher recall rates for more severe defect classes. Low visual impact defects were the ones which caused most classification errors, either for not detecting them or detecting small unlabeled defects. Several attempts to improve the results showed that there was some kind of correlation or limit between the employed evaluation metrics, as an improvement in one resulted in a reduction in the other. A posterior analysis revealed that this was probably caused by the subjectivity and inaccuracies at labeling time.

Compared to the previous classification approach, segmentation provides more detailed information on the defects, at the expense of a tedious labeling process, which increases the work involved in training the model. As no attempt has been made to classify the samples, their results are not comparable, but some of the findings in this second approach could help the classification approach: the employed loss function and the synthetic data augmentation have helped with the small and unbalanced dataset, these strategies were not tested in the previous experiments. In addition, most of the false positive occurrences in the segmentation experiments were due to small defects or contamination. This fact might have also influenced the experiments in the previous approach, a hypothesis to keep in mind in future research.

Further research should concentrate on data acquisition and defect segmentation. Regarding data acquisition, different ways of combining the captured patterns can be studied to feed richer information to the segmentation stage. Moreover, taking into account that the recall rates for low visual impact defects were lower than those for MI and HI, it would be interesting to explore the effects of the camera resolution on the LI defects. Regarding segmentation, different labeling strategies could be considered, *e.g.*, encoding severity information in the training ground truth. Finally, as has been stated in the discussion of the classification approach (Sec. 5.5.5), these experiments have been limited to a relatively small and unbalanced (only 6% of the pixels correspond to the defect class)

dataset. Future research should consider validating the approach with larger amounts of data and more quantitative labeling criteria.

5.7 Chapter Conclusions

In this chapter, the adaptation of two deep learning architectures for industrial inspection of reflective surfaces has been studied. The main hypothesis was that a combination of photometric and geometric information derived from deflectometric recordings of industrial surfaces would enable the detection of a wide range of shape and texture defects using DL architectures. A data acquisition procedure has been proposed, which consists in extracting local curvatures and data modulation from sequences of phase-shifted sinusoidal patterns to form an N-channel 'image' that can be directly fed to deep learning architectures. Two different approaches have been explored to directly exploit these measurements, one for detecting defective surfaces based on a sample-wise classification using a CNN model, and the other one for segmenting surface defects based on a pixel-wise classification using a Unet-FCN model. The proposed approaches differ in the amount of information they provide; while the first one results in a global sample evaluation, the latter provides rich information about the location and size of possible defects. However, note that this extra information comes at the expense of higher computing resources for training a model and more human resources dedicated to preparing a dataset.

The experimental results in a real industrial case study indicate that both, CNN and FCN architectures, are able to learn relevant features from deflectometric data, enabling the classification and segmentation of defects after a statistical optimization based on a dataset of user-provided examples. The employed dataset is relatively small and unbalanced, which not only appears to hamper the training process, but also restricts the conclusions to the employed case study. Therefore, future research should consider validating the approach with larger amounts of data, and also in different case studies.

In any case, compared to shallow learning approaches, the employed methods avoid the handcrafted feature selection work by learning the feature extraction itself during the training process. Furthermore, the test surface employed in this work presents some challenges, such as edges, holes, or variable curvature, that were handled seamlessly without any manual intervention, which shows the potential of the employed methods. Nevertheless, it has to be noted that obtaining accurate models does require testing different architectures and tweaking a few hyperparameters, as shown in the experiments.

Despite the promising results, the methods are still in an early stage of development. In order to contribute to a near-term industrial utilization, several improvements have been suggested for both classification and segmentation approaches, as well as for the data acquisition procedure.

Finally, a logical extension of this work would consist in the combination of the classification and segmentation architectures into a single one that copes with the classification of defect candidates and also with the final rejection decision for the inspected object.

Chapter 6

Conclusion and Future Work

Deflectometric techniques are a powerful tool for the automated quality control of specular or shiny surfaces. These techniques exploit the dependence of specular reflections on surface normals to recover shape information from the surface under inspection, showing a local sensitivity in the nanometer range.

By using quantitative deflectometry, the normal vector field and the 3D shape of a surface can be obtained. However, these techniques do not take full advantage of the potential local sensitivity, resulting in global accuracies several orders of magnitude worse. It is commonly accepted that measurement errors introduced by deflectometric setups and their miscalibration are the main shortcomings of quantitative measurements. However, their influence in the measurements has not yet been sufficiently investigated. The first part of the thesis aims to contribute to a better understanding of these problems, focusing particularly on how they affect the measurements performed in an Active Reflection Grating Photogrammetry setup.

The second part of the thesis focuses on qualitative deflectometry, which is used to detect surface imperfections rather than performing absolute measurements. These techniques are very useful for aesthetic defect detection, as measurements can be performed using simple acquisition setups and relaxed calibration requirements. However, exploiting deflectometric data for automatic defect detection requires further processing that can involve a considerable engineering effort. With the aim of reducing the complexity of this task, we have investigated the use of a deep learning (DL) approach for the automated defect identification in deflectometric data. The following sections outline the main conclusions and future lines.

