Calibration and 3D vision with a color-polarimetric camera

Joaquin Rodriguez

THESE DE DOCTORAT DE L'ETABLISSEMENT UNIVERSITE BOURGOGNE FRANCHE-COMTE

PREPAREE A L'UNIVERSITE DE BOURGOGNE

Ecole doctorale n° 37

Sciences Physiques pour l'Ingénieur et Microtechniques (SPIM)

Doctorat en Informatique

Par

M RODRIGUEZ Joaquin

Calibration and 3D vision with a color-polarimetric camera

Thèse présentée et soutenue au Creusot, le 19 décembre 2023

Composition du Jury :

Mme AINOUZ, Samia	Professeur des Universités	INSA Rouen Normandie	Présidente, Rapporteuse
M GOUDAIL, François	Professeur des Universités	Université Paris-Saclay	Rapporteur
M VIOLLET, Stéphane	Directeur de Recherche CNRS	Université d'Aix-Marseille	Examinateur
M MOREL, Olivier	Maître de Conférences, HDR	Université de Bourgogne	Directeur de thèse
M LEW-YAN-VOON, Lew	Maître de Conférences, HDR	Université de Bourgogne	Co-directeur de thèse
M MARTINS, Renato	Maître de Conférences	Université de Bourgogne	Co-encadrant de thèse

Abstract

Among the different sensing modalities, vision sensors are the ones that provide the most abundant environmental information. Additionally, the usage of a short focal length lens allows to easily increase the observed area. The release of color and polarimetric imagers makes it possible to extend even more the polarimetric application related to depth estimation. Indeed, the polarization parameters of the reflected light are related to the nature and to the geometry of the objects, which can be used advantageously. In this thesis, our main objective is to study the usage of polarization data to enhance the perception capabilities applied to robotics tasks, particularly in the task of scene depth reconstruction. Furthermore, we aim to push the knowledge in the field of polarization imaging by providing other researchers with a set of tools that will allow them to quickly access the polarization modality. After doing a complete introduction to the polarization theory and modeling, we describe how to calibrate a Division of Focal Plane (DoFP) sensor. This sensing device allows to capture two modalities (color and polarization) with a single snapshot. The new calibration technique that we propose enables this device to provide more accurate measurements by fitting a mathematical model to each individual pixel. The method we present here aims to reduce the amount of equipment and, thus the experimental time required to obtain calibrated measurements. We make a detailed explanation of the physics underlying the Shape from Polarization (SfP) technique, which enables the normal field estimation of an object by using polarization cues. All the required equations as well as their inverted versions for deriving the vector parameters from the polarization state are detailed while taking into consideration the type of reflection and material. We also put in evidence the effects of our calibration algorithm over the estimation of the normal vector field by using polarization. The estimation of depth information using artificial intelligence has seen significant growth in recent years. In this context, we also propose a deep-learning network to estimate depth based on a middle-fusion architecture, and a polarimetric loss function. The objective of this development is to show how to effectively integrate the polarization theory constraints into

a data-driven algorithm. A qualitative and quantitative evaluation of the results shows the interest of using an RGB-polarimetric imager thanks to the contribution of the polarization information. During this research work, a complete software toolbox was also developed, providing the scientific community with simplified access to polarimetric imaging.

Résumé

Parmi les différentes modalités employées en détection, les capteurs de vision sont ceux qui fournissent le plus d'informations sur l'environnement. L'utilisation d'objectifs à courte focale permet en outre d'augmenter facilement la zone observée. L'apparition sur le marché d'imageurs couleurs et polarimétriques permet d'étendre encore davantage les applications en estimation de profondeur. En effet, les paramètres de polarisation de la lumière réfléchie sont liés à la nature des objets ainsi qu'à leur géométrie et peuvent être utilisés avantageusement. Dans cette thèse, notre objectif principal est d'étudier l'utilisation des données de polarisation pour améliorer les capacités de perception appliquées aux tâches robotiques, notamment dans la reconstruction de la profondeur de scène. De plus, nous visons à enrichir les connaissances dans le domaine de l'imagerie de polarisation en fournissant à d'autres chercheurs un ensemble d'outils qui leur permettront d'accéder rapidement à la modalité de polarisation. Après avoir effectué une introduction complète à la théorie et à la modélisation de la polarisation, nous décrivons comment calibrer un capteur de polarisation de division de plan focale. Ce dispositif de détection permet de capturer deux modalités (couleur et polarisation) avec une seule prise de vue. La nouvelle technique de calibration que nous proposons permet à ce dispositif de fournir des mesures plus précises en ajustant un modèle mathématique selon chaque pixel. La méthode que nous présentons ici vise à réduire la quantité d'équipement, donc le temps expérimental nécessaire pour obtenir des mesures calibrées. Nous détaillons également toute la physique sous-jacente à la technique de Shape-from-polarization (SfP) qui permet d'estimer le champ des normales d'un objet en utilisant l'information de la polarisation. Les équations nécessaires et les modèles inverses pour dériver les paramètres des vecteurs depuis l'état de polarisation sont détaillées tout en tenant compte du type de réflexion et du matériau. Nous mettons ici en avant l'intérêt de notre algorithme de calibrage sur l'estimation du champ de normales par polarisation L'estimation d'informations de profondeur grâce à l'intelligence artificielle a connu un

essort très important ces dernières années. Dans ce contexte, nous proposons également un réseau d'apprentissage profond pour estimer la profondeur basé sur une architecture de fusion intermédiaire et une fonction de perte polarimétrique. L'objectif de ce développement est de montrer comment intégrer efficacement les contraintes de la théorie de la polarisation dans un algorithme basé sur les données. Une évaluation qualitative et quantitative des résultats démontrent l'intérêt de l'utilisation d'un imageur RGB-polarimétrique grâce à l'apport des informations de polarisation. Lors de ces travaux de recherche, une boîte à outils logiciels complète a également été développée proposant ainsi à la communauté scientifique un logiciel d'accès simplifié à l'imagerie polarimétrique.

Contents

List of Figures

List of Tables

List of Abbreviations

AoLP Angle of Linear Polarization. ix, x, xiii, 6, 11, 22–24, 26, 29, 36, 39–43, 45, 47, 49–52, 57, 61, 63, 79, 82, 85, 88, 92, 93, 96, 97, 102, 106, 109, 111, 113, 120, 133–135, 137–141, 144, 150, 151, 153, 155–157, 159–161

AoP Angle of Polarization. 9, 13, 58, 59

BRDF Bidirectional Reflectance Distribution Function. 26, 27

CCF Camera Coordinate Frame. x, 60–62, 69, 70

DoA Division of Amplitude. 13, 14

DoFP Division of Focal Plane. i, 4, 5, 13–15, 31, 34, 35, 37, 52, 53, 61, 100, 118, 133, 143, 144

DoLP Degree of Linear Polarization. ix, x, 4, 18, 19, 22–24, 26, 29, 36, 39, 43–45, 47, 49–51, 57, 61, 77–80, 82, 85, 88, 92, 93, 97, 98, 106, 109, 113, 114, 119, 120, 133, 144, 150, 151, 153, 156–161

DoP Degree of Polarization. 9, 13, 58, 59

DoT Division of Time. 13

FoV Field of View. 52, 53, 153

GAN Generative Adversarial Network. 19

GUI Graphical User Interface. 145–148, 160

HSV Hue Saturation Value. 51, 109, 150, 151, 157, 161

LCD Liquid Crystal Display. 12, 35

LCF Local Coordinate Frame. 100, 101

LiDAR Light Detection and Ranging. 27

LPF Linear Polarization Filter. 11

MAE Mean Angular Error. 76, 77

RGB Red Green Blue. 17, 18, 20–23, 26–28, 34, 39, 93, 133, 138, 144, 150, 151, 153

RMSE Root Mean Square Error. 76, 78, 105

ROS Robot Operating System. 5, 145–148

SAR Synthetic-Aperture Radar. 17

SfP Shape from Polarization. i, 4, 5, 25, 26, 30, 31, 55, 57, 63, 64, 74, 77, 80, 99, 118

SNR Signal-to-Noise ratio. 14, 79

SSIM Structural Similarity Index Measure. 84–86, 96

ULP Uniform Linearly Polarized. 39, 40, 42, 47, 49–51, 160

Acknowledgements

The journey through a PhD degree has not been an easy task to accomplish, especially in just three years. Nevertheless, it has been a valuable experience that has not only provided me with knowledge but has also instilled in me a different way of thinking and working than what I was accustomed to. Most importantly, it has allowed me to meet wonderful people who have supported me along this challenging path, and to whom I would like to express my gratitude in this chapter.

First and foremost, I would like to express my sincere gratitude to my supervision team, Dr. Olivier Morel, Dr. Lew-Fock-Chong Lew-Yan-Voon, and Dr. Renato Martins, without whom this work would have not been possible. Thank you for all the support, the guidance, the knowledge and the experience you shared with me in the field of polarization, and for taking the time to read my numerous documents. Thank you Olivier for providing me with the tools to understand the field of polarization and for your wise advice in conducting mathematical proofs. Thank you Lew for being available at any time, at any moment, to listen to me, to guide me along the research path, and to read my numerous documents. Thank you Renato for teaching me how to write a journal paper, and how to effectively present and highlight my contributions in a document.

I would like to express my gratitude to the members of the jury who played a crucial role in my PhD defense committee. I am sincerely thankful to Prof. Samia Ainouz and Prof. François Goudail for agreeing to review my thesis without hesitation in this period of high demand for PhD thesis defenses. I also want to acknowledge the effort and time put in by Prof. Stéphane Viollet as examiner. Their involvement has truly enriched the work presented in this document. Thank you all for your kindness and support.

I would also like to thank the person who encouraged me to pursue my PhD. I would

like to sincerely thank Mario Munich, who gave me the opportunity of my life to work at iRobot. With it, he opened my mind to opportunities in the robotics and computer vision field that will impact the rest of my life.

I extend my thanks to my host laboratory, the ImViA laboratory, and to both its former director, Franck Marzani, and its current director, Stéphanie Bricq. Additionally, I appreciate the dedication of the current directors of the VIBOT team, Olivier Laligant and Christophe Stolz. They have consistently listened to the needs of us, the PhD students, and their ongoing efforts to continue to improve our scientific experience and culture in the laboratory are deeply valued.

I would also like to express my sincere gratitude to my master's professor, coordinator, and now colleague and friend, David Fofi. He was the one who saw my potential for doing a PhD at the ImViA laboratory, and who advised me to go for the polarimetric imaging. Thank you for taking the time to talk to me, to advise me, and to encourage me to always do more. Thank you for being both, a guide and a friend on whom I can always count. The world needs more people like you.

During this journey, I had the opportunity to meet wonderful people in Le Creusot. The robotics lab and the VIBOT team opened their space to my creativity, and they offered me friends with whom I have the pleasure to share a meal, a conversation, a beer or a good bottle of wine. Thanks Raphaël Duverne, Fabio Ribeiro, Aurelie Antoine, Herma Adema-Labille, Elisabeth Cosson, Nathalie Choffay, and Ralph Seulin. I learned a lot from you, I appreciate your sincerity, your encouragement, your jokes, and most of all, your friendship. I would like to extend my special thanks to Dr Yohan Fourgerolle, my Master's course professor in Computer Science. Dr Yohan Fourgerolle pushed me to do programming in a way I had never done before, and I was amazed by the results I achieved through this new approach.

I am truly thankful for the encouragement and assistance of Fabrice Meriaudeau, who has been there to guide, advise, and support me before and throughout my PhD program. Thank you for your honesty, and the time you dedicated to sharing your insights which were invaluable in helping me when I faced challenges.

I would like to convey my sincere thanks to my friend and PhD colleague Antoine Lavault, who dedicated his valuable time to listen, discuss, and guide me in the deep-learning world

with his advice. I can never thank you enough.

A doctorate student is never alone, and over the past years, I have had the chance to learn from and grow alongside my PhD colleagues and lab mates. Our conversations around a coffee have been invaluable in my development as a researcher and as a person. I would like to express my gratitude for the moments I have shared with all of you.

Living in France is a challenging experience that is far from what I have known. It brought about stress, doubts, and fear, but, these feelings were alleviated by the wonderful people who surrounded me, made me embrace the French culture, and adopted me as their own son. I would like to express my gratitude to my second family in France, Solange and Joseph Buttigieg. They opened the doors of their house, taught me about France, provided their help when I needed it, and offered their time to relax me when I was overwhelmed with my research. With them, I have shared and continue to share wonderful moments of laughter and joy.

I also want to express my gratitude to all my family, especially to my parents Felix and Susana Rodriguez, and to my brother Alejandro for their support and encouragement during this long journey. I wouldn't be here, at this level of studies, in this part of the world, without their advice and guidance. They also listened to me when I complained about my problems, how things were not working, and how the research was not progressing as I would like it to be. Many times, discussing with you, and listening to your ideas opened my mind about what to do next. I would also like to say thank you to my friends in Argentina and around the world. Despite the distance, they knew what to say to cheer me up. Thank you Juan Pablo Vecchio, Gonzalo Asad, Yanina Menchon, Pablo and Lucas Grigolato, and Betty Callens. I am the luckiest person to have your friendship.

Last but not the least, I would like to thank my beloved Sylvia. Thank you for being there each time I wanted to quit, each time that I needed those warm words to cheer up, and see the light at the end of the tunnel. Thank you for taking the time to help me out when I needed it, to encourage me with your ideas and thoughts, and to organize outings to clear my mind. I would like to thank you for the efforts you have made to try to understand my complicated topics and expressions to let me rethink my problems differently. The results obtained so far would not have been the same without you. Even though not conventional,

I would also like to thank you for being the mother of our child who will soon be born, and who is a driving force for me to finish this PhD in a timeframe that nobody would have believed it was possible.

Chapter 1

Introduction

In this new era of computers and informatics, robots are becoming more present in our every day lives. When we mention robots, we do not talk particularly of human-like machines, with eyes, arms, legs, and cognitive capacities alike us, but in a broader sense of the term. Any type of machine that is able to understand its environment, or interact with it, can be considered as a robot. Robots have sensors to measure variables of interest of the environment, actuators to perform actions based on the measured parameters, and algorithms to bound both actuators and sensors. Each aspect of the robot defines a domain of study in itself: the structural design (disposition of sensors, and material selection), depending on the intended application, the electronics components, the algorithms that deal with a specific sensor, the integration of all the sensors' information to perform the task (for instance, the navigation of the robot under certain environmental conditions). Of all the possible sensing modalities, vision is one of the richest sources of data due to the large amount of information we can extract from it to develop different applications. For instance, by using a camera we can detect and classify the different objects present in a scene observed by an autonomous car, read and interpret the information shown on a traffic sign, compute the trajectory of a robotic arm to grasp an object in a manufacturing process, extract the 3D shape of an object and inspect its quality, interact with a robot by making gestures with the hands or the face, estimate the distance to different objects, and we can also help a mobile robot to navigate in an unknown environment. And the list can continue. The domain of Computer Vision is the one that deals with all these problematics, looking for new and efficient methods to

perform these tasks by fitting robots with cameras.

For most applications, a conventional color camera is used. This type of sensor mimics the way the human being sees the world: only the visible spectrum is considered, and key information is extracted from texture changes in the image. Additionally, the fact of having two eyes, and not one, allows us to measure the depth to the objects. Theoretically speaking, with only one eye, this is not possible, but our brain is already trained with plenty of real-life situations, thus it is able to estimate the distance to the objects even with a single view. There exist algorithms based on color-only cameras that produce outstanding results, and in some cases, the performance obtained is even better than what a human can achieve. However, there are recurrent problems that challenge conventional imaging techniques. If no texture is present (for instance, a mono-color surface), no information can be extracted from it with a color camera. Furthermore, highly reflective regions generally saturate the pixels, and they do not allow the sensors to capture hidden patterns under the reflection. Moreover, if the object is transparent, reflections over it may create false positives when using texture-based algorithms.

Nonetheless, we are not constrained to work exactly as the human eyes, and that is why there exist cameras that can even measure parameters that humans cannot see, such as the infrared light, or the polarization state of the light. Each type of variable a camera is sensitive to, is called a modality, and if a camera is able to sense several variables or modalities, it is called a multi-modal camera. In this thesis, our focus is on the computer vision field, by using a multi-modal camera that can measure both, the color and the state of polarization of the incoming light. In simple terms, the polarization state of the light describes the way the light moves as it travels through space. When this light reaches the camera, it can come directly from a source of light, or result from reflection of objects. In the last case, the light can provide information about the object's shape and material composition, and the direction of the source of light. This key information is useful to improve any of the above mentioned applications of color cameras, since in any case we have the color information, and then we are adding another parameter to the system that can help to decide when the original system cannot. For instance, in autonomous robotics, the polarization camera can see through fog, or rain, and it can avoid glare by blocking the light reflections from

surrounding objects. Furthermore, the polarization information can be used to analyze the sky reflections coming from the sun since this light is highly polarized. Moreover, the color of the sky depends on the distance that light must pass through the atmosphere and this distance depends on the position of the sun. As a consequence, using triangulation, it is possible to estimate with high precision, the orientation of our camera, relatively to the sun. Additionally, regarding biomedical studies, it can be shown that a polarized light reflected over the skin contains almost the same polarization as the source, and the phase shift depends on the depth to which the light has travelled before being reflected. The deeper it goes, the larger is the phase shift. This phase shift will change based on the skin composition, thus the polarization can be used to analyse skin diseases. Another field in which the polarization state of the light finds its place is atmospheric remote sensing. In this case, a polarimetric camera, combined with multi-spectral cameras can be used to measure the types of particles present in the air, to assess the health hazards of aerosols, and to probe volcanic ash clouds. Finally, since the polarization properties depend strongly on the surface shape, orientation and roughness, a polarimetric camera can be used to detect features of objects that have similar spectral characteristics as the background.

Despite the advantages of the polarization information, the amount of work in this domain is still limited and it is not growing as rapidly as it is for the conventional color cameras: there are no common benchmarks nor standard software toolkits to compare algorithms, the existing calibration methods are not easy to implement due to the need for a large set of equipment, and even though polarization provides additional information, there is not a common method to integrate this new data to an already existing pipeline for color cameras. In this thesis we address some of these points, and we expect to attract more researchers to this interesting domain of polarization imaging.

1.1 Main contributions

The work presented in this thesis aims to investigate the usage of the polarization data to improve the perception capabilities applied to robotics tasks, notably in scene depth reconstruction. In this sense, we provide the following contributions.

Firstly, we introduce a calibration method for polarization cameras based on the Division of Focal Plane (DoFP) sensors which are the most used type of sensors in the robotics domain. This is because this technology enables real-time applications by capturing all the required information to estimate the polarization state in a single snapshot. The sensor calibration is an important step in any computer vision application since it allows to ensure that the measurements are correct, and bounded by a known error interval. By correctly fitting a model to the device we want to calibrate, we account for the possible manufacturing imperfections and non-idealities that a system can have. In our case, each pixel of the sensor contains two filters: a color and a polarization filter. To compute the polarization state of the light, the exact orientation of the filters needs to be known for each pixel in the sensor. The manufacturer gives a nominal value for those orientations with a certain tolerance, thus the only way to know their exact value is through calibration. The novelty of our calibration algorithm is that it allows the user to reduce the complexity of the calibration setup by using the same camera to estimate the calibration light parameters. The only knowledge about the light is that it should be uniform, and have a moderate value in each primary color frequency. Then, by taking at least five samples of the source light, the calibration problem can be solved. The proposed algorithm accounts for both, vignetting and manufacturing problems since it fits a pixel model such that the polarization answer is flat for all the sensor surface. Therefore, we correct errors in the sensor filters orientations, non-ideal filters response, and differences in pixel gain or in measured light intensity due to lens distortion.

Next, we present a complete formulae for the Shape from Polarization problem, which has not been previously presented in the literature, to the best of our knowledge. The only cases covered are either for metallic or insulators, and either for diffuse or specular reflection. In this thesis, we consolidate all of them in a single document, and we provide the different considerations related to material type and reflection. Moreover, we present the inverted functions of the Fresnel equations that relate the DoLP with the zenith angle of the normal vector to the surface. Even though for the specular reflections the original function is not invertible, we present an alternative representation that provides the two solutions to the equation in a closed-form.

As a final contribution, we present our works in the field of depth of the objects in the

scene with respect to the camera coordinate frame by means of a deep-learning network. This application is based on an already existing method, and our objective is to develop a methodology to integrate the polarization information to it to improve the original color-based results. We propose a design to integrate and to fuse the two modalities (color and polarization) by a middle fusion technique. Additionally, by using the geometry constraints given by the polarization theory, we learn the network to decide which modality gives the most valuable information to correctly estimate the depth of the different objects in the scene.

As part of the work done for this thesis, we present an entire toolkit that enables alike usage with all the DoFP cameras available on the market. We have developed a software suite composed of two parts, a server and a client. On the server side, the raw images coming from the camera are retrieved and sent through a network socket. On the client side, as soon as a new image is received, it is pre-processed, and shown. The server side makes use of the Robot Operating System (ROS) middleware, which enables easy acquisition of a series of images through their standard tools, and it also allows straightforward integration of this camera to a larger robotics system. The client side also makes use of this middleware but it does not need to know which DoFP polarization camera is being used. This software allows to manipulate the camera parameters, to apply the different polarization algorithms to the raw images and to calibrate the camera and analyze the result of the calibration.

1.2 Thesis organization

This thesis is organized as follows. In Chapter 2 we present the polarization basis, starting from the physics of the light, up to the equations that relate the normal vectors to the polarization measurements. Then, we present the sensing modality used to estimate the polarization state, and a review of the literature regarding the applications in which the polarization imaging is commonly used. In Chapter 3 we explain the calibration algorithm we developed in this thesis, with the results that validate its quality. In Chapter 4, we present the SfP basis theory for the different possible configurations of materials and reflections, and we detail a pipeline to evaluate the calibration algorithm with this concrete application

of the polarization state. This application serves to stress the fact that the calibration step is required to make correct use of the polarization measurements. In Chapter 5 we present a Deep Neural network we have developed to estimate the depth with a single image. We explain its architecture, the input encoding, and the loss that takes into account the geometry constraints given by the polarization theory. Finally, in Chapter 6 we present our conclusions of this thesis, and our perspectives and future works. As complementary works, in Appendix A we include the demonstration of the error introduced by the Angle of Linear Polarization estimator used for our calibration algorithm, and in Appendix B we show the developed software to interact with the polarization camera, and its capabilities.

1.3 Publications

As part of this thesis, we have have produced the following journal articles:

- J. Rodriguez, L. Lew-Yan-Voon, R. Martins and O. Morel, "A Practical Calibration Method for RGB Micro-Grid Polarimetric Cameras," in IEEE Robotics and Automation Letters, vol. 7, no. 4, pp. 9921-9928, Oct. 2022, doi: 10.1109/LRA.2022.3192655.

- J. Rodriguez, L. Lew-Yan-Voon, R. Martins and O. Morel, "Pola4All: A survey of polarimetric applications and an open-source toolkit to analyze polarization" in SPIE Journal of Electronic Imaging, 2023 (Accepted for publication)

Additionally, we have presented the software introduced in the paper "Pola4All: A survey of polarimetric applications and an open-source toolkit to analyze polarization" in the National French Conference ROSConFr 2023 on July 2023 at Bordeaux, France. We have also participated in another project related to multi-modality sensing methods. The result of this work has been published in the proceedings of the ASAPI2023 conference:

- T. Clamens, J. Rodriguez, M. Delamare, L. Lew-Yan-Voon, E. Fauvet and D. Fofi, "YOLO-based Multi-Modal Analysis of Vineyards using RGB-D Detections" in proceddings of Advances in Signal Processing and Artificial Intelligence, June 2023.