6.1 Main Conclusions

Regarding the errors in absolute measurements, the camera model and its calibration are addressed in Ch. 3, where the propagation of the uncertainty from an extended pinhole camera calibration through the different stages of a deflectometric measurement process has been modeled following the GUM framework and validated using the MCM methodology. The proposed model allows us to obtain an estimate of the uncertainties in the camera rays, and their contribution to the uncertainty in the measured normal vector field and the triangulated surface. In addition, the uncertainties in the intermediate parameters can be used as a metric to select the lens distortion model. Afterwards, Ch. 4 addresses the effects of several non-ideal characteristics of the LCD in deflectometric

measurements. The analyzed sources of error have been divided into low-frequency and high-frequency errors. The former included the non-planarity of the LCD screen and the refraction in its translucent layers, while the latter correspond mainly to decoding errors.

The experimental part from these chapters has been based on the Active Reflection Grating Photogrammetry setup from our laboratory. Different experiments have been performed in order to investigate the measurement errors and their sources based on real measurements and simulations that allowed us to isolate the error sources. The simulation results indicate that the non-planarity of the screen and the uncertainty of the calibrated camera parameters are the main sources of error, as the combination of these errors reasonably approximates the errors obtained in the real measurements. In addition, the errors arising from the decoding uncertainty were also significant, which indicates that employed pattern coding strategy was not as robust as expected.

Chapter 5 was devoted to qualitative measurements. The main hypothesis was that DL architectures will learn relevant features from a combination of photometric and geometric information, enabling the detection of a wide range of shape and texture defects with a reduced feature engineering effort. Two different approaches have been explored to exploit these recordings for industrial surface inspection: a method for automated defective sample classification, where a convolutional neural network (CNN) architecture is used to extract relevant features in order to directly classify defective samples, and another method for automated segmentation of surface defects, where a fully convolutional neural network (FCN) architecture is used to perform a pixel-wise classification of the measurements.

The experimental results in a real industrial case study indicate that both CNN and FCN architectures are able to learn relevant features from deflectometric data, enabling the classification and segmentation of defects after a statistical optimization based on a dataset of user-provided examples. The experiments demonstrate that the combination of geometric and photometric information results in higher recall rates, showing a sensitivity closely correlated to the visual impact of the defects, *i.e.*, the system is more accurate with high visual impact defects than with low impact ones. However, the employed dataset is relatively small, unbalanced, and includes several labeling errors, which not only appears to hamper the training process, but also makes the conclusions preliminary. In any case, compared to shallow learning approaches, the employed methods avoid the handcrafted feature selection work by learning the feature extraction itself during the training process. Furthermore, the test surface employed in this work presents some challenges, such as edges, holes, or variable curvature, that were handled seamlessly without any manual intervention, which shows the potential of these methods.

6.2 Future lines

Regarding the errors in absolute measurements, the following extensions should be considered:

- The camera calibration errors have been studied for the extended pinhole camera model; it would be interesting to do the same for other more complex camera models in order to investigate how much can these uncertainties be reduced.

- This work has studied the effects of the LCD model errors considering the calibration and measurement holistically; further research should address the effects on the LCD

calibration more in depth.

- In order to obtain more accurate measurements, the screen model should consider at least the non-planar shape of the screen in order to reduce low-frequency errors. In addition, the sources of high-frequency errors in the decoding of the patterns should be studied more in depth in order to adopt measures to mitigate their effects. In any case, the pattern coding strategy should employ a larger number of phase-shifts sacrificing acquisition time for decoding accuracy.

- Taking into account that all the experiments have been carried out with a planar surface positioned at the calibration position, we expect larger errors when measuring non-planar shaped surfaces or planar but in different positions and/or orientations. Thus, further experiments should be performed in order to generalize these findings.

Regarding the methods proposed for qualitative measurements, the following extensions should be considered:

- In order to generalize the proposed methods, both the classification and segmentation approaches should be validated with larger amounts of data and more quantitative labeling criteria.

- In this work local curvatures and data modulation have been obtained from the captured patterns reducing the amount of data that was fed to the neural network architectures. Different ways of combining the captured patterns could be studied in order to provide richer information to the classification and segmentation architectures.

- The segmentation and classification architectures could be combined into a single one that copes with the defect candidates and also with the final rejection decision for the inspected object. This would be of interest for industrial applications as it would improve the traceability of the manufacturing process by providing specific information on the surface areas that triggered the rejection of a sample. Furthermore, we expect that using such multiple-output architecture would guide the training procedure to learn more significant features improving the performance of the resulting models.

Bibliography

[AAB+] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from https://www.tensorflow.org/.

[Ade01] E. H. Adelson. On seeing stuff: the perception of materials by humans and machines. In *Photonics West: Human Vision and Electronic Imaging VI*, pages 1–12. International Society for Optics and Photonics (SPIE), 2001.

[AHG05] P. Aswendt, R. Hofling, and S. Gartner. Industrial inspection of specular surfaces using a new calibration procedure. In *Optical Metrology: Optical Measurement Systems for Industrial Inspection IV*, volume 5856, pages 393–400. International Society for Optics and Photonics (SPIE), 2005.

[AKV+15] A. Angelova, A. Krizhevsky, V. Vanhoucke, A. S. Ogale, and D. Ferguson. Real-time pedestrian detection with deep network cascades. In *Proceedings of British Machine Vision Conference (BMVC)*, volume 2, page 4, 2015.

[Arr98] K. O. Arras. An introduction to error propagation: Derivation, meaning and examples. Technical Report EPFL-ASL-TR-98-01 R3, Autonomous Systems Lab, Institute of Robotic Systems, Swiss Federal Institute of Technology Lausanne (EPFL), 1998.

[ATHA11] L. Armesto, J. Tornero, A. Herraez, and J. Asensio. Inspection system based on artificial vision for paint defects detection on cars bodies. In *2011 IEEE International Conference on Robotics and Automation*, pages 1–4, May 2011.

[Bal12] J. Balzer. A gauss-newton method for the integration of spatial normal fields in shape space. *Journal of Mathematical Imaging and Vision*, 44(1):65–79, 2012.

[BEAA19] J. Balzategui, L. Eciolaza, and N. Arana-Arexolaleiba. Semi-automatic quality inspection of solar cell based on convolutional neural networks.

In *24th IEEE Conference on Emerging Technologies and Factory Automation (ETFA)*, 2019.

[BHB11] J. Balzer, S. Hofer, and J. Beyerer. Multiview specular stereo reconstruction of large mirror surfaces. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2537–2544, June 2011.

[Bis95] C. M. Bishop. Training with noise is equivalent to tikhonov regularization. *Neural Computation*, 7(1):108–116, Jan 1995.

[BKB19] J. Bartsch, M. Kalms, and R. Bergmann. Improving the calibration of phase measuring deflectometry by a polynomial representation of the display shape. *Journal of the European Optical Society - Rapid publications*, 15(1), 2019.

[BKC17] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *EEE Trans. Pattern Anal. Mach. Intell.*, 39(12):2481–2495, Dec 2017.

[BLH+13] J. Burke, W. Li, A. Heimsath, C. v. Von Kopylow, and R. Bergmann. Qualifying parabolic mirrors with deflectometry. *Journal of the European Optical Society - Rapid publications*, 8(0):1–6, 2013.

[BLS+10] T. Bothe, W. Li, M. Schulte, C. von Kopylow, R. B. Bergmann, and W. P. O. Jüptner. Vision ray calibration for the quantitative geometric description of general imaging and projection optics in metrology. *Appl. Opt.*, 49(30):5851–5860, Oct 2010.

[BLvKJ04] T. Bothe, W. Li, C. von Kopylow, and W. P. O. Juptner. High-resolution 3d shape measurement on specular surfaces by fringe reflection. In *Photonics Europe: Optical Metrology in Production Engineering*, volume 5457, pages 411–422. International Society for Optics and Photonics (SPIE), 2004.

[BNP+18] J. Bartsch, J. R. Nüß, M. H. U. Prinzler, M. Kalms, and R. B. Bergmann. Effects of non-ideal display properties in phase measuring deflectometry: A model-based investigation. In *Photonics Europe, Optical Micro and Nanometrology VII*, volume 10678, pages 10678 – 10678 – 10. International Society for Optics and Photonics (SPIE), 2018.

[Bou05] J. Y. Bouguet. Camera calibration toolbox for matlab, 2005. http://www.vision.caltech.edu/bouguetj/calib_doc/.

[Bra] G. Bradski. Opencv (open source computer vision library). https://opencv.org/.

[Bro66] D. C. Brown. Decentering Distortion of Lenses. *Photometric Engineering*, 32(3):444–462, 1966.

[Bro71] D. C. Brown. Close-range camera calibration. *Photogrammetric Engineering*, 37(8):855–866, 1971.

[BS03] T. Bonfort and P. Sturm. Voxel carving for specular surfaces. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 591–596. IEEE, 2003.

[BSG06] T. Bonfort, P. Sturm, and P. Gargallo. General specular surface triangulation. In P. Narayanan, S. Nayar, and H.-Y. Shum, editors, *Computer Vision ACCV 2006*, volume 3852 of *Lecture Notes in Computer Science*, pages 872–881. Springer Berlin Heidelberg, 2006.

[Bur16] J. Burke. Inspection of reflective surfaces with deflectometry. In *Proceedings of the International Conference on Processes in Combined Digital Optical & Imaging Methods*, pages 108–111, 2016.

[BW10] J. Balzer and S. Werling. Principles of shape from specular reflection. *Measurement*, 43(10):1305–1317, 2010.

[BWB06] J. Balzer, S. Werling, and J. Beyerer. Regularization of the deflectometry problem using shading data. In *Optics East, Two and Three-Dimensional Methods for Inspection and Metrology IV*, volume 6382, pages 63820B–63820B–11. International Society for Optics and Photonics (SPIE), 2006.

[C+15] F. Chollet et al. Keras. https://keras.io, 2015.

[Cau10] Y. Caulier. Inspection of complex surfaces by means of structured light patterns. *Opt. Express*, 18(7):6642–6660, Mar 2010.

[CF98] T. A. Clarke and J. G. Fryer. The development of camera calibration methods and models. *The Photogrammetric Record*, 16(91):51–66, 1998.

[CGGS12] D. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. In *Proceedings of the 25th International Conference on Neural Information Processing Systems (NIPS)*, pages 2843–2851, 2012.

[Cho17] F. Chollet. *Deep learning with Python*. Manning Publications Co., 2017.

[Cla98] J. C. Clarke. Modelling uncertainty: A primer, 1998.