Chapter 2

Polarization imaging and applications

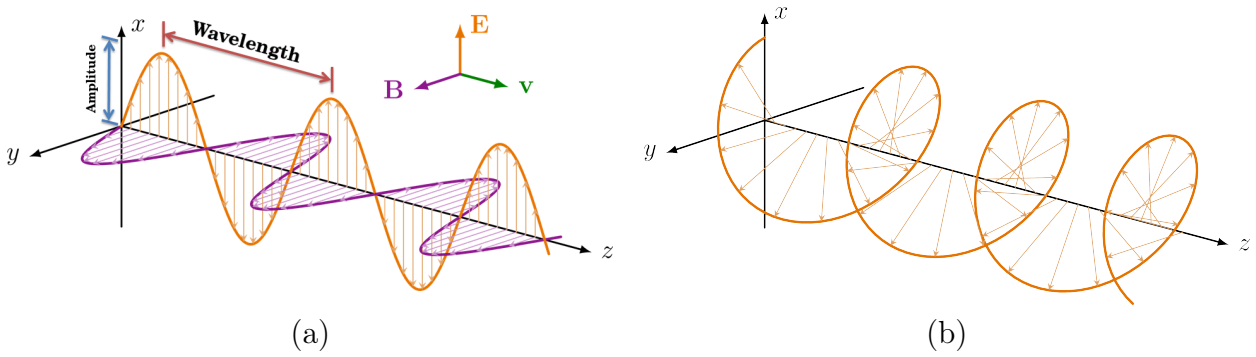
In this chapter, we present a detailed mathematical model of the polarization state of the light that we will use in the entire manuscript. Then, we introduce the different imaging sensors used to capture the polarization state of the light. Finally, we will end the chapter with a review of the different applications in which the polarization imaging has been used in the recent years.

2.1 Polarization theory basis

In this section, we provide an introduction to polarization theory. All the concepts detailed here will be used throughout the thesis manuscript, and we will refer to this section whenever a specific concept is mentioned.

2.1.1 Electromagnetic waves

Light is an electromagnetic wave of high frequency. Thus, all the properties of this type of wave can be used to explain the behavior of the light. A wave of this type is composed of two elements: an electric field, and a magnetic field. Both of them are described by vectors with a magnitude and a direction that are variables of the time and the position in space (x, y, z). Nonetheless, since the magnetic and electric fields are two waves that move similarly, shifted by 90° in space, we generally choose the vector of the electric field as reference. Hence, we



(a) (b)

Figure 2.1: Electromagnetic wave representing the light for two different polarization states [77]. (a) $\vec{E}$ is the electric field vector, $\vec{B}$ is the magnetic field, and $\vec{V}$ is the direction of propagation. The electromagnetic wave is the one described by the extreme point of the $\vec{E}$ vector. This light corresponds to a linearly polarized light. (b) Electromagnetic wave representation with a circularly polarized light.

describe the signal with respect to those of this vector. Then, the wave considered is the one whose shape is drawn by the extreme point of the vector arrow, and we characterize the electromagnetic wave by the properties of this wave. Thus, this wave that propagates through space can be defined by its amplitude, its frequency, and the way it moves as it travels. These three properties are illustrated in Fig. 2.1.

In this figure, $\vec{E}$ is the electric field vector direction, $\vec{B}$ is the magnetic field direction, and $\vec{V}$ is the direction of propagation of the wave. Taking into consideration the wave described by the $\vec{E}$ vector, the height of the wave represents its amplitude, and the length of one period of the wave is its wavelength. If we look at this electromagnetic wave as light, the amplitude of the wave is equivalent to the brightness of the light, and the wavelength is equivalent to its color. Furthermore, the way it moves transversely to the propagation direction defines its polarization state. Let us consider the projection over a plane perpendicular to the propagation direction of the extremity of the arrow vector of the electric field $\vec{E}$. The light is said to be linearly polarized if this projection gives a line. It is said to be circularly polarized if the projection describes a circle. These two cases are represented in Fig. 2.1 (a) and (b), respectively. If the vector moves in all directions without a preference, the light is said to be unpolarized. It is worth noting that when the light is totally polarized (either circularly, or linearly, or a mixture of both of them), the electric field vector end covers a continuous curve. Furthermore, a combination of linearly and circularly polarized light gives an elliptically polarized wave, and a light that has an unpolarized and a polarized component

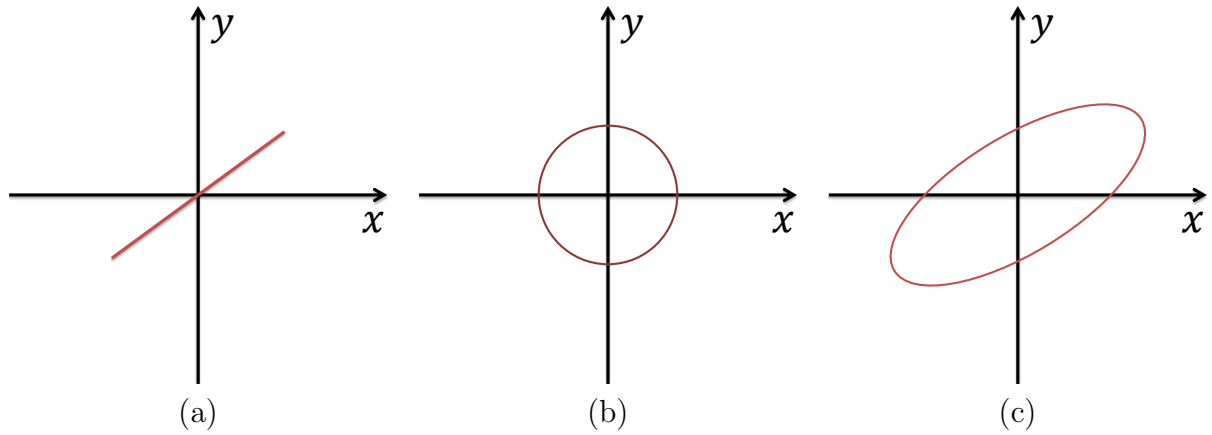


Figure 2.2: Examples of polarization states. Projection of all electric field vectors over a single plane perpendicular to the direction of propagation. The shape of this projection describes qualitatively the polarization state of the light. (a) Totally linearly polarized light. (b) Totally circularly polarized light. (c) Mixture of circular and linearly polarized light.

is a partially polarized light. Examples of totally polarized light states are shown in Fig. 2.2.

2.1.2 Polarization mathematical model

There exist several models to depict mathematically the polarization state, but the most commonly used is the Stokes model. This model defines the light wave as a 4D vector $\mathbf{S} = [S_0, S_1, S_2, S_3]^T$. S_0 represents the total light intensity (polarized or unpolarized). S_1 and S_2 describe the amount of light that is linearly polarized horizontally / vertically, and in the direction of $\pm 45°$, respectively. S_3 represents the amount of light that is circularly polarized. Using the Stokes vector, it is possible to make a representation of all the possible polarization states by using the Poincaré sphere, which is shown in Fig. 2.3.

Using the Stokes model, it is possible to define two physical parameters:

$$\rho = \frac{\sqrt{S_1^2 + S_2^2 + S_3^2}}{S_0} \quad \text{and} \quad \phi = \frac{1}{2} \arctan\left(\frac{S_2}{S_1}\right), \tag{2.1}$$

where ρ is called the Degree of Polarization (DoP) which represents the portion of the light that is polarized, and ϕ is the Angle of Polarization (AoP). This angle represents the orientation of the line segment or of the ellipse when the light is respectively, linearly or elliptically polarized. In most applications, only the first three components are used, which corresponds to the linear part of the Stokes vector. This is a common consideration since

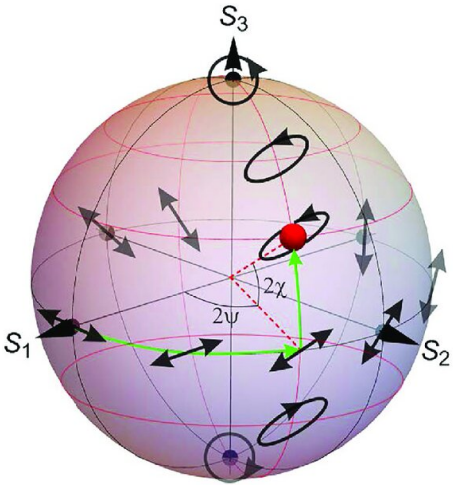


Figure 2.3: Poincaré sphere representation [4]. The surface of this unit sphere corresponds to the totally polarized light cases. The origin of the sphere represents the totally unpolarized light. All the points between the origin and the surface of the sphere represents the partially polarized light states. All the polarization states in the Equator line corresponds to the linearly polarized states. The poles correspond to the totally circularly, and all the other points are associated to the mixtures of linearly and circularly polarized light.

naturally generated light is generally linearly polarized [24]. Thus, S_3 is set to zero, and the Stokes model becomes a 3D vector. It follows that it is possible to express the Stokes vector in the function of these physical variables as:

$$\mathbf{S} = \begin{bmatrix} S_0 \\ S_0\rho\cos(2\phi) \\ S_0\rho\sin(2\phi) \end{bmatrix}. \tag{2.2}$$

2.1.3 Polarization state measurement

An advantage of the Stokes model is that the effect produced by an object (either by transmission or by reflection) over the incident wave can be modeled through a Mueller matrix [40]. More specifically, a Stokes vector $\mathbf{S}_{in}$ that interacts with an object whose Mueller matrix is $\mathbf{M}$ is converted into a Stokes vector $\mathbf{S}_{out}$ according to the following equation:

$$\mathbf{S}_{out} = \mathbf{M}\mathbf{S}_{in}. \tag{2.3}$$

If the full-Stokes vector is used, the matrix $\mathbf{M}$ has a shape of 4×4 elements, and if only the linear part of the Stokes vector is used, then this matrix has 3×3 components. Until now, the only way to measure the Stokes vector components is by using an indirect measurement

method. This is done by taking measurements of the light after several filtering steps. For the linear components of the light, the process consists in measuring the received intensity when the light passes through a Linear Polarization Filter (LPF) at different orientations. A LPF is an optical device that allows only the waves that have the same orientation as the filter axis to pass. Any other wave is filtered with a gain that has a sine curve shape. The maximum gain occurs when the filter orientation matches the AoLP of the incident light, and the minimum gain will occur when these angles are separated by $\pi/2$ radians. An optical device as the one described is modeled through the following Mueller matrix [27]:

$$\mathbf{M} = \frac{1}{2}\begin{bmatrix} q+r & (q-r)\,C_{2\theta} & (q-r)\,S_{2\theta} \\ (q-r)\,C_{2\theta} & m_{11} & m_{12} \\ (q-r)\,S_{2\theta} & m_{21} & m_{22} \end{bmatrix}, \tag{2.4}$$

where q and r are the major and minor light transmittance of the linear polarizer, respectively, θ is the orientation of the filter, and:

$$\begin{aligned} S_{2\theta} = \sin(2\theta)\,, C_{2\theta} = \cos(2\theta)\,, \\ m_{11} = (q+r)\,C_{2\theta}^2 + 2\sqrt{qr}S_{2\theta}^2, \\ m_{22} = (q+r)\,S_{2\theta}^2 + 2\sqrt{qr}C_{2\theta}^2, \\ m_{21} = m_{12} = \left(q+r-2\sqrt{qr}\right)S_{2\theta}C_{2\theta}. \end{aligned}$$

If a camera is used to take the measurements of the filtered light, then, only the first component of $\mathbf{S}_{out}$ can be retrieved, which corresponds to the total intensity of the observed light S_0^{out}. Thereby, only the first line of the Mueller matrix $\mathbf{M}$ should be considered:

$$S_{0_\theta}^{out} = \frac{1}{2}\begin{bmatrix} q+r & (q-r)\,C_{2\theta} & (q-r)\,S_{2\theta} \end{bmatrix}\mathbf{S_{in}}, \tag{2.5}$$

where $S_{0_\theta}^{out}$ is the S_0 component of the output Stokes vector, when the filter axis is oriented with an angle of θ radians. If the filter is considered ideal, then $q = 1$ and $r = 0$, and Eq. (2.5) becomes:

$$S_{0_\theta}^{out} = \frac{1}{2}\begin{bmatrix} 1 & \cos(2\theta) & \sin(2\theta) \end{bmatrix}\mathbf{S_{in}}. \tag{2.6}$$

In general, $S_{0_\theta}^{out} = I_\theta - d$, where I_θ is the readout intensity given by a pixel, and d is the pixel offset, often called *dark current*. Most works ignore d since in commercial cameras, this value is tiny compared with the camera measurement [57]. Thus, we have in general that:

$$I_\theta = \frac{1}{2}\begin{bmatrix} 1 & \cos(2\theta) & \sin(2\theta) \end{bmatrix} \mathbf{S_{in}}. \tag{2.7}$$

To find the vector $\mathbf{S}_{in}$, several measurements at different θ values are required. In the general case, if N orientations are used $[\theta_1, ...\theta_N]$, then N intensity measurements $[I_{\theta_1}, ...I_{\theta_N}]$ will be obtained. If all the measurements are stacked following the Eq. (2.7), a linear system can be built as:

$$\begin{bmatrix} I_{\theta_1} \\ . \\ . \\ . \\ I_{\theta_N} \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 & \cos(2\theta_1) & \sin(2\theta_1) \\ . & . & . \\ . & . & . \\ . & . & . \\ 1 & \cos(2\theta_N) & \sin(2\theta_N) \end{bmatrix} \mathbf{S}_{in},$$
$$\Rightarrow \mathbf{I} = \mathbf{A}\mathbf{S}_{in}. \tag{2.8}$$

where $\mathbf{I}$ is the intensity measurements vector, $\mathbf{A}$ is called the pixel matrix, and $\mathbf{S}_{in}$ is the Stokes vector we want to estimate. Then, if $N \geq 3$, it is possible to find $\mathbf{S}_{in}$ by computing the pseudo-inverse of $\mathbf{A}$.

2.2 The polarization imaging

So far, the mathematical formalism of the polarization state of the light was introduced, which comprises the most general case. In this section, the details about how the modeled polarization state of the light can be measured by different imaging devices are explained.

The polarization of the light is present in several real-world physical phenomena. Light coming from rainbows in the sky, reflections from water on highways, and monitors and cellphones based on LCD are typical examples. Light polarization is naturally generated when an unpolarized light source (e.g., light bulbs or the sun) hits a surface and is reflected.

The polarized light generated in this way can be of two types: specular, when the reflection of the light is in a single direction; or diffuse, when the reflection is in all directions. These two types of behaviors are illustrated in Fig. 4.1. In the vision sense, the specular behavior is associated to mirror-like reflections, and the diffuse one conveys the base color information.

The way the reflected wave oscillates depends on the characteristics and the shape of the material. This relationship between the observed light and the object properties is a key feature that vision algorithms can use to improve their accuracy with respect to another one that uses only texture information. These additional features can be leveraged, for instance, to improve object detection and scene segmentation results, detect mirrors and other surfaces that polarize the light, or uniquely identify places in a room that will serve as landmarks in navigation algorithms. It is worth noting that polarization cues are the main sources of information used by many biological agents such as insects and bees for their orientation in space [33].

2.2.1 Polarization imaging sensing methods

The polarization imaging systems have been introduced since Wolff [116], in which a normal CCD camera is used jointly with a rotative filter. After taking 3 or more images with it, he was able to estimate the polarization state of the light in a generic scene. There is not a method to directly estimate the polarization parameters of the light as images. Since a pixel in a camera is able to measure the total intensity received over a window of time, the Stokes vector can be measured indirectly through filtering, and with it, the DoP and the AoP can be computed. Using this concept, several polarization acquisition techniques have been introduced, namely:

- Division of Time (DoT) [71]: The most common method used before the DoFP sensor. In this case, a filter is rotated in front of the camera, and several images of an object or a scene are captured with the filter placed at different positions. The inconvenience of this technique is that it is not adapted to capture objects / scenes that change between captures. Consequently, it is not possible to capture moving objects with it.

- Division of Amplitude (DoA) [75]: A prism is placed inside the camera, and this

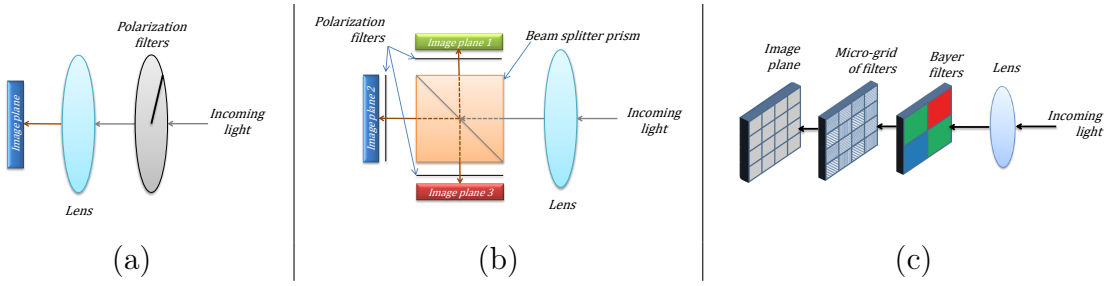


Figure 2.4: Different acquisition methods to estimate the polarization state of the light. (a) Division of time: A conventional camera is used with a linear filter in front of it that can change its orientation (either mechanically either electronically). Several images are taken, with different filter orientations, at different instants of time. (b) Division of amplitude: The incoming light hits an optical prism that splits the light beam into several directions. At each output direction, a photometric detector is placed, with a fixed linear polarization filter. Each filter is oriented differently, thus the three output images are used to estimate the polarization state. (c) Division of focal plane: In this case, each pixel contains a linear polarization filter oriented differently. By joining the measurements of four pixels in a pattern of 2×2 elements, the polarization state at those pixels is estimated. This operation can be done for all the pixels, thus a polarization image is obtained.

device splits in three the incoming light beam. Next, each beam is projected over a normal sensor doted with a linear polarization filter oriented differently from one sensor to another. Even though effective, this method comprises some disadvantages: the camera does not have a minimal dimension, the images require an alignment, and the system is too fragile since the prism can move and completely misalign the images. Furthermore, due to the split, the rays received by each sensor has less energy than the original beam, reducing the Signal-to-Noise ratio (SNR) of the sensing system.

- Division of Focal Plane (DoFP): Differently from the other methods in which all the pixels share the same filter, in the DoFP sensor each pixel has its own filter. The filter are disposed on the sensor following a fixed pattern of 2×2 pixels. With a single photo, all the required information is captured without reducing the detection intensity as in the DoA. Nonetheless, an interpolarization algorithm needs to be implemented to not lose spatial resolution in the images. This is still a challenging problem to solve particularly for the color camera, since a single sensing cell is composed of 4×4 pixels. This is the technology we used for this thesis.

The representations of the working principle of the different cameras are shown in Fig. 2.4.

In the literature, mainly two measurement systems are used: Division of Amplitude (DoA) and Division of Focal Plane (DoFP). The advantage of the second method with respect to the first one is that it can capture all the required information in a single shot: color and polarization. To do so, the sensing unit is composed of 4×4 pixels, as shown in

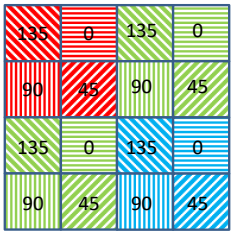


Figure 2.5: Sony Polarsens IMX250MYR RGB-polarization sensor pattern, arranged as a Bayer RG pattern with polarizer orientations of 0°, 45°, 90°, and 135° [91].

Fig. 2.5. This matrix of pixels follows a Bayer RG color pattern, where each color filter is divided in normal super-pixels. The reason of why selecting these four orientation values for the super-pixel is demonstrated in [104], in which it concludes that using equidistant angles in the range $[0°, 180°]$ optimizes the SNR of the computed Stokes vector. If such a sensor is used, the Stokes vector can be measured by using Eq. (2.8), where $N = 4$, and $[\theta_1, \theta_2, \theta_3, \theta_4] = [0°, 45°, 90°, 135°]$. It follows that:

$$\mathbf{S}_{in} = \begin{bmatrix} \dfrac{I_{0°} + I_{45°} + I_{90°} + I_{135°}}{2} \\ I_{0°} - I_{90°} \\ I_{45°} - I_{135°} \end{bmatrix}. \tag{2.9}$$

2.2.2 Real-time outdoor polarization imaging: The division-of-focal-plane sensor

The introduction of micro-grid polarization sensors, such as Sony Polarsens, boosted the research in the polarization domain since they are capable of capturing all the required information (intensity, color, and linear polarization) in a single snapshot, and they also allow measurements outside laboratory conditions. However, the number of approaches leveraging polarization for performing computer vision and robotics tasks is, unfortunately, still quite limited. For this, we found several reasons. Firstly, the Sony Polarsens sensor is available only since 2018. Thus, not too many datasets have been created with this technology so far. Before that, other DoFP sensors have been developed, but they included a micro-grid of filters placed on top of the micro-lenses of the sensor. As a consequence, there were a cross-talk effect between the pixels that increases as we approach the borders the of sensor.

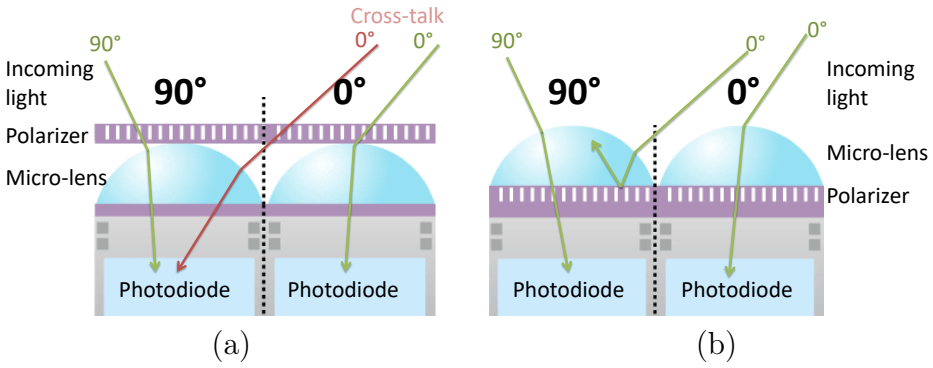


Figure 2.6: Cross-talking effect comparison between the older technology (a), and the Sony sensor technology (Polarsens) [2].

Finally, the accuracy of the sensor was acceptable only around the center of the image. Therefore, if low errors are seek with the ancient technology, the effective sensor resolution has to be drastically reduced. An sketch of this effect is shown in Fig. 2.6.

Another reason for the limitations of approaches is that, to the best of our knowledge, there is no standard method to integrate the additional polarization information to already existing texture-based algorithms. On one hand, if classical approaches are considered (as with the equations shown in Sec. 2.1 to estimate the normal vectors), a reflection model needs to be assumed, the index of refraction needs to be known, and a difference between metallic and dielectric objects needs to be done. On the other hand, when data-driven approaches are used, there is no unified model or loss function that effectively integrates the polarization information to perform its task.

It is for these reasons that we aim to open the track in this research field by adding new tools that simplify the usage of this modality, and by researching a simple but effective method to take the best of both worlds: textures and polarization.

2.3 Applications of the polarization imaging

In this section, we discuss recent advances in the field of polarization imaging within both model-based and data-driven strategies. We have reviewed the latest applications in the computer vision and robotics field that utilize polarimetry between the years 2016 and 2022. Older applications have not been considered because the most significant advancements

[2] https://thinklucid.com/tech-briefs/polarization-explained-sony-polarized-sensor/

began after the release of the first micro-grid division-of-focal plane (DoFP) sensor around 2014. Since then, DoFP sensors have been adopted and widely used as it allows real-time imaging. In 2018, Sony introduced a DoFP sensor, named PolarSens, which became the core device of many commercial polarization cameras. It offers much better quality data and facilitates the development of more performant algorithms and real-time applications beyond laboratory conditions. The works have been grouped into four representative categories and complementary tasks: image enhancement, segmentation, surface depth and normal estimation, and pose estimation. Most reviewed works focus on the application fields of computer vision and robotic vision. Particularly, we focus our study on the context of scene understanding in both ground and underwater environments Polarization has also been extensively used in the field of remote sensing, notably in combination with Synthetic-Aperture Radar (SAR) data. Applications in remote sensing are not the focus of this thesis and a good review of the recent advances of applications of polarization in this field using data-driven algorithms can be found in [65].

2.3.1 Image enhancement

In real-world applications, changes in viewing conditions can strongly impact the performance of computer vision algorithms. Thus, enhancing the quality of the visual information is often a required step to keep the accuracy of the developed computer vision system. The type of image quality improvement depends on the application itself. In some cases, this implies having high-quality measurements independently of the camera used, which can be achieved through camera calibration [27, 57, 85, 91]. In others, the improvement can be related to removing highly bright, specular reflections, requiring a separation between this type of reflection from the diffuse ones [47, 79, 106]. In more complex cases, the background structure needs to be recovered when an atmospheric phenomenon is present such as mist or fog, as shown for some examples in Fig. 2.7. In most cases, the physical constraints defined by the polarization state of the light can be used to improve the results obtained by conventional cameras.