[CSWS17] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[CW08] R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning (ICML)*, pages 160–167. ACM, 2008.

[DCB+18] S. Deitsch, V. Christlein, S. Berger, C. Buerhop-Lutz, A. K. Maier, F. Gallwitz, and C. Riess. Automatic classification of defective photovoltaic module cells in electroluminescence images. *CoRR*, abs/1807.02894, 2018.

[DL16] P. Drap and J. Lefèvre. An exact formula for calculating inverse radial lens distortions. *Sensors*, 16(6), 2016.

[Dom12] P. Domingos. A few useful things to know about machine learning. *Commun. ACM*, 55(10):78–87, October 2012.

[DV16] V. Dumoulin and F. Visin. A guide to convolution arithmetic for deep learning. *arXiv e-prints*, page arXiv:1603.07285, Mar 2016.

[EKKH08] S. Ettl, J. Kaminski, M. C. Knauer, and G. Häusler. Shape reconstruction from gradient data. *Applied Optics*, 47(12):2091–2097, 2008.

[FB18] M. Faraji and A. Basu. Simplified active calibration. *ArXiv e-prints*, June 2018.

[FKR+12] C. Faber, M. Kurz, C. Röttinger, E. Olesch, D. Domingos, A. Löwenstein, G. Häusler, and E. Uhlmann. Two approaches to use phase measuring deflectometry in ultra precision machine tools. *Proc. Euspen*, 12(2):84–87, 2012.

[FOKH12] C. Faber, E. Olesch, R. Krobot, and G. Häusler. Deflectometry challenges interferometry: the competition gets tougher! In *Proc. SPIE 8493, Interferometry XVI: Techniques and Analysis*, volume 8493, pages 84930R–84930R–15. International Society for Optics and Photonics (SPIE), 2012.

[FPT10] M. Fischer, M. Petz, and R. Tutsch. Evaluation of lcd monitors for deflectometric measurement systems. In *Photonics Europe: Optical Sensing and Detection*, volume 7726, pages 7726 – 7726 – 10. International Society for Optics and Photonics (SPIE), 2010.

[FPT16] M. Fischer, M. Petz, and R. Tutsch. Model-based noise estimation for fringe projection systems. In *18. GMA/ITG-Fachtagung Sensoren und Messsysteme*, pages 534 – 539, 2016.

[Fra01] C. S. Fraser. Photogrammetric camera component calibration: A review of analytical techniques. In A. Gruen and T. S. Huang, editors, *Calibration and Orientation of Cameras in Computer Vision*, pages 95–121. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[FRHN+16] S. Faghih-Roohi, S. Hajizadeh, A. Núñez, R. Babuska, and B. D. Schutter. Deep convolutional neural networks for detection of rail surface defects. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 2584–2589, July 2016.

[FY07] N. Funk and Y. Yang. Using a raster display for photometric stereo. In *Fourth Canadian Conference on Computer and Robot Vision (CRV '07)*, pages 201–207, May 2007.

[GBC16] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[GLGL18] Y. Guo, Y. Liu, T. Georgiou, and M. S. Lew. A review of semantic segmentation using deep neural networks. *International Journal of Multimedia Information Retrieval*, 7(2):87–93, Jun 2018.

[GP98] D. C. Ghiglia and M. D. Pritt. *Two-dimensional phase unwrapping: theory, algorithms, and software*, volume 4. Wiley New York, 1998.

[Han11] T. Hanning. *High precision camera calibration*. Springer, 2011.

[HDY+12] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, B. Kingsbury, and T. Sainath. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal Processing Magazine*, 29:82–97, November 2012.

[HE11] G. Häusler and S. Ettl. Limitations of optical 3d sensors. In R. Leach, editor, *Optical Measurement of Surface Topography*, pages 23–48. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[Hei] J. Heikkila. Camera calibration toolbox for matlab. http://www.ee.oulu.fi/ jth/calibr/.

[Hei00] J. Heikkila. Geometric camera calibration using circular control points. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(10):1066–1077, Oct 2000.

[HFOE13] G. Häusler, C. Faber, E. Olesch, and S. Ettl. Deflectometry vs. interferometry. In *Optical Metrology: Optical Measurement Systems for Industrial Inspection VIII*. International Society for Optics and Photonics (SPIE), 2013.

[HIH+13] M. B. Hullin, I. Ihrke, W. Heidrich, T. Weyrich, G. Damberg, and M. Fuchs. State of the art in computational fabrication and display of material appearance. In *Eurographics Annual Conference (STAR)*, 2013.

[HIZA18] L. Huang, M. Idir, C. Zuo, and A. Asundi. Review of phase measuring deflectometry. *Optics and Lasers in Engineering*, 107:247 – 257, 2018.

[HRW13] S. Höfer, M. Roschani, and S. Werling. Pattern coding strategies for deflectometric measurement systems. In *Optical Metrology: Videometrics, Range Imaging, and Applications XII; and Automated Visual Inspection*, volume 8791, pages 87911O–87911O–11. International Society for Optics and Photonics (SPIE), 2013.

[HS96] J. Heikkila and O. Silven. Calibration procedure for short focal length off-the-shelf ccd cameras. In *Proceedings of 13th International Conference on Pattern Recognition*, volume 1, pages 166–170 vol.1, Aug 1996.