In this context, Ono *et al.* [79] present a white balance algorithm for RGB-polarization sensors based on the achromaticity of the Stokes vector in the visible spectrum. Rodriguez

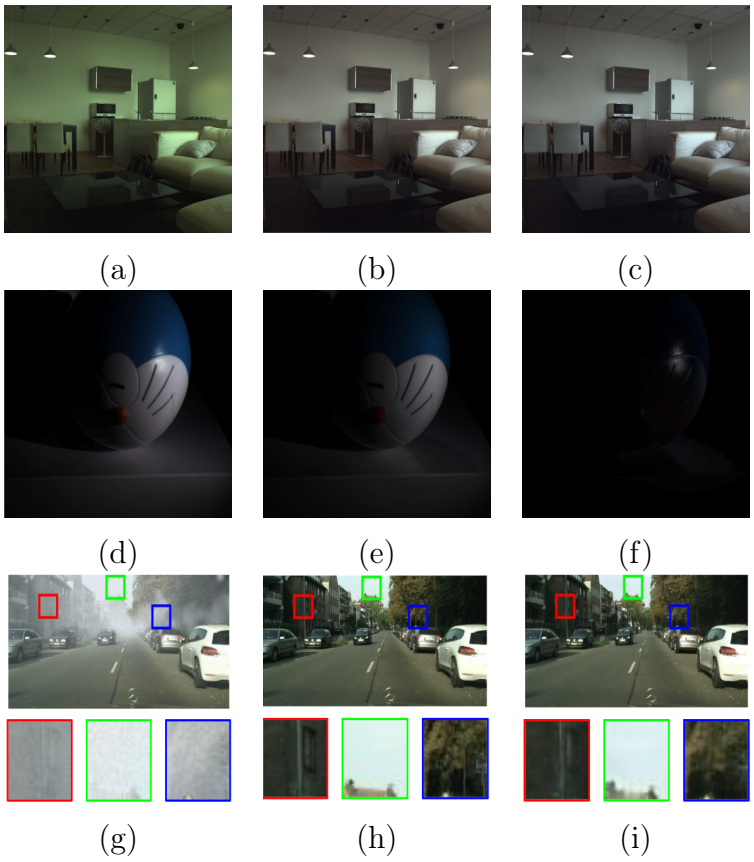


Figure 2.7: Results of some of the reviewed algorithms for image enhancement. (a)-(c) Input image, ground truth, and algorithm results of DoLP-based color constancy.[79] (d)-(f) Input image, obtained diffuse image, and obtained specular image of polarization-guided specular reflection separation.[113] (g)-(i) Input image, ground truth, and resulting images of the polarization dehazing method.[125] Images courtesy of the respective works.

et al. [91] relax the experiment setup to calibrate micro-grid color-polarization sensors to achieve a flat-field response in all the polarization parameter images. Wen *et al.* [113] solve the separation of specular from diffuse reflections with a model-based optimization strategy [15]. Their pipeline is independent of the illumination source by exploiting polarization and chromaticity images. Wen *et al.* [114] propose to jointly demosaic RGB and polarization information to obtain high-quality, 12-channeled RGB-polarization images by using a sparse representation model. The model is obtained through an optimization algorithm that uses the ADMM scheme. Similarly, Morimatsu *et al.* [76] obtain high-quality, polarization images by extrapolating the residual interpolation for RGB images [51] to the monochrome and color polarization sensors. They achieve their results by changing the guidance image so that it is edge-aware, and by making use of the raw polarization intensity measurements. Tanaka *et al.* [99] achieve better quality images by improving the condition number of

the transport matrix in comparison with conventional, passive Non-Line-Of-Sight (NLOS) system. This is done by using the polarization leakage effect model produced by the oblique reflection over a filter oriented at the Brewster angle [40] of a wall.

Using hand-crafted theories provides good results when the scene and the effect to analyze are not complex, since a high-precision mathematical model of the problem can be established. When this is not possible, data-driven algorithms can be used, as they have the capability to learn complex theories during training. For example, they can handle scenes with several objects at the same time, or model effects for which no known mathematical model exists. Data-driven approaches as described in Lei *et al.* [59] have been designed to remove the reflections produced by different types of glasses by using polarization theory. The input to their network architecture, composed of pre-trained U-net and VGG-19 networks, is an image which is a combination of the raw measurements, split by polarization channel and polarization parameters (I, ρ, ϕ). Zhou *et al.* [125] use a single polarization image and a deep learning network composed of two U-net models and two autoencoders to dehaze urban scenes. Hu *et al.* [41] developed a data-driven approach and a dataset to increase the brightness and quality of images under low-illumination conditions. They created a convolutional neural network that works in two steps based on the raw measurements of the camera: firstly, an enhancement in the intensity domain for all the color channels, and then each color channel is treated by a separate network. Liu *et al.* [68] proposed a Generative Adversarial Network (GAN) architecture to fuse the DoLP and the intensity images into a single intensity image. By dividing the image into background and foreground, the network fuses these two polarization images into a single image that has an increased and better contrast than the original intensity image. The results produced by this network can be used to train other networks, i.e., to perform an improved data augmentation, and with it, obtain models with large generalization capabilities. Despite the outstanding results of data-driven algorithms with respect to the optimization-based approaches, the quality of these results depends on the data and the type of model used. Particularly for the data, if not all the cases have been considered in the images provided during training, the missing cases might produce less accurate results during testing.

Several relevant image enhancing approaches have also been proposed to deal with the

challenging scenario of underwater imaging. Li *et al.* [64] aim to improve the contrast of underwater images due to turbidity by using polarization and an optimization strategy. They propose to split the Stokes vector into three contributions (diffuse, specular, and scattered light), since they claim that the scattering reflection underwater cannot be neglected. In the same direction, Hu *et al.* [42] present a novel CNN based on residual blocks that fuse the polarization features to restore the contrast of underwater images. By using the raw measurements of three polarization channels of a monochrome polarization camera, they are able to see through turbid water, and obtain a clear image of the hidden objects. Amer *et al.* [2] propose a static pipeline to increase the image quality for underwater applications. Based on the active cross-polarization technique, and an optimized version of the Dark Channel Prior, they achieve contrast improvement for underwater imaging, with a single snapshot in real-time. Shen and Zhao [96] developed an iterative pipeline to jointly improve the image contrast and denoise the image. With two polarization images taken with a rotative filter at 0° and 90°, they compute the transmittance and the irradiance maps in underwater conditions for each color channel. Then, they establish an iterative process to refine these results by using an adaptative bilateral filter, and an adaptative color correction routine.

Despite the remarkable improvement brought by polarization to image enhancement, as compared to similar applications for RGB-only cameras, several challenges still remain. For example, Ono *et al.* [79] outperform different baselines in many scenes, but it is left as future work to improve the results obtained when the sky occupies a large portion of the image. Similarly, Zhou *et al.* [125] retrieve the hidden structure of objects behind the haze with good accuracy in real-world situations, after training the network with computer generated images. Despite this, the authors claim that the model does not produce adequate reconstructions for fog and mist since the physical phenomena produced by these perturbations are not the same as for the haze. It is important to note that one of the barriers in the polarization image enhancement field is the lack of standard benchmarks. Indeed, most works had to create a dataset to demonstrate their contributions. Some of the created datasets, which often required a huge amount of work, can be reused as in Lei *et al.* [59] , where the authors realized acquisitions in a large variety of environments, and with different types of glasses. On the other hand, other applications have been demonstrated using small,

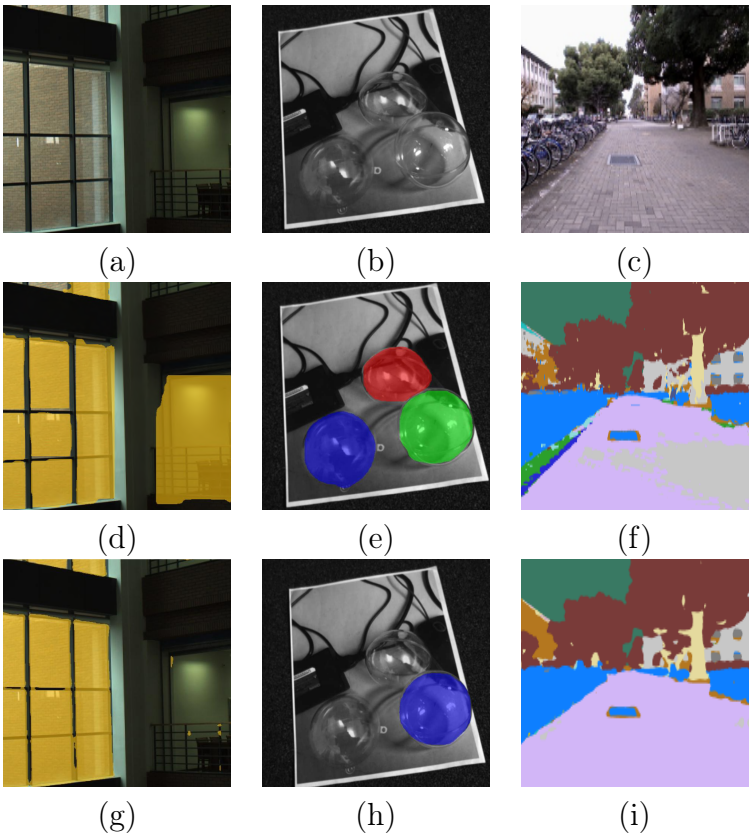


Figure 2.8: Segmentation examples with polarization cues. From left to right: (a)-(c) Input image, results obtained with an RGB-only method, and the result of glass segmentation using intensity and spectral polarization information.[72] (d)-(f) Input image with transparent objects, RGB-only method object segmentation results, and results obtained with the polarization data-driven method. [48] (g)-(i) Input image, RGB-only method results, and the results of the multimodal material segmentation algorithm.[66] Images courtesy of the respective works.

non-available datasets, that may not necessarily cover all the required cases [114], or they have created a polarization dataset based on RGB ones and a mathematical model of the polarization effect [125], which may not fit the real environment effect.

2.3.2 Image segmentation

The polarization state of the light is directly linked to the object's material and shape as mentioned in Sec. 2.1. This property can provide insightful and complementary information to guide object segmentation approaches in scenarios where only the surface color is not discriminant. Indeed, the index of refraction depends on the internal structure of the objects, and on the wavelength of the incident light as defined by the Fresnel equations [13]. It is for this reason that material classification is one of the most fruitful application of polarization theory. Previously, this task was accomplished using hand-crafted features, in controlled

acquisition conditions, using rotative filters, and considering a single object to be analyzed at a time [19, 101, 103, 115]. Nowadays, with the advances in sensors and the advent of data-driven algorithms, object characterization with polarization cues has been ported to more complex, constraint-relaxed scenarios.

In the domain of infra-red imaging, Li *et al.* [62] succeed in efficiently detecting the road area in urban scenes by using the zero-distribution prior in the AoLP and the difference in the DoLP of the objects to increase the accuracy of the segmentation. This information is further used in a visual tracking algorithm to continuously track the road online. In an extension of their previous work, Li *et al.* [63] use the zero-distribution prior of the AoLP to create a coarse map of the road. Then, they developed a deep-learning network to refine the coarse road map. Their network consists of two branches that analyze different aspects of the scene. The main branch receives the information captured by an infra-red camera, already converted into a fake color image, i.e., a 3-channel image result of stacking the AoLP, the DoLP, and the total intensity together, and then converted from the HSV to the RGB color space. The objective of this branch is to extract multi-modal features of the scene. The other branch or polarization-guided branch also receives the AoLP and the DoLP of the scene, but it does not receive the intensity image. Instead, the coarse map obtained from the zero-distribution prior of the AoLP is provided. By doing so, the authors aim to guide the network based on the polarization properties of the road, and not of the entire scene. In the visible spectrum, Xiang *et al.* [118] developed a fusion network to combine color and polarization data to better segment objects of urban scenes. They tested several combinations of polarization information with attention mechanisms and concluded that using only color and the AoLP is the best combination to improve the results. Kalra *et al.* [48] improve the instance-semantic segmentation network Mask R-CNN [50] to handle transparent objects (as shown in Fig. 2.8) by adding monochrome polarization cues to the original mid-fusion pipeline. Each polarization parameter image (intensity, AoLP, and DoLP) is fed into a different backbone encoder, and the fusion of the feature maps is performed at each encoder level. Mei *et al.* [72] extend the work presented in [48] by using RGB polarization cues, instead of monochrome, with the aim of segmenting glasses in urban scenes. Each of the two measured RGB polarization images (DoLP and AoLP)

is balanced by using an attention mechanism. Then, these two results, and the intensity RGB image are fed into three independent Conformer encoders [80] and fused using local and global guidance. In a more complex scenario, Liang *et al.* [66] build a network to fuse RGB, infra-red, and polarization cues to produce an outdoor scene segmentation based on the object material type. The proposed pipeline is composed of two core elements: a network that will classify the captured objects into one class of a subset of the segmentation labels from the CityScapes dataset [22]; and a region-based filter selection module that chooses the modality that provides the most relevant information for determining the type of material of the constitutive elements of each detected object. The full network is composed of four encoders, one for the RGB intensity, one for the AoLP image, one for the DoLP image, and one for the infra-red image.

All these works outperform RGB systems when the polarization information is added to each developed pipeline. A higher gain in performance is also often obtained when the network is adapted to correctly process the AoLP and the DoLP, and not when an RGB network is trained with polarization images. This is why most of these works propose carefully designed fusion schemes. However, the lack of datasets including polarization information in the field of image segmentation poses limitations on the development of polarization-based approaches. It is important to highlight that all the previously discussed works have presented their own dataset to show that polarization is a path to consider in image segmentation. For instance, Kalra *et al.* [48] used a private dataset acquired in a very specific environment, focused on the particular application with a pick-and-place robotic arm. Xiang *et al.* [118] provide a small-scale dataset of RGB-polarization images captured in various urban scenes. Although informative, the dataset contains only 394 annotated images segmented into 9 different classes. Mei *et al.* [72] introduce a medium-scale dataset, with 4511 images annotated only for the labels glass and no-glass. Similarly, Liang *et al.* [66] made publicly available a dataset of semantic segmentation of urban scenes, with multi-modal sensors, but it only includes 500 labeled images, and Li *et al.* [62] did it for road segmentation, and with their personalized infra-red polarization camera. Thus, there is a need for a common large-scale benchmark to evaluate the performance of these different segmentation algorithms to trace the direction toward a generalization of the polarization modality.

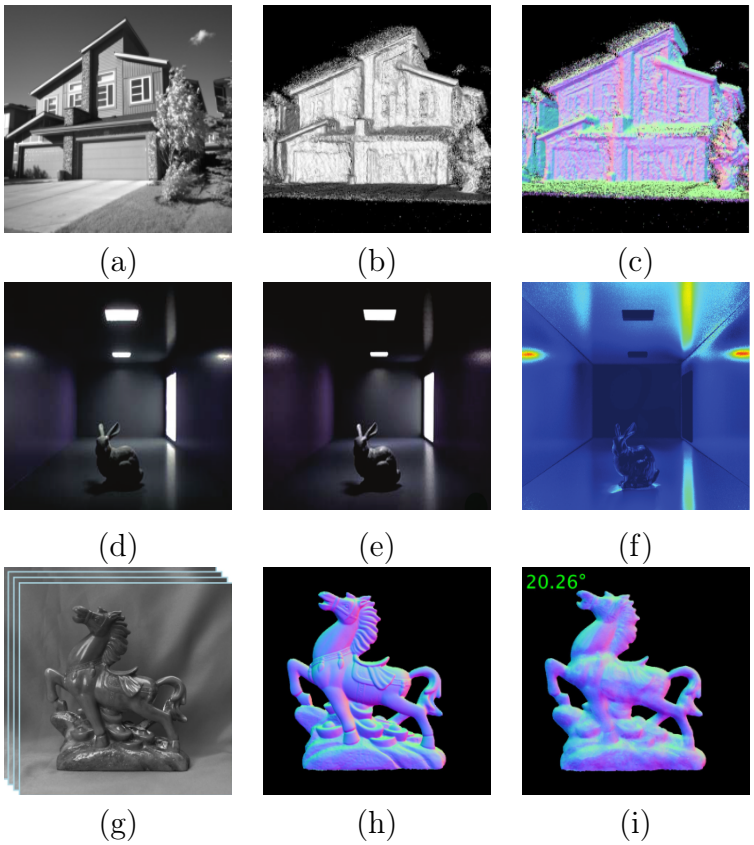


Figure 2.9: Results of some of the reviewed algorithms for surface normal and depth estimation. From left to right: (a)-(c) intensity input image, reconstructed mesh, and estimated normal map by the method introduced by [95]. (d)-(f) Real RGB image, corresponding rendered image, and the corresponding rendered degree of linear polarization. This result corresponds to the scene rendering technique implemented in [53]. (g)-(i) Input polarization images, ground truth, and estimated normal maps. These results are courtesy of [8].

2.3.3 Surface normal and depth estimation

Polarization is well known to encode shape information of the different objects being observed. The classical algorithms of Shape-from-Polarization require to make several assumptions to be able to estimate the normal vectors to the surfaces. Most existing approaches consider an orthographic projection of the incoming light to simplify the coordinate system bound between those of the normal vectors and that of the camera. Furthermore, the refractive index of the object is generally supposed to be known approximately since it is hard to run an experiment to have a ground-truth value of it. With these two assumptions the normal vectors to the surface can be estimated, and through integration, the depth map can be retrieved. These approaches are based on the Fresnel's formulae, that is detailed in Chapter 4, and it establishes the relationship between the Degree of Linear Polarization with the zenith angle of the normal, and the Angle of Linear Polarization with the azimuth angle.

Method	**Type of Algorithm**	**Object type**	**Reflection type**	**Light condition**	**Ambiguity handled**
Ba *et al.* [8]	Data-driven	Dielectric	Both	Single source, unknown	Both
Berger *et al.* [10]	Optimization	Dielectric	Specular	Non-controlled	Azimuth
Blanchon *et al.* [11]	Data-driven	Both	Specular	Non-controlled	Azimuth
Deschaintre *et al.* [26]	Data-driven	Dielectric	Diffuse	Controlled, frontal-flash	Azimuth
Fukao *et al.* [31]	Optimization	Dielectric	Both	Controlled, known direction	Azimuth
Ichikawa *et al.* [45]	Optimization	Dielectrics	Both	Uncontrolled, under clear sky	None, it uses Mueller calculus
Kondo *et al.* [53]	pBRDF formulation	Dielectrics	Both	Known positions and Stokes	Both
Lei *et al.* [60]	Data-driven	Both	Both	Uncontrolled	Both
Shakeri *et al.* [95]	Optimization	Dielectrics	Both	Uncontrolled	Both
Smith *et al.* [97]	Optimization	Uniform Dielectric	Both	Uncontrolled, under sun	Both
Zhao *et al.* [124]	Optimization	Dielectric	Both	Uncontrolled	Azimuth
Zhu and Smith [126]	Optimization	Dielectric	Both	Point source, known direction	Both

Table 2.1: Comparisons of hypothesis used in the different SfP and depth estimation methods.

Even though effective in several works and scenarios [7, 73, 74, 121], the fact that both the object's refraction index and the light direction must be known makes this approach limited to laboratory strict conditions. Additionally, the relationship between polarization and the normal vector has geometric ambiguities. Therefore, one important research direction is to reduce the constraints and priors required for the acquisition while maintaining low reconstruction errors.

Ba *et al.* [8] propose a learning-based approach to estimate the normal map of objects as shown in Fig. 2.9. The ambiguous normal maps from Fresnel's theory are used as priors given directly to a deep neural network as inputs with the objective that the network will learn to disambiguate them. Fukao *et al.* [31] present a shape from polarization algorithm that uses a stereo pair of polarization cameras. The coarse map from the stereo vision is refined by filtering the normal maps with a belief propagation scheme. They exploit Fresnel's equations, and an improved modeling of the micro-facet reflection effect by considering that it is a linear combination of the diffuse and the specular lobe reflection. Similarly, Ichikawa *et al.* [45] relaxed the constraints for shape from polarization by using the Rayleigh [98] and the Perez [81] models to estimate the sun polarization state and direction on a clear day. Then, through mathematical optimization, the normal and shading maps are obtained.

The additional cues about the incident Stokes vector serve to determine how the object modulates the incident light (Mueller calculus), and jointly with the shading constraints, the normal map can be estimated. Deschaintre *et al.* [26] propose a 3D object shape estimation, jointly with a spatially variable BRDF model estimation, by using a single-view polarization image fed to a U-Net based network architecture. The full input of the model is the intensity image, the normalized Stokes map, and the normalized diffuse color which encodes the object reflectance information. Lei *et al.* [60] propose a deep-learning network to estimate the normal map of complex scenes. Their aim is to improve the accuracy limits by incorporating viewing encoding as input to the network, which accounts for the non-orthographic projection. This input is an image where each pixel represents the direction of the incident light. When estimating the normal vectors from polarization under the orthographic assumption, all the incident light rays are supposed to be colinear to the Z axis of the camera coordinate frame. Thus, all the zenith angles are measured with respect to a common coordinate frame. When using a perspective lens, the zenith angle given by the polarization theory is measured with respect to the direction of propagation of the light, which in this case, will be different for each pixel. By providing the viewing encoding, the authors claim the network will understand the viewing direction of the polarization state and use this information to improve the results of a network that works under the orthographic assumption with a perspective lens. The other inputs to the model are the raw measurements of the camera separated by polarization channel, the AoLP, the DoLP, and the total intensity. Their network is also grounded by an architecture similar to a U-Net model, with a multi-head self-attention module in the bottleneck. Smith *et al.* [97] define the Shape from Polarization problem as a large linear system of equations. They combine the physics theory of polarization with the geometry of the problem to formulate the depth equations directly, without passing through the computation of the normals. Berger *et al.* [10] present a depth estimation algorithm that uses the polarization cues in a stereo vision system. They improve the correspondence matching by adding the AoLP-normal constraint to the intensity similarity function. In the same direction, Zhu and Smith [126] propose a hybrid RGB-polarization acquisition system to obtain a dense depth reconstruction. By classifying the pixels into specular or diffuse, they make use of normal vectors obtained

from Fresnel's theory to improve the estimation of the normal maps obtained from the stereo images. Blanchon *et al.* [11] extend the monocular depth estimation network Monodepthv2 [36] to consider polarization information by adding the azimuthal constraint to the deep-learning loss. Zhao *et al.* [124] extend the multi-view reconstruction system [52] by adding polarization cues to the optimization. They introduce a continuous function that has four minimum values, each of them at one of the ambiguous normal azimuth possibilities of Fresnel's theory. Kondo *et al.* [53] developed a polarimetric BRDF model that does not constrain the illumination nor the camera position during acquisition. This model is used to synthesize polarization images out of RGB images, easing the dataset creation for data-driven algorithms. By acquiring images with different illumination of known Stokes vectors, they use Mueller calculus to model the object reflectance. Shakeri *et al.* [95] produce a dense 3D reconstruction by using polarization cues. This is done by optimizing an initial depth map obtained from MiDaS [88], and the coarse depth map from COLMAP [94]. The optimization routine constraints the normals with the ones from Fresnel's theory. The initial depth map is used to disambiguate the polarization normals, and the coarse map is used to regularize the optimization routine, since they are metrically correct but sparse.

The previously presented works have all been developed for shape/depth estimation while leveraging the polarization information. Combining the polarization state of the light with any geometry problem developed for the RGB space results in a significant improvement in accuracy and image quality. Since the polarization measurements are provided pixel-wise, then the normal constraints are dense. Thus, passive, high-quality far-field 3D reconstructions can be retrieved using a multi-modal RGB-polarization camera, which cannot be done with active sensors such as LiDAR or Microsoft Kinect. However, these polarization constraints are still often dependent on knowing priors about the material (metallic vs Dielectric) and the reflection type (specular vs diffuse), thus sometimes they can poorly perform in the wild. To overcome this problem, some works decide to use only diffuse reflections [26], or to classify the pixels into either diffuse dominant or specular dominant (such as in [95, 97]). To deal with more complex cases, a better modeling of the reflection effect might be required [31]. For data-driven algorithms, this field also suffers from the lack of large-scale datasets that can be used as benchmarks for research. Most papers propose their own dataset by

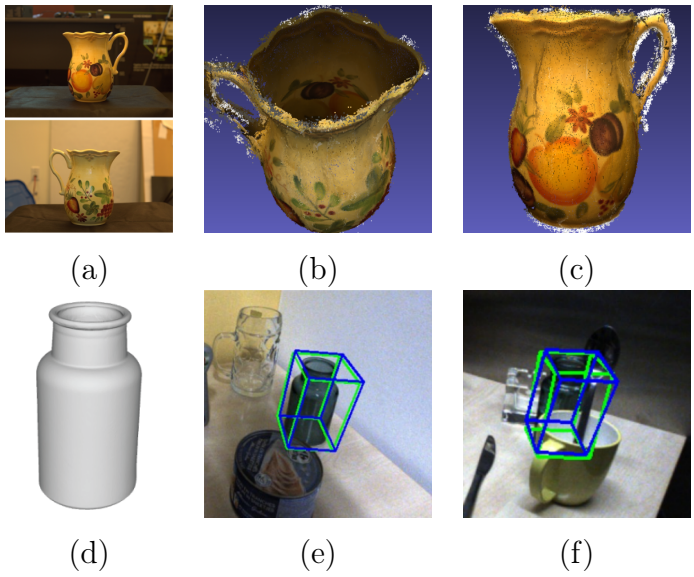


Figure 2.10: Results of some of the reviewed algorithms for pose estimation. From left to right: (a)-(c) input RGB image, reconstructed surface with the algorithm provided in [25]. (d)-(f) Glass vase model and the detected pose from two viewpoints. Results courtesy of [32].

doing acquisitions [60], or they model the entire light behavior assuming artificial conditions to simulate the polarization state over already existing RGB images [53].

In summary, the polarization clearly provides valuable cues in the field of shape estimation and 3D reconstruction but two main problems need to be addressed. Foremost, the lack of large-scale, standard evaluation benchmarks, that hinders the development of techniques using this modality in the current era of data-driven algorithms. On the other hand, there is no generic model that can effectively handle generic types of reflections over any type of material. Therefore the challenge of interpreting the measured data and determining which model to use remains open.

2.3.4 Pose estimation

The polarization of light can also play a significant role in object pose estimation since it provides valuable geometric constraints in the determination of the vectors normal to the observed surfaces (as discussed in Sec. 2.3.3). Hence, this additional information can be used to overcome the ill-posedness condition of many RGB problems, such as estimating the relative rotation and translation of textureless objects between two images. Particularly, these additional constraints are useful when the objects to analyze are highly reflective

and translucent since the polarization measurements are independent of the intensity of the light. Cui *et al.* [25] use the normal vectors estimated from the Fresnel equations to add geometric constraints for pose estimation (some visualizations of the pose estimation can be seen in Fig. 2.10). With this additional information, only two corresponding points in two views are required to estimate the rotation matrix and the translation vector. In the same direction, Gao *et al.* [32] propose a data-driven algorithm to find the pose transformation of an object in the image with respect to the camera coordinate frame. The algorithm uses three ambiguous normals as inputs to one encoder, and the polarization parameters into another. Then, the features are fused at different levels, and given to the decoder. Tzabari and Schechner [105] present a static approach in which they use the AoLP and the DoLP to expand the optical flow theory. This new component accounts for the rotational speed estimation, which cannot be done with the classical optical-flow approach. Hu *et al.* [43] utilize a monochrome polarization camera to build a complete pipeline to estimate the sun's position based on the DoLP and the AoLP measurements underwater. Jointly using the Snell and Fresnel theories, they revert the ray bending caused by the change in medium and handle the problem as if the measurements were done outside the water. Similarly, by using the Rayleigh model of the polarization pattern given by the sky, Kronland-Martinet *et al.* [55] developed a bio-inspared algorithm to estimate the North Celestial Pole (NCP) with a polarimetric camera and a fish-eye lens. This reference point is calculated based on the DoLP measurements over three instant of time, with which the intersection point of the invariance axes of the DoLP is obtained. Zou *et al.* [127] push forward the accuracy in the human shape and pose estimation by building a two steps network with polarization cues. Assuming the human cloth to be diffuse dominant, they retrieve the human features by using the raw polarization intensity images, and the ambiguous normal maps obtained from Fresnel theory. The first network produces a high-quality normal map, and the second one uses this result, jointly with the output of the SMPL human shape model [70] to estimate the final shape and pose of a clothed person.

Due to the geometric nature of the pose estimation problem, the polarization state of the light provides valuable clues that can be used in any computer vision algorithm in this field. The applications included in this section of the review demonstrate the potential of

accuracy gain obtained with the polarization constraint, while at the same time relaxing the other algorithms hypothesis. In [25], it is shown that only two points are required to estimate the pose transformation between two views, resulting in an improvement in speed and accuracy when added to any structure from motion algorithm. Without any requirement in the type of clothes to be used, Zou *et al.* [127] were able to estimate the human pose with a lower error and fewer constraints than competitors. In underwater applications, Global Positioning System (GPS) signals cannot be used because their intensities decrease rapidly with the depth in the water. To address this issue, Hu *et al.* [43] propose an autonomous underwater navigation system that uses polarization instead of a GPS signal. In their system, the camera's global position is estimated by applying geometrical constraints that link the sun's position to its known trajectory [29, 61]. However, several limitations still remain when doing pose estimation with polarization. For example, in [32], only the position of one object can be done each time, while others adopt known object materials and physical properties. Furthermore, most algorithms only consider one type of reflection (either diffuse or specular), which limits their generalization to any type of scene.

2.4 Conclusions

In this chapter, we have reviewed the basis of the polarization state mathematical modeling through the Stokes vector, the sensing techniques, and the most relevant applications of the last years. From this last part, we have seen that there is a larger community working in the development of techniques oriented to the depth estimation by including polarization cues than in the other domains. This is expected since the type of information given by the polarization state is mainly geometrical, thus its use is directly related to the outcome of a depth estimation algorithm. On the other hand, we can also conclude that almost none of the published articles make use of a sensor calibration algorithm, or at least, it is not mentioned on them. This is an important step when developing a a vision algorithm since the results can be biased by the particular sensor and lens used to do the data acquisition. Additionally, we have observed that there is a lack of details in the algorithms related to the SfP algorithms. Indeed, there is no article that makes a full description of the problem, or the evaluation of

the consequences of assuming either a single reflection model, or a single type of material. In the following chapters of this thesis, we address these problems by providing a sensor calibration algorithms that reduces the material and procedure requirements to its bare minimum, and by writing the SfP algorithm in its complete form. Additionally, we evaluate the results obtained when considering both reflection models, the effects of considering only one of them, and the improvements introduced by the calibration algorithm. Furthermore, we introduce a deep learning network to estimate the scene depth from monocular images, and a complete toolkit to capture, analyze, and process polarization images from any DoFP camera.

Chapter 3

Polarimetric sensor with lens calibration

Camera calibration is a very important procedure to be included in any computer vision algorithm. It allows for compensating the errors due to manufacturing and the addition of any other device placed between the variable to measure and the sensing unit. This procedure enables the use of the camera as a measuring device. It also allows any computer vision algorithm to be used with any camera. Indeed, after calibrating our measuring device, we are independent of any particularity of our camera or our lens. In this chapter we introduce a new calibration algorithm pipeline to improve the sensor measurements of a color-polarization sensor when it is used with a lens. We do a complete description of the method, and we provide exhaustive experiments that justify its validity.

3.1 State of the art

3.1.1 Motivation

In the computer vision field, the most widely known and used calibration algorithm for perspective cameras [83] is the geometric calibration algorithm. Since the camera is composed of pixels, which are the sensing units, and a lens that projects the light over each pixel, there are some imperfections we need to account for. Indeed, the lens will produce non-linear deformations in the object's aspect present in the image due to the modifications it introduces in the light path. Furthermore, for this type of system, the projection point

of the lens will not be placed at the center of the sensor, the exact distance between the projection point and the image plane is not known, and the distortion coefficients for the lens deformation are also unknown. To estimate the parameters of the model for a system as the one described, a calibration procedure needs to be executed. For this configuration (photometric sensor and perspective lens), the Zhang's method [122] is generally used, since it produces good estimates of the camera parameters with a small calibration setup (only some images of a checkerboard taken with the camera to calibrate are required).

Similarly, the polarization camera needs to be calibrated before being used as a measuring device. In our case, a DoFP camera has a sensing unit composed of super-pixels. We remember from Sec. 2.2 that a super-pixel is a set of 2×2 pixels, where each of its components has a linear polarization filter on top of it. Additionally, each filter is oriented differently, according to the following set of orientations $\alpha_i = \{0°, 45°, 90°, 135°\}$, where i is the pixel's position. For this sensor, the calibration algorithm will consider the fact that these filters are not ideal, the orientations are not exactly the ones provided by the manufacturer, and the lens will affect also the polarization state of the light source. Therefore, a model of the measurement unit is required, and this model will account for these deviations from an ideal polarization state analyzer. The objective is that, after correction, two pixels that receive the same light intensity will provide the same polarization measurements.

In this chapter, we will detail the calibration procedure that we have developed. The aim of this work is to reduce the calibration setup to its bare minimum without requiring strict laboratory conditions so that it can be applied by a wider public. Indeed, by using a uniform light source and a linear polarizer, the calibration procedure can be carried out without any knowledge about the polarization state of the source light. Only a few samples of this source light are enough to get good-quality results.

3.1.2 Previously developed calibration algorithms

Diverse calibration methods have been developed and reported in the literature to correct polarization measurements. However, they are either not suitable for a camera based on an RGB polarization sensor or they require complex equipment, making it hard to replicate the experiments. For example, the method developed by Schechner [93] considers a

Method	Requirements				
	Known Stokes	Using integrating sphere	Band-pass filter	Motorized turning filter	Polarization filters used
Hagen *et al.* [38]	✓		✓		1
Chen [20]	✓	✓	✓	✓	1
Ding *et al.* [27]			✓		2
Powell and Gruev [84]	✓	✓	✓	✓	1
Proposed					1

Table 3.1: Comparative table of the requirements for the DoFP calibration methods available in the literature.

conventional camera with a polarizer filter in front of it. In this setup, all the pixels share the same polarization filter and thus, to solve the calibration problem, only a few polarized points in a generic scene are needed. This is not the case for a DoFP sensor where each polarization analyzer is composed of four different pixels with polarizers oriented in four different directions. The method by Wang *et al.* [111] uses an LCD screen to achieve both, polarimetric and geometric calibration of a camera mounted with a polarization filter. Nevertheless, this method cannot be used in our case, because to illuminate all the pixels with this type of screen, the sensor to the LCD distance must be so short that the pixel pattern of the screen is captured by the camera. Thus, the light cannot be considered uniform. Regarding calibration algorithms dedicated to DoFP sensors, Hagen *et al.* [38] proposed a method that requires a few samples only, but the angle of polarisation of those samples must be known accurately. Chen [20] introduced a calibration approach that has several constraints in the experiment setup: the light source is expected to come from an integrating sphere, and a band-pass filter is added to enable the estimation of the missing pixel through a Fourier-based approach. Moreover, a motorized rotative polarization filter is used, and all the light parameters should be known beforehand. Ding *et al.* [27] propose a method to calibrate a micro-grid, monochrome polarization camera by using two polarization filters and a band-pass filter. With it, they show that the calibration algorithm does not require to have a constant intensity when calibrating, but the calibration setup is not minimal, and an additional step is required to align the two linear polarization filters. To proceed with the alignment, the authors use a power meter. Powell and Gruev [84] calibrate monochrome DoFP polarimeters, by two approaches: the single- and super-pixel algorithm. In their paper, they explain that a simple gain calibration is not enough, since it does not

have effect over the polarization parameters. Indeed, when a gain is added to the pixel intensities, this gain disappears when computing the AoLP and the DoLP, and there is no consideration about the change in the filter orientation, nor the non-ideality of the filter. This is not the case of the super-pixel method, in which a full pixel model is used. The drawback of their method is that the calibration set up is too elaborated to be replicated : the light information should be known beforehand, they use a band-pass filters and a heat filter, a shutter to control the amount of light, a motorized linear polarization filter, and an integrating sphere to create uniform unpolarized light.

In the rest of this chapter, we will present the details about our approach. Then, we include the results showing the effects of the calibration over the images captured with the Sony Polarsens sensor. We will finish with the conclusions about this part of the thesis work. In Tab. 3.1 we show a summary of the requirements of the proposed method, and the ones detailed above to compare the reduction in the resources needed to calibrate the camera.

3.2 Developed method

As part of this thesis work, we propose an algorithm to calibrate a color-polarization camera. The aim of the model is to compensate the effects of the manufacturing imperfections and the lens on the parameters on which the Stokes depends, i.e., the total intensity, the DoLP, and the AoLP. An overview of the proposed pipeline is sketched in Fig. 3.1. It is grounded on the super-pixel calibration method detailed in [35, 84], which is a well-established method in the literature. Different from other approaches, our calibration algorithm does not require the knowledge of the polarization state of the input light. Instead, we propose to estimate them and use the estimated polarization state in the calibration method. This section is split as follows: Firstly, in Sec. 3.2.1, we depict the super-pixel calibration method, in which the input light polarization parameters are required. Then, in Sec. 3.2.2 and Sec. 3.2.3, we present methods to estimate these light parameters.

3.2.1 Base super-pixel calibration

As mention in Sec. 2.1, a DoFP sensor measures the polarization state of the light by using super-pixels, i.e., sets of 2×2 pixels with linear polarization filters on top of them. These pixels are not perfect since they have a particular pixel gain, a filter that is not ideal, and their orientation maybe deviated from their theoretical value. Additionally, the lens will modify the polarization state of the light, and the transmission axis seen by the light. Indeed, the transmission axis of a filter is measured with respect to a light that arrives perpendicularly to it. In a perspective camera, the light rays arrive at different angles, depending on the pixel's position with respect to the central point [54]. This transmission axis seen by the light is called *effective transmission axis*. In summary, even if the sensor is ideal, the fact of using a lens will modify the effective parameter values of the pixel model.

To account for these non-idealities of the sensing unit, the pixel model of Eq. (2.5) should be used. This equation has been copied here for convenience in Eq. (3.1)

$$S_{0_\theta}^{out} = \frac{1}{2} \begin{bmatrix} q+r & (q-r)\cos(2\theta) & (q-r)\sin(2\theta) \end{bmatrix} \underline{\mathbf{S}}, \tag{3.1}$$

We remember that in this equation, q and r are the major and minor light transmittance of the linear polarizer, respectively, and θ is the orientation of the micro-filter of the pixel. Additionally, $S_{0_\theta}^{out}$ is the total intensity of the output Stokes vector, and $\underline{\mathbf{S}}$ is the Stokes vector of the incident light. Ignoring the dark current of the pixel, and letting $T = \frac{q-r}{2}$ be the pixel gain, and $P = \frac{(q-r)}{(q+r)}$ a coefficient that represents the non-ideality of the filter, we obtain Eq. (3.2):

$$I_\theta = \begin{bmatrix} \frac{T}{P} & T\cos(2\theta) & T\sin(2\theta) \end{bmatrix} \underline{\mathbf{S}}, \tag{3.2}$$

As we can see, to fully describe a pixel in a polarization camera, we need to know three parameters: (T, P, θ). Then, considering that we use a super-pixel of the sensor, we have four measurements for a single Stokes vector. Therefore, we obtain the following matrix

equation:

$$\begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \end{bmatrix} = \begin{bmatrix} {}^{T_1}/_{P_1} & T_1\cos(2\theta_1) & T_1\sin(2\theta_1) \\ {}^{T_2}/_{P_2} & T_2\cos(2\theta_2) & T_2\sin(2\theta_2) \\ {}^{T_3}/_{P_3} & T_3\cos(2\theta_3) & T_3\sin(2\theta_3) \\ {}^{T_4}/_{P_4} & T_4\cos(2\theta_4) & T_4\sin(2\theta_4) \end{bmatrix} \underline{\mathbf{S}}$$
$$\Rightarrow \underline{\mathbf{I}} = \mathbf{A}\underline{\mathbf{S}} \tag{3.3}$$

where θ_i with $i = \{1, 2, 3, 4\}$ are each of the super-pixel micro-filter effective orientation, $\underline{\mathbf{I}}$ is the measured intensity vector, and $\underline{\mathbf{S}}$ is the Stokes vector of the incident light.

Calibrating a polarimetric camera consists in determining the super-pixel matrix $\mathbf{A}$ by solving the Eq. (3.3) for each super-pixel individually. To be able to solve this equation, that has $4 \cdot 3 = 12$ unknowns in the matrix $\mathbf{A}$, at least three calibration light samples must be acquired by the camera. Considering the general case where N calibration samples are acquired with $N \geq 3$, the left hand side intensities vector of Eq. (3.3) becomes a $4 \times N$ matrix, and the Stokes vector becomes a matrix of size $3 \times N$. The matrix equation to solve is, thus, defined as:

$$\mathbf{I} = \mathbf{AS}, \tag{3.4}$$

where $\mathbf{I}$ is the intensity matrix of the N calibration light samples, $\mathbf{A}$ is the super-pixel matrix, and $\mathbf{S}$ is the Stokes vectors matrix of the N calibration light samples. Consequently, using a least-squares approach, the matrix $\mathbf{A}$ is equal to:

$$\mathbf{A} = \mathbf{IS}^{+}, \tag{3.5}$$

where $\mathbf{S}^{+}$ is the pseudo-inverse of $\mathbf{S}$. Eq. (3.5) constitutes the super-pixel calibration equation, and it can be solved for $\mathbf{A}$ if the polarization states of the N input calibration light samples are known. These states corresponds to the N columns of the matrix:

$$\mathbf{S} = \begin{bmatrix} S_{01} & \dots & S_{0N} \\ S_{01}\rho_1\cos(2\alpha_1) & \dots & S_{0N}\rho_N\cos(2\alpha_N) \\ S_{01}\rho_1\sin(2\alpha_1) & \dots & S_{0N}\rho_N\sin(2\alpha_N) \end{bmatrix}, \tag{3.6}$$

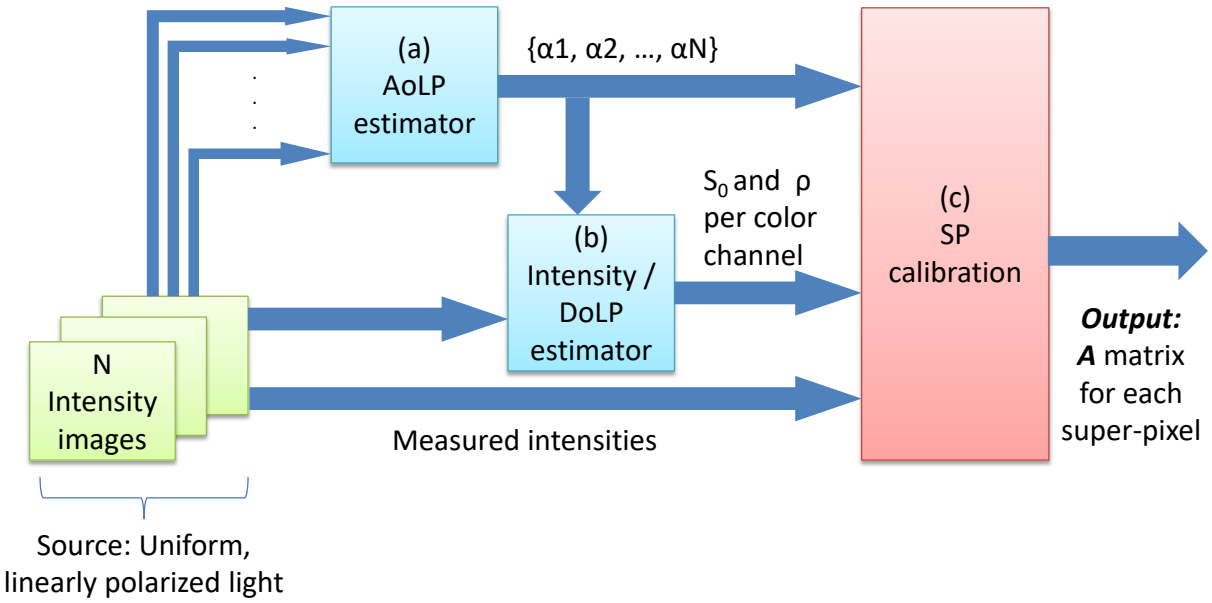


Figure 3.1: Proposed calibration pipeline. From a Uniform Linearly Polarized light, N samples are captured. The block (a) will estimate the angle of linear polarization of each sample, and with them and the captured images, the block (b) will estimate the degree of linear polarization and the intensity of the source. Finally, with these estimations and the intensity measurements, the super-pixel calibration is done (c).

where S_{0n} is the intensity of the n^{th} calibration light sample, ρ_n is its Degree of Linear Polarization (DoLP), and α_n is its Angle of Linear Polarization (AoLP), for $n = 1, ..., N$. These parameters can be obtained with high accuracy but at the expense of a complex laboratory set-up and time consuming experiments. In the following sections, we will show how to estimate them using a Uniform Linearly Polarized (ULP) light that has moderate values in each of the three RGB channels.

For creating this type of light, two configurations are possible, for which the proposed calibration algorithm is valid: i) a linearly polarized light emitting device is fixed, and the camera is rotated to obtain samples at different angles of linear polarization, or ii) a rotative linear polarization filter is placed between a fixed camera and a fixed unpolarized light source. These two cases are illustrated in Fig. 3.2. Due to equipment availability, the second configuration is used for the experiments.

3.2.2 Input light angle of polarization estimation

In this section, we describe how to estimate the AoLP of the calibration light samples represented by α_n with $n = 1, ..., N$ in Eq. (3.6). Considering that the polarization angle

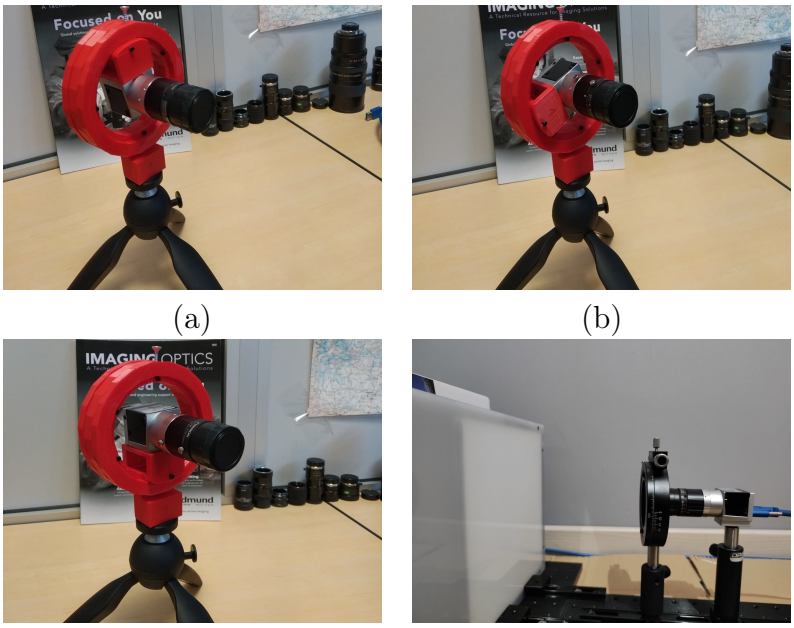


Figure 3.2: Possible configurations of the camera and the illumination system for which our calibration algorithm is valid. (a), (b) and (c) Linearly polarized light source is fixed, and the camera is rotated around its axis. (d) the camera is fixed, and a linear polarization filter placed between the camera and the unpolarized light source is rotated

measured by a single super-pixel is independent of the light wavelength [1], then all the super-pixels will observe the same polarization angle if they are illuminated by a ULP light. In the real-world, this is not the case due to parameters dispersion, and there is not a single pixel in the sensor that can provide an accurate measurement of this angle. Nonetheless, if the AoLP error has a mean of zero, then the mean of all the estimations should be close to the true value.

To compute this mean AoLP, we select a certain number of pixels that are negligibly affected by several undesired effects such as vignetting due to the lens and the aperture, and polarization state errors due to light rays of large angle of incidence. These pixels are those that are found in the central region and that receive light rays pertaining to a small solid angle or Angular Field Of View ($AFOV$). The relationship between the $AFOV$, the length h in mm of a square Region Of Interest (ROI) and the focal length f of the lens in mm is given by:

$$AFOV = 2 \arctan\left(\frac{h}{2f}\right). \tag{3.7}$$

[1] Strictly speaking, they are not equal, but for the visible wavelength, the difference between bands is negligible.