[HS97] J. Heikkila and O. Silven. A four-step camera calibration procedure with implicit image correction. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1106–1112, Jun 1997.

[HXG+16] L. Huang, J. Xue, B. Gao, C. McPherson, J. Beverage, and M. Idir. Modal phase measuring deflectometry. *Opt. Express*, 24(21):24649–24664, Oct 2016.

[HZ04] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.

[IKL+10] I. Ihrke, K. N. Kutulakos, H. Lensch, M. Magnor, and W. Heidrich. Transparent and specular object reconstruction. *Computer Graphics Forum*, 29(8):2400–2426, 2010.

[IS15] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 448–456. JMLR.org, 2015.

[JCGM 100:2008] JCGM100:2008. Evaluation of measurement data - guide to the expression of uncertainty in measurement (gum). Joint Committee for Guides in Metrology, 2008.

[JCGM101:2008] JCGM101:2008. Evaluation of measurement data - supplement 1 to the "guide to the expression of uncertainty in measurement" - propagation of distributions using a monte carlo method. Joint Committee for Guides in Metrology, 2008.

[JCGM102:2011] JCGM102:2008. Evaluation of measurement data – supplement 2 to the "guide to the expression of uncertainty in measurement" – extension to any number of output quantities. Joint Committee for Guides in Metrology, 2011.

[KB06] J. Kannala and S. S. Brandt. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(8):1335–1340, Aug 2006.

[KKH03] J. Kaminski, M. Knauer, and G. Häusler. Calculating curvatures from discrete slope data. Technical report, Annual Report Optik, Universität Erlangen-Nürnberg, 2003.

[KKH04] M. C. Knauer, J. Kaminski, and G. Häusler. Phase measuring deflectometry: a new approach to measure specular free-form surfaces. In *Photonics Europe*, pages 366–376. International Society for Optics and Photonics, 2004.

[KL08] S. Kammel and F. P. Leon. Deflectometric measurement of specular surfaces. *IEEE Transactions on Instrumentation and Measurement*, 57(4):763–769, April 2008.

[KLF+14] B. Komander, D. Lorenz, M. Fischer, M. Petz, and R. Tutsch. Data fusion of surface normals and point coordinates for deflectometric measurements. *Journal of Sensors and Sensor Systems*, 3(2):281–290, 2014.

[KRH06] M. C. Knauer, C. Richter, and G. Häusler. 3d sensor zoo–species and natural habitats. *Laser Technik Journal*, 3(1):33–37, 2006.

[KSH12] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems (NIPS)*, NIPS'12, pages 1097–1105, USA, 2012.

[KSM17] C. Kofler, G. Spöck, and R. Muhr. Classifying defects in topography images of silicon wafers. In *2017 Winter Simulation Conf. (WSC)*, pages 3646–3657, Dec 2017.

[LBOM98] Y. LeCun, L. Bottou, G. B. Orr, and K. R. Müller. Efficient backprop. In G. B. Orr and K.-R. Müller, editors, *Neural Networks: Tricks of the Trade*, pages 9–50. Springer Berlin Heidelberg, 1998.

[LGG+17] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar. Focal loss for dense object detection. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[LOYH14] Y. Liu, E. Olesch, Z. Yang, and G. Häusler. Fast and accurate deflectometry with crossed fringes. *Advanced Optical Technologies*, 3(4):441 – 445, 2014.

[LSD15] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, June 2015.

[LZGH16] T. T. Le, M. Ziebarth, T. Greiner, and M. Heizmann. Systematic design of object shape matched wavelet filter banks for defect detection. In *39th Int. Conf. on Telecommunications and Signal Processing (TSP)*, pages 470–473, June 2016.

[LZZ19] S. Li, X. Zhao, and G. Zhou. Automatic pixel-level multiple damage detection of concrete structure using fully convolutional network. *Computer-Aided Civil and Infrastructure Engineering*, 0(0):1– 19, 2019.

[Mal12] A. S. Malek. *Online fabric inspection by image processing technology.* PhD thesis, Université de Haute Alsace-Mulhouse, 2012.

[MANC17] T. Malekzadeh, M. Abdollahzadeh, H. Nejati, and N. Cheung. Aircraft fuselage defect detection using deep neural networks. *CoRR*, abs/1712.09213, 2017.

[MAT] MATLAB. version 9.4.0.949201 (R2018a). The MathWorks Inc., 2018.

[Mel94] T. Melen. *Geometrical modelling and calibration ofvideo cameras for underwater navigation.* PhD thesis, Norges tekniske høgskole, Institutt for teknisk kybernetikk, 1994.

[MIA17] D. Maestro-Watson, A. Izaguirre, and N. Arana-Arexolaleiba. Lcd screen calibration for deflectometric systems considering a single layer refraction model. In *2017 IEEE International Workshop of Electronics, Control, Measurement, Signals and their Application to Mechatronics (ECMSM)*, pages 1–6, May 2017.

[MMC+12] J. Masci, U. Meier, D. Ciresan, J. Schmidhuber, and G. Fricout. Steel defect classification with max-pooling convolutional neural networks. In *2012 Int. Joint Conf. on Neural Networks (IJCNN)*, pages 1–6. IEEE, 2012.

[MP06] K. L. Mak and P. Peng. Detecting defects in textile fabrics with optimal gabor filters. *Enformatika*, 13:274–282, 2006.