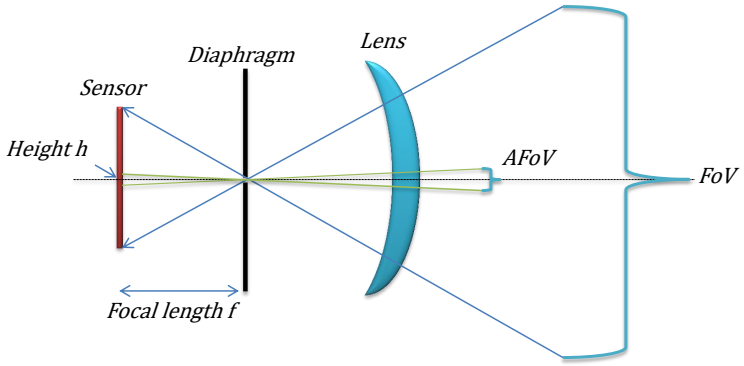


Figure 3.3: Angular field of view definition. The distance between the sensor and the projection center is the *focal length*. The solid angle covered by the camera is called *field of view*. The small solid angle around the optical axis of the camera is the *angular field of view*, and the region in the sensor that corresponds to the projection of this solid angle on it defines the height *h*.

This relationship can be deducted from the geometry shown in the Fig. 3.3. Let p be the size of a super-pixel in mm and $AFOV_{max}$ the maximum allowed angular field of view. Then, we can use the Eq. (3.7) to compute an upper limit for the amount of pixels that enters in the central region denoted by N_{sp}.

$$N_{sp} \leq \left\lfloor \frac{h}{p} \right\rfloor = \left\lfloor \frac{2f}{p} \tan\left(\frac{AFOV_{max}}{2} \right) \right\rfloor \tag{3.8}$$

In other words, any central region of $N_{sp} \times N_{sp}$ super-pixels that satisfies Eq. (3.8) constitutes an acceptable angular field of view for the estimator. If we set the angular limit to 1° or 2°, then we can say that such small region can be used for estimating the AoLP of the n^{th} light sample using contiguous 2×2 pixels. We should note that the pixels used must correspond to the same color filter, i.e., to estimate an AoLP, we cannot take one intensity coming from one color channel and another intensity from another color channel. Then, the obtained intensities can be used jointly with Eq. (3.3), in which:

- $\underline{\mathbf{I}} = \left[I_0^j, I_{45}^j, I_{90}^j, I_{135}^j \right]^T$ is the intensity vector of the j^{th} super-pixel,
- $\underline{\mathbf{S}} \simeq \underline{\hat{\mathbf{S}}_\mathbf{n}^\mathbf{j}} = \left[\hat{S_{n0}^j}, \hat{S_{n1}^j}, \hat{S_{n2}^j} \right]^T$ is the Stokes vector of the incoming light sample measured by the j^{th} super-pixel,
- $\left(T_i^j, P_i^j, \theta_i^j \right)$ are the parameters of a pixel that belongs to the j^{th} super-pixel, and to the i^{th} pixel in the super-pixel arrangement.

For the central pixels, we assume that the lens has no influence in the measurements,

and the pixels are not far away from the ideal conditions. Then, the ideal values of $T_i^j = 0.5$, $P_i^j = 1$, and $\theta_i = \{0°, 45°, 90°, 135°\}$ for all $i = \{1, 2, 3, 4\}$ can be adopted as a good approximation. Consequently, the matrix $\mathbf{A}$ is completely known, and with its pseudo-inverse, the Stokes vector of the n^{th} light sample, $\underline{\hat{\mathbf{S}}_\mathbf{n}^\mathbf{j}}$ can be obtained by:

$$\underline{\hat{\mathbf{S}}_\mathbf{n}^\mathbf{j}} = \begin{bmatrix} \hat{S_{n0}^j} & \hat{S_{n1}^j} & \hat{S_{n2}^j} \end{bmatrix}^T = \mathbf{A}^+\mathbf{I}. \tag{3.9}$$

Finally, the AoLP α_j measured by the j^{th} super-pixel is then given by:

$$\alpha_j = \frac{1}{2}\arctan\left(\frac{\hat{S_{n2}^j}}{\hat{S_{n1}^j}}\right). \tag{3.10}$$

This operation is repeated for all the $K = N_{sp} \cdot N_{sp}$ super-pixels in the central region, and their average will give a good approximation of the real Angle of Linear Polarization. Nonetheless, it is worth noting that, due to the periodicity of the AoLP, the normal average of the K angles might conduct to wrong results. Thereby, we have to consider the circular average expressed by Eq. (3.11), and whose result is an estimation of the AoLP of the n^{th} light sample denoted by α_n with $n = 1, ..., N$.

$$\begin{aligned} \sin\left(2\hat{\alpha_n}\right) &= \tfrac{1}{K}\textstyle\sum_{j=1}^{K}\sin\left(2\alpha_j\right) \\ \cos\left(2\hat{\alpha_n}\right) &= \tfrac{1}{K}\textstyle\sum_{j=1}^{K}\cos\left(2\alpha_j\right) \\ \hat{\alpha_n} &= \tfrac{1}{2}\arctan\left(\sin\left(2\hat{\alpha_n}\right)/\cos\left(2\hat{\alpha_n}\right)\right) \end{aligned} \tag{3.11}$$

The algorithm to estimate the AoLP as explained in this section is detailed in Alg. 1.

Algorithm 1 Estimator for the angle of polarization of an image

1: **Input**: Image of a ULP light taken by the camera, and default angles of the micro-polarizers θ_i
2: Compute ideal pixel matrix $\mathbf{A}$ and its pseudo-inverse $\mathbf{A}^+$.
3: **for** each group of pixels **do:**
4: Build intensity vector $\mathbf{I}$
5: Compute matricial product $\mathbf{A}^+\mathbf{I}$
6: Estimate α_j as in Eq. (3.10)
7: Add the value to the circular avg. estimation
8: **end for**
9: Compute $\bar{\alpha}$ by using Eq. (3.11).
10: **Output**: Average value $\bar{\alpha}$

3.2.3 Light samples intensity and DoLP estimation

In what follows, we present a method to estimate the other two light parameters required to construct the Stokes vector, i.e., the light intensities and the DoLP of the N calibration light samples.

For the same reasons as explained in Sec. 3.2.2, a $N_{sp} \times N_{sp}$ super-pixels region around the center of the sensor is considered. Since the light source and the filter placed between the camera and the light source do not change, all the light samples will share the same intensity and DoLP values, and a different AoLP between samples. Thus, in Eq. (3.6), $S_{01} = ... = S_{0N} = S_0$ and $\rho_1 = ... = \rho_N = \rho$. Consequently, the Stokes matrix of the N calibration light samples $\mathbf{S}$ can be split into two matrices: a 3×3 matrix $\mathbf{L}$ that only depends on (S_0, ρ), and a $3 \times N$ matrix $\mathbf{G}$ that only depends on the angles of linear polarization α_n estimated in Sec. 3.2.2, such that $\mathbf{S} = \mathbf{LG}$:

$$\mathbf{S} = \begin{bmatrix} S_0 & 0 & 0 \\ 0 & S_0\rho & 0 \\ 0 & 0 & S_0\rho \end{bmatrix} \begin{bmatrix} 1 & ... & 1 \\ \cos(2\alpha_1) & ... & \cos(2\alpha_N) \\ \sin(2\alpha_1) & ... & \sin(2\alpha_N) \end{bmatrix} \tag{3.12}$$

Combining Eq. (3.4), which is the super-pixel calibration equation, and Eq. (3.12), yields:

$$\mathbf{IG}^{+} = \mathbf{AL}, \tag{3.13}$$

where $\mathbf{I}$ is the $4 \times N$ matrix of the measured intensities. For the j^{th} super-pixel, each row i of the result $\mathbf{IG}^{+}$ can be expressed as:

$$\left(\mathbf{IG}^{+}\right)_i^j = \begin{bmatrix} X_i^j & Y_i^j & Z_i^j \end{bmatrix}, \tag{3.14}$$

where $X_i^j = \frac{T_i^j S_{0_i}^j}{P_i^j}$, $Y_i^j = T_i^j S_{0_i}^j \rho_i^j \cos\left(2\theta_i^j\right)$, and $Z_i^j = T_i^j S_{0_i}^j \rho_i^j \sin\left(2\theta_i^j\right)$. If the camera is considered ideal for the central pixels, as in the previous section, each of the four rows of

$\mathbf{IG}^+$ allows to calculate a pair $\left(S_{0_i}^j, \rho_i^j\right)$ as follows:

$$S_{0_i}^j = 2X_i^j \quad \rho_i^j = \frac{\sqrt{Y_i^{j2} + Z_i^{j2}}}{X_i^j}. \tag{3.15}$$

Repeating this procedure for the K super-pixels and the N samples will yield two sets of $R = N_{sp} \cdot N_{sp} \cdot 4$ intensities and DoLP: $\left\{S_0^1, S_0^2, ..., S_0^R\right\}$ and $\left\{\rho^1, \rho^2, ..., \rho^R\right\}$. From these two sets, the light parameters, S_0 and ρ, can be estimated by extracting either the maximum (highly sensitive to noise and outliers), the average (affected by lens vignetting and outliers) or the median (affected only by lens vignetting) value. Because of its robustness to outliers, the median value has been chosen and implemented for the experiments.

It is important to note that a color camera is used. To be free from the requirement of using a white light, the detected intensities and DoLP are classified per color channel, without mixing them. The color channel to which a super-pixel belongs to is given by its position j. Indeed, considering that a colored super-pixel has a size of 4×4 elements, a pixel with location (U, V) belongs to a super-pixel location (u, v) defined by:

$$\begin{cases} u = U \ \% \ 4 \\ v = V \ \% \ 4, \end{cases} \tag{3.16}$$

where the operation $(A \ \% \ B)$ is the remainder of the integer division of the two values A and B. With these coordinates, and the colored super-pixel pattern shown in Fig. 2.5, the color channel can be retrieved.

At this point, an estimation of the light intensity $\hat{S}_0$, the degree of linear polarization $\hat{\rho}$ per color channel, and the angle of polarization $\hat{\alpha_n}$ at the n^{th} position of the linear filter has been obtained. Therefore, the Stokes matrix $\hat{\mathbf{S}}$ can be built, as in Eq. (3.17),

$$\hat{\mathbf{S}} = \begin{bmatrix} \hat{S}_0 & 0 & 0 \\ 0 & \hat{S}_0\hat{\rho} & 0 \\ 0 & 0 & \hat{S}_0\hat{\rho} \end{bmatrix} \begin{bmatrix} 1 & ... & 1 \\ \cos\left(\hat{\alpha_1}\right) & ... & \cos\left(\hat{\alpha_N}\right) \\ \sin\left(\hat{\alpha_1}\right) & ... & \sin\left(\hat{\alpha_N}\right) \end{bmatrix}. \tag{3.17}$$

Then, its pseudo-inverse $\hat{\mathbf{S}}^+$ can be computed and used in Eq. (3.5) to calculate the j^{th}

Algorithm 2 Light parameters estimator

1: **Input**:
- N light samples I_m
- Angles of polarization of the samples α_m
2: Compute matrix $\mathbf{G}$ from Eq. (3.12) and its pseudo-inverse $\mathbf{G}^+$
3: **for** each group of pixels **do**:
4: Detect color of super-pixel j
5: $\mathbf{I}$ = Matrix of samples for the group of pixels j (Eq. (3.4))
6: $\mathbf{AL} = \mathbf{IG}^+$
7: Store the coefficients X_i^j, Y_i^j, and Z_i^j from Eq. (3.14)
8: Compute $S_{0_i}^j$ and ρ_i^j by Eq. (3.15)
9: Add $S_{0_i}^j$ and ρ_i^j to average (or max. or med.)
10: **end for**
11: Compute the Stokes matrix $\hat{\mathbf{S}}$ as in Eq. (3.17)
12: **for** each pixel **do**:
13: Compute pixel parameters (T_i, P_i, θ_i) from Eq. (3.3)
14: **end for**
15: **Output**:
- $\hat{S}_0$, $\hat{\rho}$ per color channel and (T_i, P_i, θ_i) for each pixel

super-pixel matrix that we will denote here by $\hat{\mathbf{A}}_\mathbf{j}$:

$$\hat{\mathbf{A}}_\mathbf{j} = \mathbf{I}\hat{\mathbf{S}}^+. \tag{3.18}$$

It follows that, from $\hat{\mathbf{A}}_\mathbf{j}$, each row allows to compute the parameters (T_i, P_i, θ_i) for each of the four pixels that compose the j^{th} super-pixel.

The detailed algorithm for this estimator is detailed in Alg. 2.

3.3 Experiments

Our experimental setup is composed of a Basler acA2440-75ucPOL camera with a Sony Polarsens IMX250MYR sensor of pixel size equal to $3.45\mu m$ or super-pixel size equal to $6.9\mu m$, and a Fuji-film HF16XA-5M - F1.6/16mm lens. To compute an initial estimation of the AoLP and DoLP required by our calibration method, we have chosen a central region of 50×50 super-pixels determined according to Eq. (3.8). This region corresponds to incident light rays with a maximum angle of incidence of $0.625°$ that is relatively small and will give a good initial estimation of the polarization parameters of the incident light.

The developed algorithm runs on a computer with Intel Core i7-10850H @ 2.7 GHz and 32 GB of RAM. The OS is Ubuntu 18.04 LTS 64 bits. The program runs in 7 seconds for 7 samples, and 8 seconds for 73 samples approximately. The experimental set-up model is

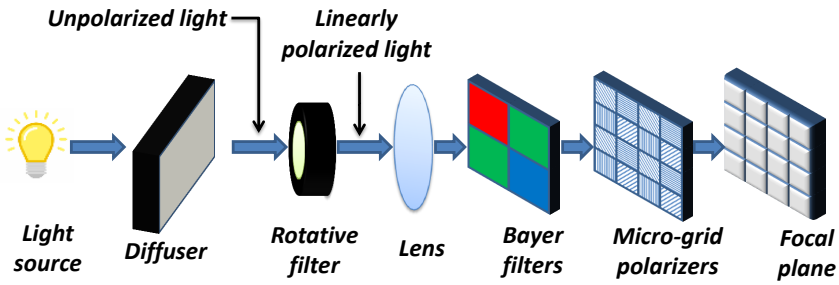


Figure 3.4: The experiment set up where only one super-pixel is represented[91].

shown in Fig. 3.4. The uniform, unpolarized light source device is a Schott Fostec DCR III fiber optic illuminator, with a Schott ColdVision back light A08927. A 50mm linear glass polarizing filter is used (Edmund Optics Inc #56-329), mounted on a metric polarizer mount (Edmund Optics Inc #43-787). The linear polarizer filter is rotated by hand. Each position of the filter corresponds to a light sample, and for each sample ten acquisitions of the light are done and averaged to reduce the effects of the noise in the parameters estimation. The acquisitions are done in a dark room to reduce the influence of the environment. Furthermore, the recommendations given in [57] have been followed. Particularly, the lens has been correctly focused at the light source plane, and the f-number has been set higher than 2.8 for all the experiments. Finally, we have acquired several images with the camera in total darkness and verified that, with a 12-bit pixel count and an exposure time of 200ms, the dark current can effectively be neglected as reported in [57].

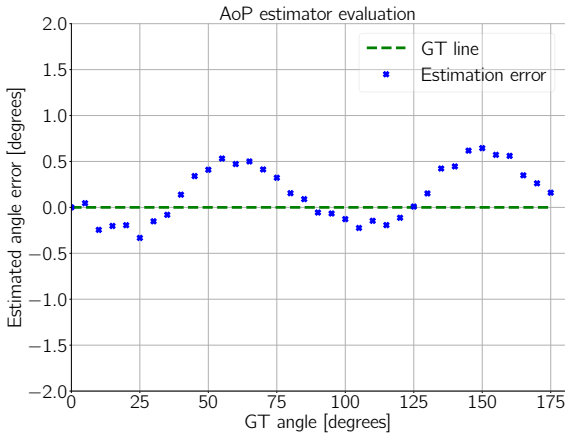


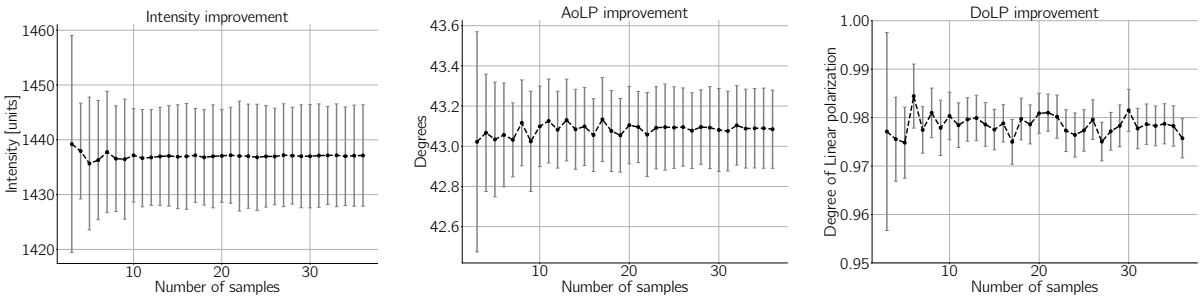
Figure 3.5: AoLP estimator evaluation. The maximum error is of 0.65°, and the RMSE is of 0.3316° for all the range from $[0°, 180°]$. The sine-like evolution is mainly due to the error in the polarizers orientations. See Appendix A for the demonstration.

3.3.1 Evaluating the AoLP estimator

The first experiment that we have done is to verify the quality of the estimated AoLP with the camera. The different AoLP values are set by turning the polarization filter in front of the light source by steps of 5°, in the range $[0°, 175°]$. Thirty-six images of the source light are acquired with the camera, and from them, the AoLP are calculated and compared to the true reference values given by the position of the polarization filter. For better visualization, only the deviations from the reference values are represented in Fig. 3.5. A reference horizontal line is shown in this figure to indicate the positions at which the estimator produces an error of zero degrees. As we can see, the AoLP error curve exhibits a sine-like shape that is due to errors in the parameters of the camera. Indeed, it can be proven that a small error in the parameters of the camera due to imperfections will induce, in first order approximations, four error terms in the expression of the estimated AoLP. These error terms are functions of the sine and cosine of the true AoLP and they appear in the expressions of the computed Stokes components S_1 and S_2. Because of these additional components, and that the ratio of these two Stokes components is proportional to the tangent of the AoLP (as explained in Sec. 2.1), the error curve follows a sine and cosine rule. The detailed demonstration of this effect can be found in Appendix A. Nonetheless, by considering pixels around the center, and averaging several samples, the estimation error is reduced, such that the RMSE is 0.3316°, and the maximum error is ±0.65° in all the range. Hence, the experiment confirms that the camera can be used to provide reliable measurements of the AoLP of the ULP light. Additionally, it avoids the requirement of aligning the rotative filter and the camera, since the measurements are already in the camera's coordinate frame.

3.3.2 Evaluation of sensor and measurement quality

The next step is to evaluate the accuracy of the calculated intensities, AoLP and DoLP with the uncalibrated and the calibrated camera with N number of calibration light samples. Prior to the test, a database has been created with all the required calibration light samples images. These samples have the same intensity and DoLP, but different AoLP in the range of $[0°, 180°]$. For the test, N sample images are randomly selected from the database and used



(a) (b) (c)

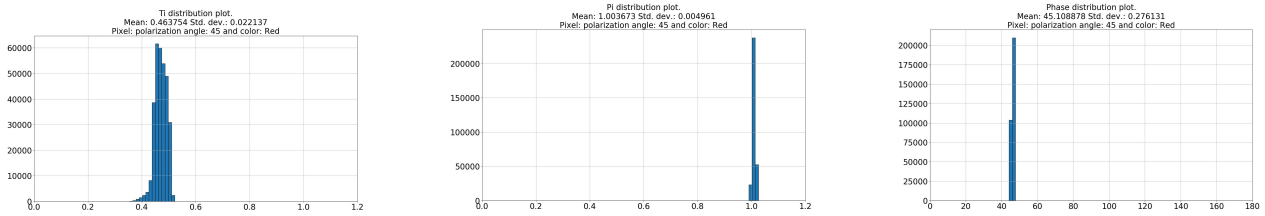
Figure 3.6: Comparative graphs of how the accuracy changes with the number of light samples. Each sample is taken at a different position of the rotative filter. Each point in the curve is the average of several runs of the algorithm. The black curves are the mean values, and the gray vertical bars are the standard deviation. When uncalibrated, the parameters are: $S_0 = 1336.327 \pm 61.731$, $\rho = 0,9776 \pm 0,006$, and $\alpha = 43,024° \pm 0.48°$. When calibrated and the amount of samples $N \geq 10$, $S_0 = 1437.134 \pm 9.25$, $\rho = 0.98 \pm 0.005$ and $\alpha = 43.099° \pm 0.2°$. (a) Intensity S_0. (b) AoLP α. (c) DoLP ρ.

to compute the pixel parameters. Once the camera is calibrated, the polarization parameters of a test image are estimated. The mean and standard deviation of the test light parameters are calculated over all the sensor area. This test is repeated several times for each value of $N = 3, ..., 36$. For each run of the algorithm, a new set of N random calibration images is chosen to calibrate the camera. Due to similarity of the results for different channels, only the results for the red channel are shown in Fig. 3.6. The GT values of the test image are: $S_0 = 1437$, $\rho = 0.97$, and $\alpha = 43°$.

As shown in Fig. 3.6, when five or more calibration light samples are used, the standard deviation is considerably reduced with respect to the case when only three samples are used, and when $N \geq 10$, the values are stabilized. More precisely, for $N \geq 10$, $S_0 = 1437.134 \pm 9.25$, $\rho = 0.98 \pm 0.005$ and $\alpha = 43.099° \pm 0.2°$. The same test image has been used with the uncalibrated camera , and the obtained parameters were: $S_0 = 1336.327 \pm 61.731$, $\rho = 0.9776 \pm 0.006$, and $\alpha = 43.024° \pm 0.48°$. This experiment corroborates that the camera calibrated with our algorithm reduces the disparity between values over the sensor area with respect to the uncalibrated camera.

It is possible to inspect the sensor quality by plotting the histograms of the pixel model parameters. The results for the polarization channel of 45° and the red color channel are shown in Fig. 3.7. Similar plots are obtained for the other channels.

As we can see, the P and θ parameters have a very low standard deviation, and their mean value is very close to the default value. Nonetheless, the T parameter has a wider distribution than the other two. This is normal since it is this parameter that accounts for



(a) (b) (c)

Figure 3.7: Histograms of the pixel parameters obtained after calibration for the polarization channel of 45°, and the red color channel. (a) Pixel parameter T. (b) Pixel parameter P. (c) Pixel parameter θ.

the vignetting effect compensation, i.e., it is in charge of compensating the variations in the intensity due to the lens. The other two parameters are responsible for the flat-field correction in the polarization parameters only. Since the sensor used is already of good quality and the used lens does not introduce too much distortion in the polarization parameters, it is expected to have a low correction in the P and θ parameters.

To confirm the validity of our method with respect to other algorithms, the calibration outcomes have been compared with the super-pixel (SP) method described in [35, 84], and the results are included in Tab. 3.2. The AoLP used for the SP method has been measured from the rotative filter, while for our method they have been estimated with the proposed algorithm in Sec. 3.2.2. The difference between the mean values of the intensity and the DoLP is expected, since each method uses a different reference during calibration. However, the most important results are the standard deviations that reflect how similar the measurements of the ULP light are after the correction over the entire sensor area. One can notice that the results obtained by both approaches have similar accuracy. However, our method has the advantage of being experimentally simple: it does not require any specific devices to measure the light polarization state. Additionally, the time required to take the samples is reduced for the user since it only needs to randomly turn the polarizer a few times. Also, the measurements of the orientation from the rotative filter are not required since the algorithm will estimate them.

3.3.3 Polarization state before and after calibration

Finally, the effect of calibration on an image of a ULP light is presented in Fig. 3.8. Again, since the results over all the color channels are very similar, only the images for the red chan-

	S_0	$AoLP$	$DoLP$
Uncalibrated	3399.2 [65.475]	59.983 [0.215]	0.9863 [0.0041]
SP method	3402.4 [10.723]	60.035 [0.128]	1.005 [0.004]
Our method	3298.0 [10.402]	59.701 [0.128]	0.985 [0.004]

Table 3.2: Comparison of our method with the super-pixel (SP) method [35]. Content of each cell: mean value [standard deviation]

nel are shown. In this figure, the images (a) and (b) correspond to the total intensity of the light, (c) and (d) are for the AoLP of the light, and (e) and (f) are the corresponding images of the DoLP. The top row of Fig. 3.8 shows the uncalibrated images, and the bottom row, the calibrated ones. These images are the measurements of the camera when it is illuminated by a ULP light. For the uncalibrated case, the values of the total intensity, AoLP and the DoLP are in the intervals $[1017, 1463]$, $[0.9465, 1.0]$, and $[41.099°, 44.879°]$, respectively. For the calibrated camera the corresponding values are in the intervals $[1370, 1521]$, $[0.9741, 1.0]$, and $[41.934°, 43.824°]$. What is important to note in these images is not the color but the variations in the colors. Since these images represent the values of the measured polarization parameters by the individual pixels of the camera, they should respectively be the same, i.e. per image, all the pixels should have the same color, due to the fact that the observed light is uniform and linearly polarized. However, this is not the case. The variations in the color represent the measurement errors by the individual pixels. One can note that the color is more uniform (and therefore, there are less errors) for the calibrated setup than for the uncalibrated one.

Additionally, these images show that, for the uncalibrated images, both polarization parameters present changes in the borders with respect to the center of the image. This is a consequence of the angle of incidence. Indeed, the filter characteristics are given for a light ray that arrives perpendicularly to the filter surface. The further we are from the center of the image, the larger the angle of incidence, thus the larger the difference between the default and the effective filter values. As one can see in Fig. 3.8, the uncalibrated measurements have strong variations in the four corners of the images, and in the center they are mostly constant. The proposed algorithm accounts for this alteration of the polarization state in the pixel model, by modulating the polarizers parameters P_i and θ_i, to fit the measurements with a flat-field response. T_i has no effect over these parameters since it is a factor of all

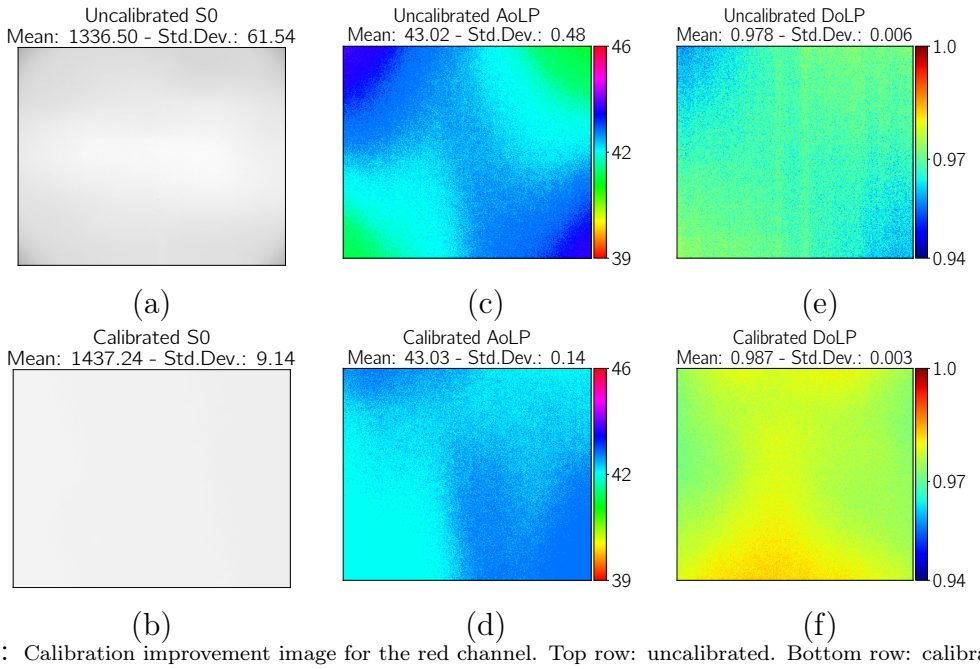


Figure 3.8: Calibration improvement image for the red channel. Top row: uncalibrated. Bottom row: calibrated. (a) and (b) Intensity images. (c) and (d) AoLP images. (e) and (f) DoLP images. AoLP uses the HSV color palette, and the DoLP uses the Jet palette. Since the light source is ULP, the ideal response would be images with a single value for all the pixels for the AoLP and another value for the DoLP. Due to characteristics dispersion in the pixels and the presence of the perspective lens, the uncalibrated images present different values between the center of the sensor and its borders in the two polarization parameters. After calibration, the distribution of measured values is reduced. Even though there seems to be a difference in the colors of the calibrated images, looking at the color bars range, it is clear that the range of values present in the image is small compared to the uncalibrated cases. The mean and the standard deviation of values in the images are detailed in the images titles.

the three Stokes parameters, thus it is cancelled when computing the DoLP and the AoLP. However, T_i corrects the vignetting effect over the intensity image.

3.3.4 Ablation study

To test the influence of each module, several scenarios have been evaluated and summarized in Tab. 3.3. The first row of this table, SC1, corresponds to the results when using the uncalibrated camera. Then, in SC2, the AoLP estimator has been used and the DoLP and intensity of the input light have been fixed to 0.8 and 2000, respectively. As shown in the table, this modification changes the corresponding mean value measured in the entire image, but their standard deviation (SD) is not modified. The advantage of having a module that estimates these parameters is that an early saturation of the measured value is avoided. Then, in SC3, the AoLP estimator is disabled, and these values have been measured from the filter ruler. In this case, the AoLP after calibration presents a slightly smaller SD than when using the estimator, due to the small error in the measurements of this parameter.

	S_0 $[0, 4095]$	$AoLP$ $(°)$	$DoLP$ $[0, 1]$
SC1	2750.66 [153.46]	60.08 [0.416]	0.986 [0.0033]
SC2	1985.63 [4.88]	59.87 [0.126]	0.797 [0.00362]
SC3	2967.34 [7.24]	59.61 [0.125]	0.987 [0.0045]
SC4	2967.38 [7.24]	69.61 [0.125]	0.978 [0.0044]
SC5	**2969.6 [7.26]**	**59.87 [0.126]**	**0.973 [0.0044]**

Table 3.3: Results of the ablation experiments.

Nonetheless, this has as trade off a large experiment time, and the requirement of a rotative mount with a ruler. In SC4, a similar experiment to SC3 is done, but a fixed shift is introduced in the AoLP measurements. This effect corresponds to a rotational difference between the coordinate systems of the camera and the filter. From Tab. 3.3 it can be seen that this affects the mean value of the measured AoLP, but the SD remains low. This is normal, since the AoLP is relative to the measurement system. Finally, our entire pipeline is tested in SC5, in which a small SD is obtained in all the variables, and additionally, the mean values are close to the GT values. Therefore, using a simple calibration set-up as ours can provide not only accuracy, but also precision in the measurements given by the camera.

3.4 Discussions

The results obtained so far show that the uncalibrated Sony Polarsens sensor is of high quality. This manifests as a very low standard deviation in the model parameters plot. The differences in the mean value of the filter orientations may be due to a misalignment between the filter used to create linearly polarized light and the sensor reference axis. As mentioned in [44], the low standard deviation in the pixel parameters is expected since the manufacturer ensures a spatial uniformity of approximately 0.5%, and an extinction ratio higher than 300. This was not the case of the first DoFP sensors [84], in which the micro-grid of filters was placed on top of the micro-lenses (which contributes to increase the cross-talk effect), the spatial uniformity was 8%, and the extinction ratio was about 50.

Additionally, the lens used for these experiments does not produce large deformations in the image since the largest Field of View (FoV) is less than 60°. Unfortunately, we could not compare the results obtained with other cameras due to their unavailability in the laboratory. Nonetheless, the proposed method is still valid since if used, we can ensure the measurements

are correct, no matter if the camera is of the newer or older technology, or if the lens has a small or large FoV. Hence, based on the camera configuration and the results provided by the calibration algorithm, one can consider that the system without calibration is accurate enough and thus decide, depending on the application, not to apply any correction.

3.5 Conclusions

In this chapter, we have introduced a polarization camera calibration algorithm developed to achieve a flat field response over the entire sensor area. The results included in the experiment section show that following the proposed procedure improves the quality of the measurements in both, accuracy and precision. Contrary to other methods, we have managed to reduce the experiment setup to its bare minimum, and we have simplified the camera calibration step for the user who does not need to know the polarization state of the calibration light beforehand. The calibrated measurements are not too different from the uncalibrated ones for the setup used since the Sony Polarsens sensor is already of high quality, and the tested lenses do not produce large deformations in the intensity image. Nonetheless, the method is still valid, and it can be used with several combinations of DoFP sensor, and lenses. Furthermore, as mentioned in the Discussions, if previously released DoFP technology is used, the calibration procedure is mandatory to be able to have more reliable polarization measurements. In that case, our calibration method will allow the user to achieve this with a very simple setup.

Chapter 4

Shape from polarization application

The direct application of the polarization information is the normal estimation at a surface point. When this application is done for the whole surface of an object it is called Shape from Polarization (SfP). Nonetheless, several conditions regarding the reflection type and object material need to be met to correctly reconstruct the normal vector. In this chapter we provide a complete description of the physical problem, an explanation of the procedure for computing the ground-truth normal vector, and a qualitative and a quantitative evaluation of the obtained results. We test the algorithm for the different reflection models individually, and we conclude that using the diffuse- or specular-dominant hypothesis introduce a non-negligible error in the final results. Additionally, we test the SfP algorithm with our calibration procedure and we confirm that even small measurement corrections result in large quantitative improvements in the Mean Angular Error of the reconstruction.

4.1 Properties of the naturally generated polarization

An important property of the polarization state is that it conveys information about the shape and the composition of objects. According to the electromagnetic wave theory, a wave that hits a surface will create two new waves: a reflected, and a refracted wave. In general, the light captured by a camera that is observing an object is the result of reflection. A close zoom to the surface of a generic object is shown in Fig. 4.1. This sketch shows the incident, the reflected and the refracted light at the surface level, and the interaction of the light with

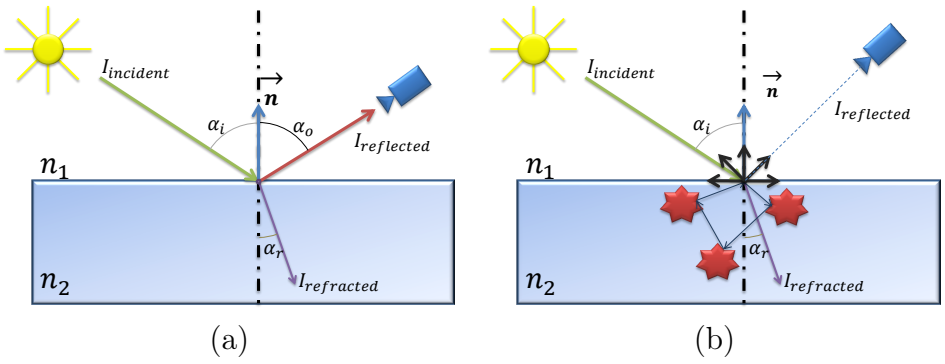


Figure 4.1: Sketch of the interaction between the light and an object. The top medium is the air, and it has an index of refraction n_1. The bottom medium has an index of refraction n_2. The light arrives at the surface with an angle α_i. The portion of the light that enters the object is the refracted light, and it travels in the direction α_r (a) Specular reflection: the incident light hits the surface at the point, creating a single ray of reflected light. The light leaves the surface with an angle α_o, which is the same as the incident ray angle. (b) Diffuse reflection: the incident light enters into the surface, but it hits the sub-surface particles several times before exiting the material. When the internally reflected light hits the material interface, it exits by spreading the light in all the directions with the same intensity, and a low degree of polarization. The intensity of the exiting light depends on the angle of incidence.

the different mediums (air and the object).

A particular natural case appears when the sum of the angles of the specular reflected and the refracted light with respect to the normal vector $\mathbf{n}$ is equal to 90°. In this case, the reflected light will be 100% linearly polarized, and the polarization direction will be parallel to the surface at which it is reflected [40]. This type of reflection occurs only at a single angle, called the *Brewster angle*, and it can be found using Eq. (4.1), also known as the Snell's law:

$$\frac{n_2}{n_1} = \frac{\sin(\alpha_o)}{\sin(\alpha_r)} \tag{4.1}$$

where n_1 is the index of refraction of the top medium in Fig. 4.1, and n_2 is the corresponding index for the bottom medium. Furthermore, α_o is the angle of the reflected light w.r.t. the normal vector $\mathbf{n}$, and α_r is the angle formed by the refracted light with respect to the opposite to the normal vector. Since for this case, to have 100% linearly polarized light $\alpha_o + \alpha_r = 90°$, then $\sin(\alpha_r) = \cos(\alpha_o)$. Under this condition, the angle α_o corresponds to the Brewster angle α_{Brew}. Therefore,

$$\begin{gathered} \tan(\alpha_{Brew}) = \frac{n_2}{n_1}, \\ \text{and then } \alpha_{Brew} = \arctan\left(\frac{n_2}{n_1}\right). \end{gathered} \tag{4.2}$$

For each frequency, and combination of materials, the ratio $n = {}^{n_2}/_{n_1}$ is fixed, thus there is a single incidence angle at which it is possible to obtain totally linearly polarized light. Any other angle α_o will produce partially linearly polarized light.

Another important detail regarding the reflected light is that the polarization state is related to the surface orientation, therefore to the normal vector to the surface at the interface point. The exact relationship between the normal to the surface and the polarization state depends on the type of reflection and the type of material. In what follows, we will describe the general method to estimate the normal vectors from the polarization parameters, and the ground-truth normal. There is no modification to this general method when we apply the calibration algorithm since this additional step is done before starting the algorithm. Once the image is corrected, it can be treated as an image coming directly from the camera. In the following section, the entire pipeline to estimate the normal vector from the polarization state will be described, as well as the effects of using a calibrated setup with respect to an uncalibrated one.

4.2 Normal vectors from polarization theory

In what follows, we will detail the results obtained with a Shape from Polarization (SfP) method, implemented from scratch, by using only the physical constraints. Our contribution consists of a complete formulation of the normal estimation theory based on the polarization theory, which has not been done previously in the literature. We include the original Fresnel theory equations that relate the DoLP to the normal zenith angle and their inverted equations, and the relationship between the AoLP and the normal azimuth angle. Additionally, we give an in-depth explanation of how the light interacts with the different type of materials, and we quantify the errors produced by the diffuse-dominant and specular-dominant assumptions. It is important to clarify that we did not develop a new method. Our objective is to show the effect produced by the sensor calibration over a concrete application. To avoid misunderstanding in the obtained results, we have considered a single type of object: a plane surface made of a single material. In this way, the ground-truth information can be geometrically estimated, and then used to quantify the normal vector error. We will describe

a pipeline that we have implemented to estimate the normal field of an object, by using the polarization theory. To be able to produce simple but valuable comparisons between a calibrated and an uncalibrated setup, we will use a simple arrangement of a light source, a polarization camera, and a plane surface. This surface is made of a single dielectric material, thus all the observed pixels has a single index of refraction. Additionally, since a plane surface is used, the normal vectors should be the same for every super-pixel. Therefore, the ground-truth consists of a single vector the describes the plane.

4.2.1 Mathematical formulation of the SfP problem

The mathematical description of the relationships between the normal vectors and the polarization parameters can be obtained from Fresnel formulae [74]. Particularly for the zenith, an additional parameter requires to be known to estimate this angle. This parameter is the index of refraction which has a unique value for each material and frequency. It is a real number if the material is dielectric and a complex number if the material is metallic [74].

Another point to consider when estimating the object's normal field is the reflection type, which can be of two types: diffuse or specular. In the former case, part of the light penetrates the object, and it is reflected several times in the sub-surface layer particles, before exiting the object. At each internal reflection, the light is depolarized, and when it passes through the surface layer to the air, it is spread in all the directions with the same intensity, and with a small DoP, as shown in Fig. 4.1 (b). In the specular reflection case, a portion of the light does not penetrate the object, and it is reflected at the surface layer directly, as shown in Fig. 4.1 (a). In this case, a single ray of light is reflected for a single ray of light that hits the surface. The sketch of how the light interacts with the materials is included in Fig. 4.1.

With these two supplementary information (index of refraction and reflection type), a unit length normal vector to the surface can be estimated. This vector is defined by the zenith and the azimuth angles, which can be obtained from the polarization state as follows [97]:

Diffuse reflection: The azimuth angle α is related to the AoP ϕ. There are two possibilities: either $\alpha = \phi$, or $\alpha = \phi + \pi$. This fact is known as the π-ambiguity of the AoP, since it

is not possible to distinguish the azimuth angle from the orientation of the oscillating wave. Regarding the zenith angle θ, it is related to the DoP by the following formula [97]:

$$\rho_d = \frac{\sin^2(\theta)\left(\eta - \frac{1}{\eta}\right)^2}{4\cos(\theta)\sqrt{\eta^2 - \sin^2(\theta)} - \sin^2(\theta)\left(\eta + \frac{1}{\eta}\right)^2 + 2\eta^2 + 2} \tag{4.3}$$

where η is the index of refraction of the material, and θ is the zenith angle of the normal vector to the surface. This equation has a closed-form solution for the zenith given by [97]:

$$\cos(\theta) = f(\rho, \eta) = \sqrt{\frac{2\rho + 2\eta^2\rho + \rho^2 + 4\eta^2\rho^2 - \eta^4\rho^2 - 4\eta^3\rho\sqrt{(1-\rho^2)} + (\eta^2-1)^2}{(1+\eta^4)(1+\rho)^2 + 2\eta^2(3\rho^2 + 2\rho - 1)}} \tag{4.4}$$

Specular reflection: As for the diffuse reflection, the azimuth angle α is related to the AoP ϕ with a π-ambiguity. In this case, $\alpha = \phi \pm \frac{\pi}{2}$. Furthermore, the zenith angle θ of the normal vector is related to the DoP by the following equation [74]:

$$\rho_s = \frac{2\sin^2(\theta)\cos(\theta)\sqrt{\eta^2 - \sin^2(\theta)}}{\eta^2 - (1+\eta^2)\sin^2(\theta) + 2\sin^4(\theta)} \tag{4.5}$$

This function is not invertible, since for each DoP value, there are two possible zenith angles, except for the Brewster angle at which the function has a maximum. Nonetheless, if we split the function around the maximum value, we can obtain the two solutions as:

$$\sin(\theta) = \sqrt{\frac{\eta}{\sqrt{\frac{2\beta\sqrt{1-\rho^2}}{1-\beta\sqrt{1-\rho^2}} + \left(\frac{1+\eta^2}{2\eta}\right)^2} + \frac{1+\eta^2}{2\eta}}} \tag{4.6}$$

where again, η is the index of refraction of the material, and θ is the zenith angle of the normal vector to the surface. Additionally, the factor β is either 1 or -1, depending which solution we want to get (the left or right side to the maximum DoP value).

To visualize the dependency of the DoP with the zenith angle of the normal, the plot of the Eqs. (4.3) and (4.5) are show in Fig. 4.2, for different indexes of refraction in the range $\eta = 1.3$ and $\eta = 1.6$.

In sum, the normal to a surface point can be determined from the polarization state of the light. Indeed, if the index of refraction is known, the azimuth and zenith angles can be estimated as explained above. Nonetheless, there is not a unique solution because if the reflection produced by the object is purely diffuse, two possibilities for the normal are given due to the π-ambiguity in the azimuth angle. If the reflection is purely specular, there are four possible normal vectors, since we have 2 possibilities for the azimuth, and two possibilities for the zenith. In the real world, the reflections are in general a combination of diffuse and specular light, thus an approach must be chosen to select which reflection model best fits the problematic to solve. In any case, the information given by the polarization state of the light geometrically constraints the problem we are trying to solve, and these constraints can be used to improve the results obtained with color-only based approaches.

4.2.2 Hypothesis and normal definition

From the explanation of the previous section, we deduce that an algorithm that uses the physics theory contains several constraints: the type of reflection needs to be known to discard ambiguities, the index of refraction of the material needs to be known, and a way to disambiguate the vectors and find the correct one has to be implemented. Despite these difficulties, it is possible to solve the problem. For the index of refraction, as mentioned in [6, 8, 97], the dependency of the normal vector with respect to this value is weak, and it will only affect the zenith estimation. Since for s the index of refraction is in the interval $[1.3, 1.6]$, we fixed this index to the value 1.5. In Fig. 4.2, we have plotted the equations of the zenith angle for diffuse and specular reflections, for three values of the index of refraction: 1.3, 1.6, and 1.5. As we can notice, the difference among the three cases is very small, except for the diffuse reflection when the zenith angle is higher than $60°$. Therefore fixing this value will not introduce a large error if the plane inclination is lower than $60°$.

The next point to consider is the Camera Coordinate Frame (CCF), and how the measured angles are related to it. This will allow us to construct the normal vectors to the surface based on the polarization information.

The first aspect to note is that, for a given surface, at each point there are two possible unit normals to the surface: the one pointing towards the camera, and the one pointing

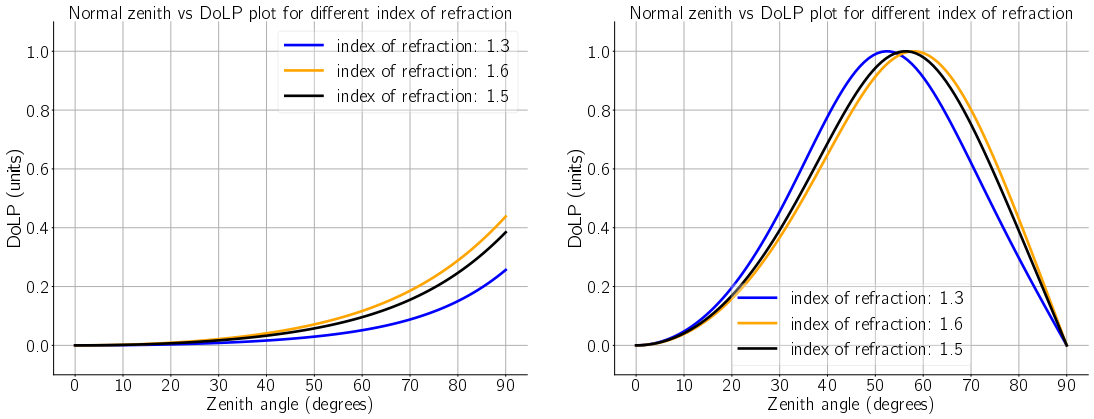


Figure 4.2: Plots of the DoLP as a function of the zenith angle for different values of the index of refraction η. (a) Diffuse reflection. (b) Specular reflection.

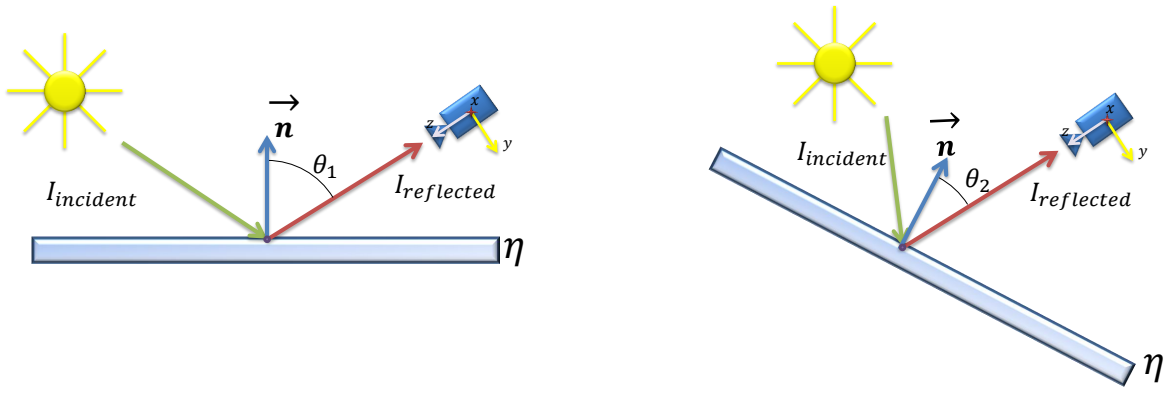
outwards the camera. If the two normals are $\overrightarrow{\mathbf{n_1}}$ and $\overrightarrow{\mathbf{n_2}}$, this aspect implies that $\overrightarrow{\mathbf{n_1}} = -\overrightarrow{\mathbf{n_2}}$. At the end, the vector that points towards the camera is the one that has a negative Z axis value.