[MPB13] J.-L. Maire, M. Pillet, and N. Baudet. Measurement of the perceived quality of a product. *International Journal of Metrology and Quality Engineering*, 4(02):63–69, 2013.

[MSAT17] J. Molina, J. E. Solanes, L. Arnal, and J. Tornero. On the detection of defects on specular car body surfaces. *Robotics and Computer-Integrated Manufacturing*, 48:263 – 278, 2017.

[MW04] J. Mallon and P. F. Whelan. Precise radial un-distortion of images. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 1, pages 18–21 Vol.1, Aug 2004.

[MWBEAA19a] D. Maestro-Watson, J. Balzategui, L. Eciolaza, and N. Arana-Arexolaleiba. Deep learning for deflectometric inspection of specular surfaces. In *International Joint Conference SOCO'18-CISIS'18-ICEUTE'18*, pages 280–289. Springer International Publishing, 2019.

[MWBEAA19b] D. Maestro-Watson, J. Balzategui, L. Eciolaza, and N. Arana-Arexolaleiba. Deflectometric data segmentation based on fully convolutional neural networks. In *SPIE International Conference on Quality Control by Artificial Vision (QCAV)*, volume 11172, 2019.

[MWIAAI15] D. Maestro-Watson, A. Izaguirre, N. Arana-Arexolaleiba, and A. Iturrospe. A simple deflectometric method for measurement of quasi-plane specular surfaces. In *2015 IEEE International Workshop of Electronics, Control, Measurement, Signals and their Application to Mechatronics (ECMSM)*. IEEE, 2015.

[MXZ19] L. Ma, W. Xie, and Y. Zhang. Blister defect detection based on convolutional neural network for polymer lithium-ion battery. *Applied Sciences*, 9(6), 2019.

[NFK13] T. Nagato, T. Fuse, and T. Koezuka. Defect inspection technology for a gloss-coated surface using patterned illumination. In *Proc. SPIE 8661, Image Processing: Machine Vision Applications VI*, 2013.

[NH10] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pages 807–814. Omnipress, 2010.

[NHVC17] V. Natarajan, T. Hung, S. Vaikundam, and L. Chia. Convolutional networks for voting-based anomaly classification in metal surface inspection. In *2017 IEEE International Conference on Industrial Technology (ICIT)*, pages 986–991, 2017.

[NTO+18] F. Nagata, K. Tokuno, A. Otsuka, T. Ikeda, H. Ochi, K. Watanabe, and M. K. Habib. Design tool of deep convolutional neural network for intelligent visual inspection. *IOP Conference Series: Materials Science and Engineering*, 423:012073, nov 2018.

[NWR08] D. Nehab, T. Weyrich, and S. Rusinkiewicz. Dense 3d reconstruction from specularity consistency. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, June 2008.

[OH14] E. Olesch and G. Häusler. Deflectometry with better accuracy. In *DGaO Proceedings*, 2014.

[PAK19] J.-K. Park, W.-H. An, and D.-J. Kang. Convolutional neural network based surface inspection system for non-patterned welding defects. *International Journal of Precision Engineering and Manufacturing*, 20(3):363–374, Mar 2019.

[PBKB19] S. K. Patra, J. Bartsch, M. Kalms, and R. B. Bergmann. Phase measurement deviations in deflectometry due to properties of technical surfaces. In E. Novak and J. D. Trolinger, editors, *Applied Optical Metrology III*, volume 11102, pages 180 – 188. International Society for Optics and Photonics, SPIE, 2019.

[Pet06] M. Petz. *Rasterreflexions-Photogrammetrie - Ein neues Verfahren zur geometrischen Messung spiegelnder Oberflächen.* PhD thesis, Technische Universität Braunschweig, 2006.

[PFT13] M. Petz, M. Fischer, and R. Tutsch. Systematic errors in deflectometry induced by use of liquid crystal displays as reference structure. In *21st IMEKO TC2 Symposium on Photonics in Measurement*, 2013.

[PFT17] M. Petz, M. Fischer, and R. Tutsch. Defekterkennung an spiegelnden und transparenten oberflächen durch abbildung einer örtlich modulierbaren lichtquelle. *tm - Technisches Messen*, 85(2):79–87, 2017.

[PLK06] F. Puente León and S. Kammel. Inspection of specular and painted surfaces with centralized fusion techniques. *Measurement*, 39(6):536 – 546, 2006.

[PPD+13] T. Puntous, S. Pavan, D. Delafosse, M. Jourlin, and J. Rech. Ability of quality controllers to detect standard scratches on polished surfaces. *Precision Engineering*, 37(4):924 – 928, 2013.

[PT05] M. Petz and R. Tutsch. Reflection grating photogrammetry: a technique for absolute shape measurement of specular free-form surfaces. In *Optics and Photonics: Optical Manufacturing and Testing VI*, volume 5869, pages 58691D–1–58691D–12. International Society for Optics and Photonics (SPIE), 2005.

[PTB+15] J. Posada, C. Toro, I. Barandiaran, D. Oyarzun, D. Stricker, R. de Amicis, E. B. Pinto, P. Eisert, J. Dollner, and I. Vallarino. Visual computing as a key enabling technology for industrie 4.0 and industrial internet. *Computer Graphics and Applications, IEEE*, 35(2):26–40, 2015.