Secondly, we have to consider the increasing direction of the azimuth angle. In the polarization measurement system used in the DoFP sensor, the Angle of Linear Polarization increases in the counter-clockwise direction. On the other hand, the positive direction in the CCF is clockwise direction, since the Y axis is always pointing downwards. Therefore, $\angle\overrightarrow{\mathbf{n}}_{azim} = -\phi_{disam}$, where ϕ_{disam} is the angle given by the AoLP constraint, already disambiguated.

Another geometrical constraint is that the zenith angle of the seen normal is in the range $\left[0, \frac{\pi}{2}\right]$. Indeed, since we can only reconstruct the vectors that are in the field of view of the camera, only the angles in that range have to be considered.

Finally, based on the electromagnetic theory, the zenith angle of the normal vector is measured from the Z axis. Thereby, if the normal to the surface is parallel to the Z axis, the zenith angle is equal to zero, and if the normal to the plane is perpendicular to the Z axis, the zenith angle is equal to 90°. This fact is illustrated in Fig. 4.3.

The equation of a normal vector whose azimuth angle α measured in the clockwise direction, and with a zenith angle β measured as the elevation of the vector with respect to



(a) (b)

Figure 4.3: Sketch of the evolution of the zenith angle θ of the normal vector to the plane with respect to the Z axis of the CCF. The index of refraction of the object is η. (a) The zenith of the normal to the plane is θ_1. (b) The zenith of the normal to the plane is $\theta_2 < \theta_1$.

the XY plane is:

$$\vec{\mathbf{n}} = \begin{cases} x = \cos(\alpha)\cos(\beta) \\ y = \sin(\alpha)\cos(\beta) \\ z = \sin(\beta) \end{cases} \tag{4.7}$$

If we consider the constraints given above, the normal vector as a function of the angles estimated from the polarization state is:

$$\vec{\mathbf{n}} = \begin{cases} x = -\cos(-\phi)\sin(\theta) \\ y = -\sin(-\phi)\sin(\theta) \\ z = -\cos(\theta) \end{cases} \tag{4.8}$$

By applying the trigonometric properties, we obtain the final equation of the normal from polarization:

$$\vec{\mathbf{n}} = \begin{cases} x = -\cos(\phi)\sin(\theta) \\ y = \sin(\phi)\sin(\theta) \\ z = -\cos(\theta) \end{cases} \tag{4.9}$$

Eq. (4.9) fulfils all the constraints to bound the CCF and the polarization theory:

- Given that the zenith from polarization goes from $\left[0, \frac{\pi}{2}\right]$, the Z axis is in the interval

$[-1, 0]$. Therefore, the normal points towards the camera.

- When the normal vector is parallel to the Z axis (i.e., the zenith angle is equal to zero), the X and Y coordinates are zero.

- When the zenith θ is not zero, and the AoLP rotates in the counter clockwise direction, the projection of the normal onto the XY plane also moves in the counter-clockwise direction. Since we take the vector that points towards the camera, this vector starts from the negative X axis when the azimuth is zero, and its projection on the XY plane moves towards the Y axis as the azimuth increases.

4.2.3 Normal vector disambiguation

With the details included so far, from a given polarization measurement, there exist several possibilities for the normal vector at the measured point. Therefore, a method to disambiguate the azimuth and the zenith angles must be found. If the reflection type is unknown, then we have 6 possibilities per point to consider. In general, the works that propose SfP algorithms try to increase the constraints of the system, or to make assumptions or approximations to be able to automatically obtain the normal vector that solves the problem. In our case, we are not developing or evaluating a system capable of doing that. Instead, we are interested in showing how the polarimetric calibration improves the measurements and what is its impact on the normal estimation. Therefore, the disambiguation will be done considering we know the ground-truth normal to the plane.

Let us consider the ground-truth normal to be $\overrightarrow{\mathbf{n_{gt}}}$. This vector can point towards the camera or away from it. As in other polarimetry works [18, 25, 97], we assume the object is either diffuse-dominant or specular-dominant. In other words, even though the light is a mixture of diffuse and specular intensities, we assume that there is one that contributes more than the other to the normal vector.

Additionally, we assume there is a set of ambiguous normals $\mathbf{A}$ to the same point given by the polarization theory: $\mathbf{A} = \left[\overrightarrow{\mathbf{n_1}}, \overrightarrow{\mathbf{n_2}}, \overrightarrow{\mathbf{n_3}}, \overrightarrow{\mathbf{n_4}}, \overrightarrow{\mathbf{n_5}}, \overrightarrow{\mathbf{n_6}}\right]$. Then, the function $f\left(\overrightarrow{\mathbf{n_{gt}}}, \overrightarrow{\mathbf{n}}\right)$ defined

in Eq. (4.10) can be computed:

$$f\left(\overrightarrow{\mathbf{n_{gt}}}, \overrightarrow{\mathbf{n}}\right) = 1 - \|\overrightarrow{\mathbf{n_{gt}}} \cdot \overrightarrow{\mathbf{n}}\| \tag{4.10}$$

where $\overrightarrow{\mathbf{n}}$ is one of the ambiguous normals in $\mathbf{A}$, $\|(.)\|$ is the absolute value operator, and $(.)\cdot(.)$ denotes the dot product operator. Then, $f\left(\overrightarrow{\mathbf{n_{gt}}}, \overrightarrow{\mathbf{n}}\right)$ is zero if the estimated vector $\overrightarrow{\mathbf{n}}$ is parallel or anti-parallel to $\overrightarrow{\mathbf{n_{gt}}}$, and it is equal to the value 1 if they are orthogonal. Therefore, this function can be used to find which is the best ambiguity vector to fit the ground-truth normal. In general, the estimation will not be equal to the ground-truth vector (i.e., we are not going to get a perfect zero for $f\left(\overrightarrow{\mathbf{n_{gt}}}, \overrightarrow{\mathbf{n}}\right)$) for several reasons: sensor noise, discretization noise, approximate index of refraction, non-ideality of the filters, and the fact that we are not considering a mixture of light reflection models (specular and diffuse). Nevertheless, we can ensure that the chosen normal pixel-wise will be the one that produces the lowest reconstruction error. Additionally, we do not need to take into account the reflection type since we disambiguate all the normals at once.

We again stress the fact that we are not aiming to develop a new SfP method, but just to establish a pipeline that will allow us to quantify the impact of the calibration correction in the input measurements on the estimation of the normals. In what follows, we will detail our method to estimate the ground-truth normal vector to the plane surface.

4.3 Ground-truth normal estimation and Region of interest selection

In this section we will detail a method to estimate the normal vector to the plane. The first consideration is regarding the index of refraction value. Since we aim to show the results of the most basic SfP method by using pure physics, we have to chose a uniform object, with a uniform material without texture change. This way, the index of refraction will be the same for all the points.

Then, to be able to quantify the error of the estimated normals, we need to know where are the points of interest. This is a requirement that allows us to be independent of the

environment in which the board to reconstruct is placed. In other words, instead of using a specific background material to easily hide objects around the objective plane, we use a mask image to erase everything that does not belong to the plane surface of interest. In what follows, we will explain the tools used to estimate the normal vector, and then we will detail how to combine them to obtain the final result.

4.3.1 Geometric Camera Calibration

All the operations required to estimate the ground-truth normal are grounded on the geometric camera calibration. This procedure is the base of many computer vision algorithms, in which any camera equipped with any lens is approximated by a given camera model (generally, either pin-hole either fish-eye). As any calibration procedure, the objective of doing the geometric calibration is to be independent of our acquisition setup. In our particular case, we will focus on the pin-hole model, since the lens we have used for this application is not a fish-eye lens.

The geometric calibration is a required procedure to restore the original aspects of the objects captured such that they respect the perspective geometry principles. By doing so, the camera can be used as a measurement device in applications in which the pose and size of objects are required.

The geometric camera calibration consists in estimating two elements: a projection matrix and a set of distortion coefficients. The projection matrix $\mathbf{P}$ consists of a mixture of two matrices: a pose matrix $[\mathbf{R}|\mathbf{t}]$ and the camera intrinsics parameters matrix $\mathbf{K}$:

$$\mathbf{P} = \mathbf{K}\,[\mathbf{R}|\mathbf{t}] \tag{4.11}$$

The pose matrix depends on where the origin of the coordinate system is placed. In the particular case in which the coordinate system is centered in the camera, the pose matrix has only ones in its principal diagonal and zeros elsewhere. The intrinsics matrix has a shape of 3×3 elements, and it is upper-triangular. It contains the information regarding the focal length, and the principal point of the camera system. With the projection matrix, a 3D point in homogeneous coordinates $\mathbf{X} = [X, Y, Z, 1]^T$ is projected onto the image plane at the

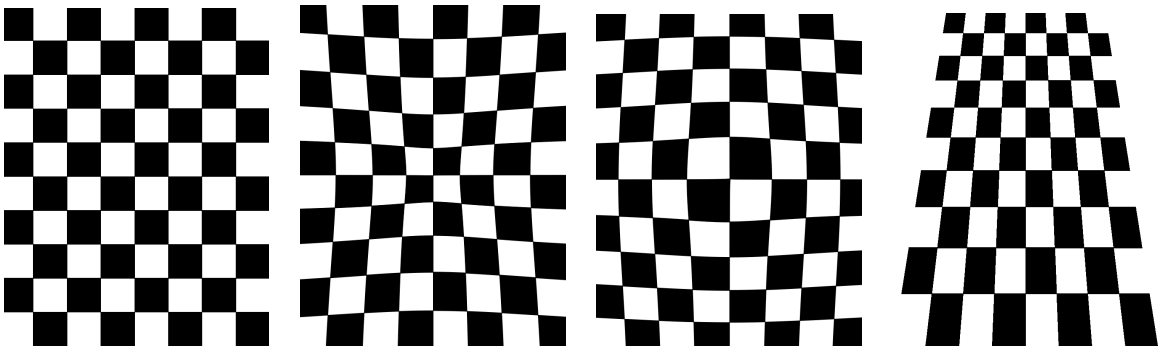


Figure 4.4: Examples of distortions produced by the lens. (a) Original image. (b) Radial distortion with negative displacement light rays (also known as *pincushion distortion*). (c) Radial distortion with positive displacement light rays (also known as *Barrel distortion*). (d) Tangential distortion due to the misalignment between the lens and the image plane.

homogeneous coordinates $\mathbf{x}' = [u, v, 1]^T$ following:

$$\mathbf{x}' = \mathbf{P}\mathbf{X} \tag{4.12}$$

u and v are, respectively, the columns and rows coordinates of the projection of the 3D point onto the image plane. Additionally, the fact of using a lens introduces a modification of the ideal pin-hole projection model and the real light projection over the sensor. This difference is mainly due to the light rays blend at the lens border (radial distortion) and due to the non-alignment between the lens and the image plane (tangential distortion). An example of these types of distortions are shown in Fig. 4.4.

Therefore, a function D is applied to the ideal pin-hole pixel location, giving the modified pixel location $\mathbf{x} = D(\mathbf{x}', \mathbf{s})$. In this equation, $\mathbf{s}$ is a vector with the estimated coefficients that consider the radial and tangential distortion introduced by the usage of a lens.

Several algorithms are available to geometrically calibrate a camera with a lens. In all the cases, a set of 3D points and their 2D projection on the camera plane are required. If the pin-hole model and the two distortion types are considered, a set of at least 10 pairs of 3D-2D points are needed. Thus, a certain calibration rig is required in which the 3D data is known, and once the pairing is done with the measured 2D points, the calibration problem can be solved. For instance, the Direct Linear Transform (DLT) method [86] for the intrinsics parameters, and an optimization routine for the distortion can be used. Since the DLT method is not constraint to a particular calibration rig, any system able to provide

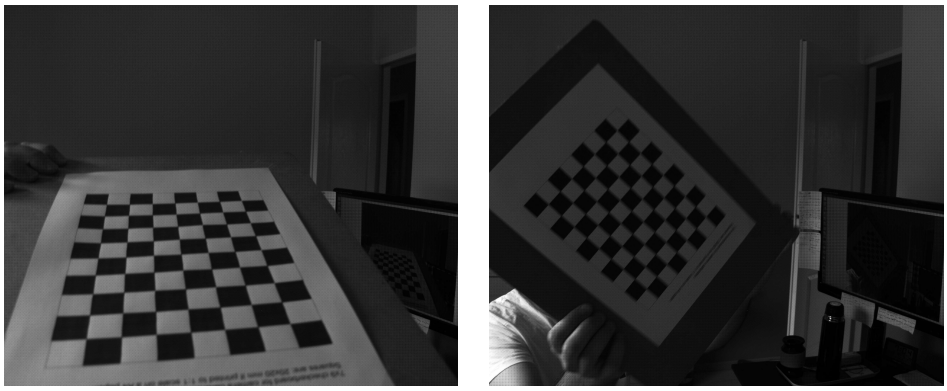


Figure 4.5: Examples of raw images captured with the color-polarization camera. These images of the checkerboard pattern are then used to compute the calibration parameters.

pairs of points can be used. For instance, a specific system with bright LED lights can be built with the most strict precision, and by placing the camera at a given position, the projection of the 3D points can be measured and paired. In practice, and due to equipment availability, Zhang's method is preferred [122]. In this method, the calibration rig consists of a rectangular, planar checker board pattern, as the one shown in Fig. 4.4 (a). One of the key points of this calibration algorithm is that the coordinate system is supposed to be placed over the pattern, and the Z axis points outwards to it. In this way, all the Z coordinates of the 3D points are equal to zero, reducing the amount of parameters to consider. Furthermore, the squares of the pattern have a fixed known size, measured in millimeters. Therefore, the 3D points are known all the time. Then, thanks to the particular structure of the pattern, it is possible to accurately detect the intersection points of the white and black squares in the image, and since it is rectangular, the correspondence between the 3D and 2D points can be established. To obtain a good calibration quality, several images of the calibration pattern, at different distances and orientation must be taken. From practice, it is recommended to use more than 20 images to obtain a good estimation of the intrinsics parameters. Examples of the raw images obtained with the color-polarization camera are shown in Fig. 4.5.

4.3.2 ArUco markers

To estimate the ground-truth normal, points that belong to the plane surface are required. More specifically, we need a method to extract at least three points from a textureless object,

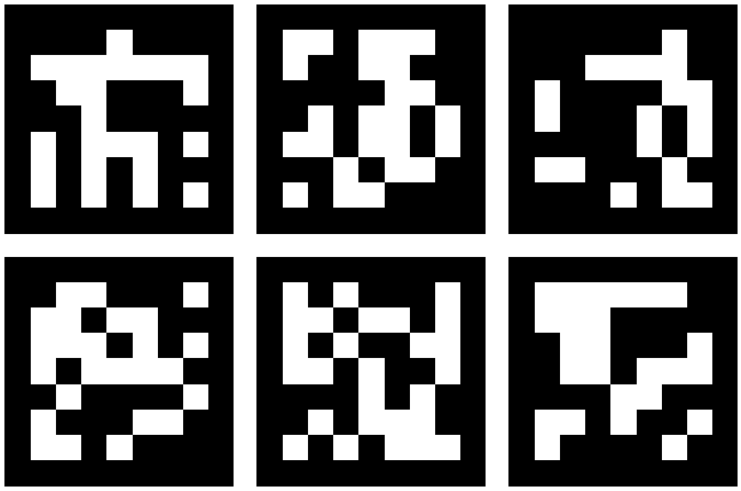


Figure 4.6: Example of ArUco markers used. The markers encodes a binary word, in which the black color represents 0 and the white color represents 1.

made of a uniform material. Since it is not possible to extract features of such an object, the simplest and most robust method is to add easy-to-identify objects over the plane. From the available options, we have chosen the ArUco markers [34].

An ArUco marker or ArUco tag is a fiducial marker similar to a QR code, but it is not intended to encode large strings of characters. Instead, they store only a word that represents a unique identifier for that tag. They are aimed to enable robust, real-time detection and pose estimation in robotics and Augmented Reality (AR) applications through vision algorithms. To enable detection, decoding, and pose estimation, the calibration parameters need to be known to undistort the raw images from the imaging sensor.

The first step to use the markers is the generation step. The original paper [34] provides an algorithm to generate a set of markers called the *dictionary of markers*. The used markers are generally made of black and white squares, as shown in Fig. 4.6. Based on the size of the markers and the amount of possibilities that are required to be generated for the given application, the dictionary of options will change. In any case, a new generated marker must comply with certain conditions:

- The minimum Hamming distance [3] between the new marker and all the other markers in the dictionary must be greater than a value τ. This consideration should be also valid for the four possible rotations of the new marker $(0°, 90°, 180°, 270°)$.

- The minimum Hamming distance between a new marker and its possible rotations should also be greater than τ. This will increase the robustness of the markers in pose estimation applications.

An iterative process is run in which the threshold τ is reduced to allow the generation of more tags. The minimum τ value is defined by the user, and its initial value is estimated as described in [34]. It is important to note that only a single position of a marker is stored in the dictionary.

The detection part consists of a series of image processing functions applied to the undistorted image from the camera to find the good marker candidates. These operations include local adaptive thresholding, edge detection, and 4-vertex polygon approximation. Once the markers have been enclosed by a minimum polygon of 4 vertices, and some operations to reject badly formed detected tags have been run, the marker is projected into a perfect square shape by warping the detected marker. The resulting image is then binarized and divided into a grid to transform the image into a code. Finally, to identify to which marker the detected element belongs to, four possible codes are extracted (one per possible rotation of the square). Then, a search is performed over all the elements in the dictionary until the one with the closest distance is found.

With this information (the image region where a code is, and its non-rotated version from the dictionary), it is possible to extract the four corners of the marker square, and with them, estimate the projection from the camera plane into the 3D world. This projection is given by a projection matrix that encodes the pose of the tag with respect to the CCF. In other words, we obtain the 3D coordinates of the pose of the tag. Finally, if that tag is pasted over the plane surface, we obtain the 3D coordinates of a point that belongs to the plane. An example of the results of detection and pose estimation of the tags is shown in Fig. 4.7 (b).

4.3.3 Ground-truth normal estimation

Once the camera intrinsics parameters are found, the perspective projection matrix that converts 3D into the 2D points in the image plane is obtained. With it, the camera can be

used as a measurement device. Particularly, we can estimate the 3D coordinates of points at which an ArUco tag is placed in the CCF. If we place these markers over a plane surface, we consequently obtain the poses of points that belong to the plane. The more tags we place, the more points we obtain. If we know the pose of N points in the plane, with $N > 3$, it follows that the normal vector to the surface can be computed. Indeed, let us consider three points in the plane P_A, P_B, and P_C. Then, two vectors parallel to the plane are $\overrightarrow{AB} = P_B - P_A$ and $\overrightarrow{AC} = P_C - P_A$. A unit normal vector to the plane can be obtained by the cross-product of these vectors, as follows:

$$\overrightarrow{\mathbf{n}'} = \frac{\overrightarrow{AB} \times \overrightarrow{AC}}{\|\overrightarrow{AB} \times \overrightarrow{AC}\|} \tag{4.13}$$

As mentioned in Sec. 4.2, the same surface point can be addressed towards the camera or outwards to it. However, we are interested in the vector that points towards the camera. To find this vector, we need to consider the viewing direction, which can be defined as the vector that goes from the origin of the CCF towards the plane. This vector can be found with any 3D point of the plane, for instance, P_A. Therefore, the viewing direction is $\overrightarrow{\mathbf{v}} = P_A - P_{\mathbf{0}}$, where $P_{\mathbf{0}}$ is a 3D point filled with zeros (i.e. the coordinates of the origin in the CCF). Finally, to disambiguate the obtained normal vector, and obtain the ground-truth normal vector $\overrightarrow{\mathbf{n}}$ to the plane surface, we have that:

$$\overrightarrow{\mathbf{n}} = \begin{cases} \overrightarrow{\mathbf{n}'} & if \quad \overrightarrow{\mathbf{n}'} \cdot \overrightarrow{\mathbf{v}} < 0 \\ \\ -\overrightarrow{\mathbf{n}'} & a.o.c. \end{cases} \tag{4.14}$$

Since the proposed system to estimate the normal vector to the surface is not perfect, some technique should be added to make the estimation more robust. This step is required since the calibration result is not perfect, the pixel positions are discrete, and not continuous in the sensor, the environment and the electronics introduce noise to the intensities, and the ArUco detector is not perfect neither to estimate the exact position of the center of the marker. To reduce the estimation error, we have to consider more than three points to estimate the normal, so that we can find a consensus between the estimated normal vector

and the fitting error of all the points to the resulting plane. Nonetheless, a large number of tags is not a good idea neither because we will reduce the effective area that we can use to estimate normals by using the polarization theory. For the work done in this thesis, we have decided to use four points, and we establish an optimization routine to find the best normal out of the measured points.

The optimization routine uses the Cartesian form of a plane surface. Let us consider a plane surface defined by the normal vector $\overrightarrow{\mathbf{n}} = [n_1, n_2, n_3]^T$. Then, the Cartesian formula of a plane is:

$$\overrightarrow{\mathbf{n}} \cdot \mathbf{x} + K = 0 \tag{4.15}$$

where the operation $\mathbf{v_1} \cdot \mathbf{v_2}$ is the dot product between the vectors $\mathbf{v_1}$ and $\mathbf{v_2}$, and K is a constant that can be found if a point that belongs to the plane is known. The coordinates $\mathbf{x} = [x, y, z]^T$ are the 3D coordinates of a point that belongs to the plane. Thus, with one of the points of the plane $\mathcal{P}$, and the normal given by Eq. (4.13), the variable K can be estimated as:

$$K = -\overrightarrow{\mathbf{n}} \cdot \mathbf{x} \tag{4.16}$$

Therefore, the equation of the plane is fully determined as in Eq. (4.15). Then, a distance function of a point to the plane is given by $d(\mathbf{x})$:

$$d(\mathbf{x}) = |\overrightarrow{\mathbf{n}} \cdot \mathbf{x} + K| \tag{4.17}$$

The optimization routine will generate a set $\mathcal{C}$ containing all the possible triplets from the N input points, and use them to derive the equation for all the possible planes. For each triplet of points, this routine computes the distance $d(\mathbf{x})$ for the remaining $N - 3$ points, and counts how many of them have a distance smaller than a threshold Ω. The number of points that fulfils this threshold are the inliers of the set. Finally, the best normal vector will be the one that has the largest number of inliers. The algorithm implemented to find the best normal vector to a surface when there are more than three points is detailed in Alg. 3.

The ground-truth normal estimation pipeline can be summarized as:

1. Calibrate the camera using Zhang's method [122],

Algorithm 3 Optimization routine to find the best normal vector to a plane.

1: **Input**: Set of $N > 3$ points that belong to the plane
2: Create a set $\mathbf{C}$ with all the possible combinations of triplets.
3: **for** each triplet in $\mathbf{C}$ **do:**
4: Compute the plane equation (Eq. (4.15))
5: For the rest of the set $\mathbf{C}$, compute $d(\mathbf{x})$ (Eq. (4.17))
6: Find the number of inliers with Ω
7: Store the number of inliers and the plane equation.
8: **end for**
9: Extract plane equation with maximum inliers.
10: **Output**: Best normal vector to plane

2. Generate a set of ArUco tags [34], and print them in black and white,

3. Paste at least 4 tags over the board to which we want to estimate the normal vectors,

4. Use the ArUco detection algorithm to estimate the pose of each tag,

5. From the estimated pose, extract the translation vector of the central point of each of them, and

6. Compute the normal vector to the surface using Alg. 3.

4.3.4 ROI retrieval

Unless the plane surface to reconstruct is relatively close to the camera, background objects will appear in the image. Additionally, since the markers to estimate the ground-truth normal will be placed on the board itself, these areas must be also removed from our image. These steps are required since our hypotheses for reconstruction is that all the considered points share the same index of refraction, and they are placed over a plane surface. Any other object in the scene will necessarily violate one of these hypotheses. Therefore, we need to remove them. One idea will be to cover the background area with a cloth material, and then do some image processing to remove unnecessary pixels in the image. Nonetheless, this does not solve the problem of removing the regions that are covered by the tags.

A more complex, but also more flexible approach, will be to create a mask that automatically adapts to the view of the board we have. This is possible since we know the configuration of our board (it is us who decided how to place the tags), and since we have the camera intrinsics parameters. This mask will allow us to consider only the pixels that are of our interest, and removing the rest.