[QDA18] Y. Quéau, J.-D. Durou, and J.-F. Aujol. Normal integration: A survey. *Journal of Mathematical Imaging and Vision*, 60(4):576–593, 2018.

[Rap12] H. H. Rapp. *Reconstruction of Specular Reflective Surfaces Using Auto-Calibrating Deflectometry*. KIT Scientific Publishing, 2012.

[RFB15] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention*, pages 234–241, 2015.

[RGBV11] S. Rifai, X. Glorot, Y. Bengio, and P. Vincent. Adding noise to the input of a model trained with a regularized objective. *CoRR*, abs/1104.3250, 2011.

[RHT18] R. Ren, T. Hung, and K. C. Tan. A generic deep-learning-based approach for automated surface inspection. *IEEE Transactions on Cybernetics*, 48(3):929–940, March 2018.

[RJGZ14] H. Ren, X. Jiang, F. Gao, and Z. Zhang. Absolute height measurement of specular surfaces with modified active fringe reflection photogrammetry. In *Proc. SPIE 9204, Interferometry XVII: Advanced Applications*, volume 9204, pages 920408–920408–8. International Society for Optics and Photonics (SPIE), 2014.

[RSL05] S. Ramalingam, P. Sturm, and S. K. Lodha. Towards complete generic camera calibration. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 1093–1098 vol. 1, June 2005.

[San14] M. Sandner. Hybrid reflectometry - 3d shape measurement on scattering and reflective surfaces. In *DGaO Proceedings*. German Society of Applied Optics, 2014.

[SB06] H. Schreiber and J. H. Bruning. Phase shifting interferometry. In *Optical Shop Testing*, chapter 14, pages 547–666. John Wiley & Sons, Ltd, 2006.

[SC06] W. Sun and J. R. Cooperstock. An empirical evaluation of factors influencing camera calibration accuracy using three publicly available techniques. *Machine Vision and Applications*, 17(1):51–67, Apr 2006.

[SDS+18] P. Stavroulakis, O. Davies, A. Shaheen, G. Tzimiropoulos, and R. Leach. Artificial intelligence-enhanced multi-material form measurement for additive materials. In *Solid Freeform Fabrication Symposium*, 2018.

[SFA11] C. Schmalz, F. Forster, and E. Angelopoulou. Camera calibration: active versus passive targets. *Optical Engineering*, 50(11):113601, 2011.

[SFPL10] J. Salvi, S. Fernandez, T. Pribanic, and X. Llado. A state of the art in structured light patterns for surface profilometry. *Pattern recognition*, 43(8):2666–2680, 2010.

[SH97] H. O. Saldner and J. M. Huntley. Temporal phase unwrapping: application to surface profiling of discontinuous objects. *Appl. Opt.*, 36(13):2770–2775, May 1997.

[SHK+14] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[SHL95] S.-W. Shih, Y.-P. Hung, and W.-S. Lin. When should we consider lens distortion in camera calibration. *Pattern Recognition*, 28(3):447 – 461, 1995.

[SHM14] D. Soukup and R. Huber-Mörk. Convolutional neural networks for steel surface defect detection from photometric stereo images. In *International Symposium on Visual Computing*, pages 668–677. Springer, 2014.

[SMG01] R. Seulin, F. Merienne, and P. Gorria. Dynamic lighting system for specular surface inspection. In *Proc. SPIE 4301, Machine Vision Applications in Industrial Inspection IX*, volume 4301, 2001.

[SPS+] K. Strobl, C. Paredes, W. Sepp, S. Fuchs, and K. Arbter. The DLR Camera Calibration Toolbox (DLR CalLab and CalDe). Institute of Robotics and Mechatronics of the German Aerospace Center (DLR). http://www.robotic.dlr.de/callab.

[SPW+10] P. Su, R. E. Parks, L. Wang, R. P. Angel, and J. H. Burge. Software configurable optical test system: a computerized reverse hartmann test. *Applied Optics*, 49(23):4404 – 4412, Aug 2010.

[SRT+11] P. Sturm, S. Ramalingam, J.-P. Tardif, S. Gasparini, and J. Barreto. Camera models and fundamental concepts used in geometric computer vision. *Foundations and Trends in Computer Graphics and Vision*, 6(1–2):1–183, 2011.

[SSGJ99] H. Sari-Sarraf and J. S. Goddard Jr. Vision system for on-loom fabric inspection. *Industry Applications, IEEE Transactions on*, 35(6):1252–1259, 1999.

[Str05] G. Strang. *Linear algebra and its applications*. Brooks, 4ed. edition, 2005.

[Sur04] Y. Surrel. Deflectometry: a simple and efficient noninterferometric method for slope measurement. In *Xth SEM international congress on experimental mechanics*, 2004.

[SWS08] P. Sengottuvelan, A. Wahi, and A. Shanmugam. Automatic fault analysis of textile fabric using imaging systems. *Research Journal of Applied Sciences*, 3(1):26–31, 2008.

[THTB18] R. Tennakoon, R. Hoseinnezhad, H. Tran, and A. Bab-Hadiashar. Visual inspection of storm-water pipe systems using deep convolutional neural networks. In *Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics - Volume 1: ICINCO,*, pages 135–140. INSTICC, 2018.

[TIK82] M. Takeda, H. Ina, and S. Kobayashi. Fourier-transform method of fringe-pattern analysis for computer-based topography and interferometry. *JosA*, 72(1):156–160, 1982.

[TLGS02] M. Tarini, H. Lensch, M. Goesele, and H.-P. Seidel. Shape from distortion: 3d range scanning of mirroring objects. In *ACM SIGGRAPH 2002 conference abstracts and applications*, pages 248–248. ACM, 2002.

[TLGS05] M. Tarini, H. P. Lensch, M. Goesele, and H.-P. Seidel. 3d acquisition of mirroring objects using striped patterns. *Graphical Models*, 67(4):233 – 259, 2005.

[Tsa87] R. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal on Robotics and Automation*, 3(4):323–344, August 1987.

[TSU+17] Y. Takada, T. Shiina, H. Usami, Y. Iwahori, and M. K. Bhuyan. Defect detection and classification of electronic circuit boards using keypoint extraction and cnn features. In *PATTERNS 2017 - The Ninth International Conferences on Pervasive Patterns and Applications*, pages 97–102, Athens, Greece, 2017.

[TvGMM17] Z. Tang, R. G. von Gioi, P. Monasse, and J. Morel. A precision analysis of camera distortion models. *IEEE Transactions on Image Processing*, 26(6):2694–2704, June 2017.

[TZM+18] X. Tao, D. Zhang, W. Ma, X. Liu, and D. Xu. Automatic metallic surface defect detection and recognition with convolutional neural networks. *Applied Sciences*, 8(9), 2018.

[Vás17] P. Vásquez. *Ultrasound Image Processing in the Evaluation of Labor Induction Failure Risk*. PhD thesis, Mondragon University, 2017.

[Wal14] G. Walker. Fundamentals of projected-capacitive touch technology, June 2014. www.walkermobile.com Accessed 2019-05-20.

[WCH92] J. Weng, P. Cohen, and M. Herniou. Camera calibration with distortion models and accuracy evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(10):965–980, Oct 1992.

[WCQS18] T. Wang, Y. Chen, M. Qiao, and H. Snoussi. A fast and robust convolutional neural network-based defect detection model in product quality control. *The International Journal of Advanced Manufacturing Technology*, 94(9):3465–3471, Feb 2018.

[WM94] G.-Q. Wei and S. D. Ma. Implicit and explicit camera calibration: theory and experiments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):469–480, May 1994.

[WORK13] M. Weinmann, A. Osep, R. Ruiters, and R. Klein. Multi-view normal field integration for 3d reconstruction of mirroring objects. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2504–2511. IEEE, 2013.

[WSRS16] D. Weimer, B. Scholz-Reiter, and M. Shpitalni. Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection. *CIRP Annals*, 65(1):417 – 420, 2016.

[WYY+15] Y. Wu, H. Yue, J. Yi, M. Li, and Y. Liu. Phase error analysis and reduction in phase measuring deflectometry. *Optical Engineering*, 54(6):1 – 9, 2015.

[XBK+15] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015.

[XGJ18] Y. Xu, F. Gao, and X. Jiang. Enhancement of measurement accuracy of optical stereo deflectometry based on imaging model analysis. *Optics and Lasers in Engineering*, 111:1 – 7, 2018.

[YNDP18] D. Yudin, A. Naumov, A. Dolzhenko, and E. Patrakova. Software for roof defects recognition on aerial photographs. *Journal of Physics: Conference Series*, 1015:032152, may 2018.

[YW14] D.-K. Yang and S.-T. Wu. *Fundamentals of Liquid Crystal Devices*. John Wiley & Sons, Ltd, 2014.

[YWG17] Z. Yu, X. Wu, and X. Gu. Fully convolutional networks for surface defect inspection in industrial environment. In M. Liu, H. Chen, and M. Vincze, editors, *Computer Vision Systems*, pages 417–426, Cham, 2017. Springer International Publishing.

[ZCZ+17] S. Zhou, Y. Chen, D. Zhang, J. Xie, and Y. Zhou. Classification of surface defects on steel sheet using convolutional neural networks. *Materials Technology*, 51(1):123–131, 2017.

[Zha00] Z. Zhang. A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1330–1334, 2000.

[Zha09] S. Zhang. Phase unwrapping error reduction framework for a multiple-wavelength phase-shifting algorithm. *Optical Engineering*, 48(10):105601–105601–8, 2009.

[Zie14] M. Ziebarth. Empirical comparison of defect classifiers on specular surfaces. In J. Beyerer and A. Pak, editors, *Proceedings of the 2013 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory*, number IES-2013-13 in Karlsruher Schriften zur Anthropomatik, pages 155–169. KIT Scientific Publishing, 2014.

[Zie15] M. Ziebarth. Defect perception thresholds on specular surfaces. In *Proceedings of the 2014 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory*, volume 20, pages 123–132. KIT Scientific Publishing, 2015.

[ZKTF10] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus. Deconvolutional networks. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2528–2535, June 2010.

[ZWH+17] Z. Zhang, Y. Wang, S. Huang, Y. Liu, C. Chang, F. Gao, and X. Jiang. Three-dimensional shape measurements of specular objects using phase-measuring deflectometry. *Sensors*, 17(12):2835, 2017.