As mentioned above, in our work, we have decided to use only four tags since it is more than the minimum required, and it allows us to have the largest profitable region of the board. More precisely, we have placed our tags in the corners of the board, and we have measured with a meter the horizontal and vertical distance of each tag with respect to the top left corner. For each tag, we have also noted its unique identifier jointly with its center coordinates in the board frame. All the regions of the board covered by the tags have been set to zero, and the rest of the board has been set a value of one.

To do the warping, we consider the measured coordinates of the center of each tag, and the estimated position of that point given by the camera. With the unique identifier of each tag, it is possible to make the correspondence between a point in the software-created mask, and the respective central point in the tag. Since we have four tags, we have four pairs of points, therefore, it is possible to compute the projection matrix that will produce the perspective warping of our mask into the image.

The projection matrix $\mathbf{H}$ is a 3×3 matrix that will take a point with coordinates $\mathbf{x_1^i} = [x_1^i, y_1^i, 1]^T$ and it will convert it into another point with coordinates $\mathbf{x_2^i} = [t^i \cdot x_2^i, t^i \cdot y_2^i, t^i]^T$, where $\mathbf{x_1^i}$ is the i^{th} correspondence point in the first view, $\mathbf{x_2^i}$ is the corresponding point in the second view, and t^i is a scaling factor. For a pair of points, the projection matrix should comply with the following relationship:

$$\mathbf{x_2^i} = \mathbf{H}\mathbf{x_1^i}$$

$$\begin{bmatrix} t^i x_2^i \\ t^i y_2^i \\ t \end{bmatrix} = \mathbf{H} \begin{bmatrix} x_1^i \\ y_1^i \\ 1 \end{bmatrix} \tag{4.18}$$

To compute this matrix, only four pairs of corresponding points are required. This projection matrix will consider a rotation, translation, and scaling operations of the points in the first view into the second view. Once the matrix $\mathbf{H}$ is obtained, the intensity of a pixel at the coordinates $\mathbf{x_1} = [x_1, y_1, 1]$ in the image I_{in} is placed at the coordinates $\mathbf{x_2} = [x_2, y_2, 1]$ in the image I_{out}. This operation maps the intensities as follows: Given a transformation matrix $\mathbf{H}$ from the view 1 to the view 2 such that $t\mathbf{x_2} = \mathbf{H}\mathbf{x_1}$, where t is a scaling factor, the intensity

from the view 1 I_{in} is converted into the intensity in the view 2 I_{out} as follows:

$$I_{out}\left[x_2, y_2\right] = I_{in}\left[x_1, y_1\right] \tag{4.19}$$

Since this operation may leave blank pixels after the mapping, an interpolation algorithm is usually added to the pipeline. The output image I_{out} is the result of warping

In sum, to be able to easily find the useful pixels in the image by software, we need to:

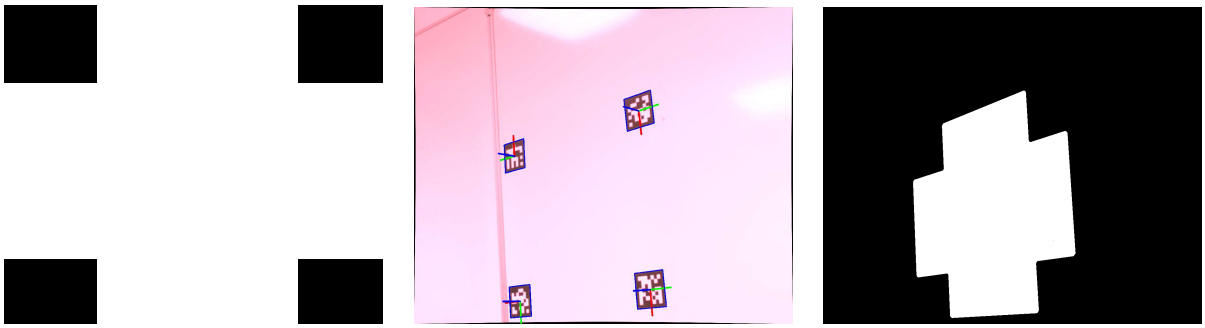
1. Create our pattern of tags on the board, with four tags placed at each corner of the board,
2. Put the board in a particular position, and measure the coordinates of the central point of each tag, and note their unique identifier,
3. Create a mask by software, with the board size as image size, and ones in all the places where the tags are not present,
4. Run the detector on the particular configuration of the board, and extract the tags IDs, and the coordinate of their central point,
5. Compute the projection mask from the measured coordinates to the camera estimated coordinates,
6. Do the warping of all the pixels of the original mask to the current board view.

An example of the mask warping is shown in Fig. 4.7.

This mask has a value of one for the valid pixels, and zero elsewhere. Therefore, only the normals at the pixels where this warped mask has a non-zero value have to be considered for the error estimation. In other words, the points at which this mask has a value of one defines our region of interest.

4.4 Effects of the calibration over the normal estimation

In this section, we will summarize the results obtained with the Shape from Polarization algorithm explained in Sec. 4.2. Our setup consists of a color-polarization camera equipped



(a) (b) (c)

Figure 4.7: Example of mask used to extract the pixels from the region of interest. A white color corresponds to a valid pixel, and a black one means that pixel is not useful. (a) Mask created out of the measurements on the real board. Only the internal region, whose limits is set by the tags positions is valid. (b) Color image captured with the color-polarization camera of a board with a set of ArUco tags on it. (c) Mask warped to the image captured by the camera.

with the Sony Polarsens IMX250MYR sensor. We use the Fujifilm Fujinon HF16XA-5M lens which has a focal length of 16mm. The lens aperture has been set to f/8, to meet the requirements of the polarimetric calibration procedure established in [57], and to have a camera behavior close to the pin-hole model. The camera has been calibrated geometrically as described in Sec. 4.3.1, and with our polarimetric calibration algorithm detailed in Sec. 3.2. The camera focus has been correctly set and fixed for all the experiments.

4.4.1 Experiments

We tested several board materials, made of glass or glossy reflective surfaces as a lacquered whiteboard. The index of refraction has been set to 1.5 in all the tests, and the tags to estimate the ground-truth normal to each surface are placed at the four corners of the region of interest. These regions have a size that varies between $[40 \times 40]$ and $[60 \times 60]$ centimeters.

The experiments have been carried our in several lighting conditions, with diffuse light coming either from the sun in a very overcast day, or LED lights bulbs with diffusers. For each image, we have performed the pipeline mentioned in the previous sections: we have detected the tags, we have created and warped the mask to consider only the region of interest, we have computed the ground truth normal, and we have estimated the normals at each super-pixel using the polarization theory and the ground-truth normal to disambiguate them. Then, for each super-pixel, we have computed two metrics to evaluate the quality of the estimated normal vector by polarization with respect to the ground-truth. These metrics

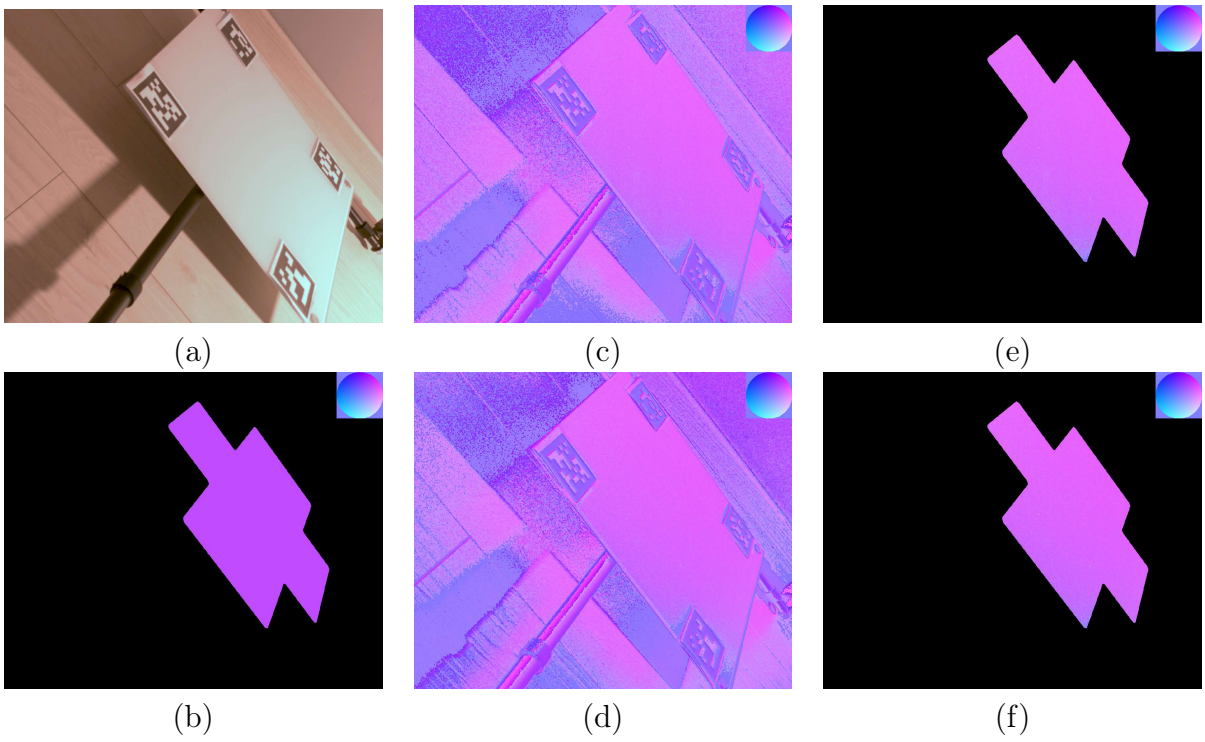


Figure 4.8: Results of the normal estimation algorithm using physics. (a) Original color image estimated from the measured intensities. (b) Ground-truth normal vector image to give a reference of the color to expect, only in the region of interest. (c) and (d) Uncalibrated and calibrated normal vectors estimated from the polarization theory, for the entire scene. Only the board area is considered for the error estimation. (e) and (f) Uncalibrated and calibrated normal vectors estimated from the polarization theory, masked to the area of interest.

are the Mean Angular Error (MAE) and the Root Mean Square Error (RMSE), defined as follow:

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |\arccos\left(\overrightarrow{\mathbf{n_i}} \cdot \overrightarrow{\mathbf{n_{gt}}}\right)| \tag{4.20}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{n=1}^{N} \|\overrightarrow{\mathbf{n_i}} - \overrightarrow{\mathbf{n_{gt}}}\|_2^2} \tag{4.21}$$

where $\overrightarrow{\mathbf{n_i}}$ is the i^{th} normal vector estimated from the polarization theory, N is the number of super-pixels in the region of interest, $\overrightarrow{\mathbf{n_{gt}}}$ is the ground-truth normal, and $\|.\|_2$ is the L_2 vector norm.

In Fig. 4.8 we show the image of the ground-truth normal for all the regions of interest, and the results obtained without and with the polarimetric calibration of the estimated normal vectors. The quantitative results of the metrics are shown in Tab. 4.1.

As mentioned above, we assume that the reflection produced by the board is either

Method	Reflection type	MAE	MAE Std.Dev.	RMSE
Uncalibrated	Only diffuse	19.8796	4.6641	0.355475
	Only specular	25.0383	1.25402	0.431848
	Both reflections	14.529	2.68189	0.256998
Polarimetric calibration	Only diffuse	18.4985	4.25462	0.332538
	Only specular	25.086	1.13723	0.432282
	Both reflections	**12.8839**	**2.49678**	**0.229201**

Table 4.1: Quantitative evaluation of the error in the normal estimation when using polarization physics-based theory. The index of refraction η is fixed to 1.5 for all the experiments. Only specular and only diffuse are the experiments in which only the specular and diffuse normals are considered when doing the disambiguation, respectively. The normal estimation algorithms have been performed using the raw measurements from the camera, and using the measurements corrected by the polarimetric calibration algorithm, introduced in Sec. 3.2.

diffuse dominant or specular dominant. This is a common hypothesis when doing Shape from Polarization. Nonetheless, that does not imply that all the pipeline will use only diffuse or only specular reflection. In fact, it considers all the normals at once, and the one that produces the less error pixel-wise is the normal that will be used. As a consequence, there will be pixels whose normal is given by the diffuse reflection formulae, and there will be pixels that will return a normal computed from the specular equations. The fact of considering both cases until the disambiguation step is clear in the results of Tab. 4.1. In that table, we have joined the results of using only diffuse theory and only specular theory. In both setups, calibrated and uncalibrated, using only one type of reflection increases the error metrics values with respect to the case in which both reflections are considered. This is expected, since in the most general case, a reflection is a combination of both specular and diffuse light [47, 113], but there is always one that has a higher weight than the other one. Therefore, the DoLP can be approximately assigned to one or another reflectance model. It is for this reason that there are points of the board where the normal is better estimated using the diffuse reflection model and others using the specular reflection model.

4.4.2 Discussion

From Tab. 4.1, it can be noted that the results obtained after calibration outperform the uncalibrated setup. Indeed, the fact of correcting the measurements by model fitting allows for compensating the lens distortion and the filter properties. Even though the calibration does not affect too much the base sensor measurements, we can see that when dealing with a SfP problem, this small correction is responsible for an improvement of 12.8% in the MAE

Index of refraction	**MAE**	**MAE Std.Dev.**	**RMSE**
$\eta = 1.3$	15.2286	3.09944	0.271558
$\eta = 1.7$	13.0311	2.31398	0.230977

Table 4.2: Evolution of the error in the normal metrics introduced by a bad index of refraction under a calibrated setup.

and 12.13% in the RMSE. Additionally, we can see that the standard deviation in the error has also been reduced, which is an expected effect of the calibration procedure. Indeed, after correcting the measurements in the sensor, two pixels that receive the same input light will produce readout values that are closer than a similar arrangement without calibration. This is translated as a reduction in the difference between the normal vectors of a plane, thus, a smaller distribution of values in the estimation errors. We should note that the sensor is already of high quality, so achieving improvement through calibration represents a significant step forward in the field of 3D reconstruction.

While the calibration algorithm produces improved results, it is important to note that the error remains relatively high compared to other methods. One of the main sources of this error is the sensing noise. To avoid sensor saturation, we have set up the exposure time and the sensing gain to a value that is enough to have measurements without reaching the maximum sensor readout value. Nonetheless, since each pixel has a different filter, the amount of light that passes through them is different. Additionally, since the sensor noise effect is relatively more prominent when the measurement is lower than when it is higher, avoiding saturation means that our measurements will be noisy. Then, this noise is transferred to the polarization measurements. Furthermore, there are two points that also justify this large error metrics values. Firstly, the index of refraction is not known with precision, and it changes depending on the wavelength. As a consequence, there is a deformation in the estimation of the normal field of the board due to that uncertainty. To show this point, we have attached the normal metrics for the calibrated setup, for the indexes of refraction $\eta = 1.3$ and $\eta = 1.7$ in Tab. 4.2.

The other point to consider is the error due to the diffuse-dominant and specular-dominant model consideration. As mentioned above, the measured light is a combination of diffuse and specular light, and there are two different mathematical models to describe their behavior. Therefore, in the measured DoLP there is a part of that component that belongs to the diffuse behavior, and another one that belongs to the specular one. Consequently, the

measured AoLP and DoLP are a combination of both models. To exemplify this situation, let us consider the zenith angle computation. If the DoLP is a combination of two contributions, the value to be used for each model (diffuse or specular) will be lower than the measured one, hence there will be a modification of the output normal zenith angle. Similarly, the AoLP given by each model is different, but there is no equivalent formula to relate to this angle with the normal azimuth. Instead, there is a set of four discrete possibilities between the AoLP and the azimuth angles.

A better formulation might be to separate the incoming light into the specular and the diffuse components, and to do a normal estimation for each model separately. Then, the model whose DoLP is the strongest can be used to estimate the normal vector field[1]. Nonetheless, the separation of the light into diffuse and specular components is a challenging problem to solve, and even though the problem has been addressed in several works using color [56, 67, 78, 119] and polarization images [47, 49, 102, 106, 113], there is still no generic solution for any situation or that does not need special lighting devices. The analysis and the effects of this separation is out of the scope of this thesis. We leave this topic as a clue to explore in the future.

4.5 Conclusions

In this chapter, we have given an in-depth description of a method to reconstruct the normal vectors to a surface by using the polarization theory. In this contribution of the thesis, we have covered all the required formulas to estimate the normal vectors from the polarization theory, we have presented the inverted functions of the Fresnel equations for both types of reflections, and we have considered the different types of materials and reflections. To the best of our knowledge, there is no document that describes the normal estimation from polarization at this level of detail. We have run experiments with the color-polarization camera, with and without calibration of the sensor, and we have found that there is an improvement in the reconstruction quality when a calibrated setup is used. This chapter served to objectively

[1]We should use the strongest value since it means that we will use the signal with the highest SNR value. If the model is correctly computed, then the two models should give the same normal vector.

evaluate the influence of the calibration in the most common polarization application: the SfP algorithm. For doing so, we developed a methodology that systematically allows us to estimate the ground-truth normal of a plane surface, estimate the ambiguous normals from the polarization constraints, and then disambiguate them by using the ground-truth data. Despite the high quality of the sensor, the presence of small parameter dispersion in the pixels and the use of a lens introduce a significant error that requires correction. Although the experiment setup has been carefully built, the error in the final reconstruction is still high. This error has several sources such as the sensing noise, the approximate value of the index of refraction, and the usage of the reflection-dominance model. This last source of error is due to the diffuse and specular components. If it has been possible to divide the information into diffuse and specular components, the resulting DoLP would have been smaller for each case, introducing a change in the zenith angle. This point is still a big challenge in the computer vision domain. There is no straightforward solution to split the input light into these two components, and to determine the corresponding diffuse and specular Stokes vector.

Chapter 5

Deep-learning depth estimation with polarization cues

Nowadays, data-driven approaches have proven to be more accurate than optimization algorithms based on hand-crafted designs for hard perception problems. Since the polarization data provides valuables geometrical constraints, we aim to use them in an important and challenging perception task: monocular depth estimation. In this chapter, we will describe our methodology that integrates the polarization cues into visual features to improve the results of a texture-based monocular depth estimation neural network. Furthermore, we include tests of the baseline networks, and of our method, concluding that our network correctly integrates the polarization measurements without degrading the baseline network performance.

5.1 Introduction

One of the main tasks of an autonomous navigation robotic system is to be able to understand the environment in which it is immersed, and to detect any obstacles as it moves. One of the foundations in this task is depth estimation. Different active sensors have been developed to ease this task, such as the Microsoft Kinect [123], or the LiDAR [110], but either the depth range is not large, or the information is scattered. Additionally, when the sensor measurement is based on the reflection of a light ray, it becomes hard to have correct depth measurements

for transparent or highly reflective objects. Since the properties of the polarization state for this type of materials impose geometrical constraints to the objects in the scene, we aim to improve the monocular depth estimation by using the color and polarization data. In this chapter, we design a deep neural network to outperform the depth estimation produced by algorithms that make exclusive use of color images in these hard conditions.

As in the rest of this thesis, we aim to bring more tools to the community that eases the integration of the polarization state into texture-based algorithms. We are convinced that there is a method to do so, in a systematic way, without penalizing the performance of the system that already works solely with color images. In general, texture-based algorithms fail when a surface is made of a single color, or of highly reflective materials, or if the object is transparent. In these types of objects, the polarization state of the light provides important clues to estimate the normal vectors as we have seen in Chapter 4. Therefore, we are looking for a system that can estimate depth whenever the texture information is not enough, but at the same time, do not degrade the performance of the system whenever the color information can correctly determine the depth.

Some previous works try to accomplish an improvement in depth estimation by using polarization, but the developed systems have different limitations. Berger *et al.* [10] and Blanchon *et al.* [11] use monochrome polarization cameras, thereby part of the texture information is lost during the acquisition, limiting the accuracy they could get. Both works make use of the azimuth constraint to quantify the orientation error of the normal vector to the surface estimated by the network with respect to the AoLP. Although effective, they only consider the specular reflections by doing a thresholding operation to the DoLP.

Other data-driven algorithms that make use of the polarization state of the light limit their usage to a particular environment, and condition. Ba *et al.* [8] introduce a supervised deep-neural network to estimate the normal field to a single object surface. In this case, the provided results do not vary by changing the lighting condition, but the network must receive an image with a single object to analyze. Similarly, Deschaintre *et al.* [26] present a data-driven approach to jointly estimate the normal field, the Spatially Varying Reflectance, and the depth map of a single object. Kondo *et al.* [53] developed a Polarimetric Bidirectional Reflectance Distribution Function to simulate the polarization state of the light using a

renderer, and with it, create images of random objects to train data-driven algorithms. This way, the acquisition problem is simplified since the images can be synthesized with ground-truth without user efforts. Although accurate, the system is aimed at analyzing the normal vector field of a single object at the time.

Finally, some developed algorithms have acquisition conditions far away from practical applications, as the one from Ichicaka *et al.* [45]. They use the sun to create a known Stokes vector and by measuring the change of the polarization state, they are able to reconstruct the normal vectors of a single object. The counterpart of this algorithm is that the method requires at least two measurements at two different moments of the day, during a clear day, and these images should be taken several hours apart.

The objective that we pursuit in this work of the thesis is to find a multi-modality fusion network that allows to:

1. Take the best of each modality to outperform the texture-based only algorithm,
2. Do close-to-real-time estimation of the depth. Therefore, a network with the minimum number of parameters possible, and
3. Assimilate the polarization data in a simple, systematic way, by using the polarization constraints.

Due to its training robustness and wide adoption in the computer vision community, we have decided to use as backbone the work from Godard *et al.* [36], which is a depth estimation network for urban scenes. In the rest of this chapter, we will detail the network architecture and the dataset we will use, the loss function designed to consider the polarization data constraints, and the training process we have implemented.

5.2 Deep learning-based depth estimation using a color-polarization fusion network

In this section, we detail our deep learning architecture designed to do depth estimation from monocular color-polarization images. The pipeline that we have developed is based on

the monocular depth estimation network Monodepthv2 [36].

Monodepthv2 is a self-supervised approach, i.e. no ground-truth data is required to train the network. The training process is guided by the input images, and a set of geometrical constraints that guide the network towards a correct depth estimation. Indeed, two images are compared (either from a stereo setup either from a monocular setup) through the projection of one image into the other one, using the Structural Similarity Index Measure (SSIM) to quantify the reconstruction error. Since the polarization state also constraints the geometry of the problem, it is a good candidate to improve the results obtained by the baseline network.

Secondly, Monodepthv2 has served as a base for other works by using either color-only cameras, or polarimetric cameras [11, 26, 58, 60, 89, 100, 112, 120], but either the improvement is marginal, or they were unable to highlight the improvements of the results where generally texture-only methods fail. To the best of our knowledge, there is no method that shows how the estimation over reflective, transparent or textureless surfaces are improved in the depth estimation sense. Additionally, for multi-modal methods (i.e. those that combine color and polarization), they do not show that the newer method performs similarly or better than the color-only method in the regions where the performance was already satisfactory.

We aim to show that it is possible to add the polarization constraints to a texture-based network to outperform the original results. In what follows, we will give the details of the candidate deep learning network we develop, and include the experiments used to evaluate its performance.

5.2.1 The baseline model architecture

The base network model Monodepthv2 is inspired by the general U-Net model [92], which consists of a feature encoder-decoder architecture with skip connections. This means information from the encoder is copied and added to the corresponding size feature vector in the decoder, providing spatial cues into the decoder reconstruction. The encoder model is a ResNet-18 [39], and the decoder used is based on the depth estimator introduced in [37]. This architecture has five skip connections between the encoder and the decoder. The main aspects carried out by Godard *et al.* [36] are:

1. A pose encoder is used to estimate the pose transformation between two consecutive frames.

2. Pixel-wise, the loss is the minimum of all the SSIM values computed for all the times t considered, with respect to the reference frame. This allows to not penalize the network when there are occluded pixels.

3. An auto-masking is performed to avoid using points that do not move between frames. This is required since the disparity estimation through geometry assumes that there is a relative movement in the pixels between two consecutive frames. Therefore, if the camera is stopped, or if there are objects that move at the same speed as the camera, this hypothesis is violated.

4. To avoid texture copy and depth "holes" in the output image, the loss is computed at the different layers of the depth decoder at the final output resolution. Thus, the output of each layer is first re-scaled to the output resolution, and then the loss is computed as if it is the output image. The result of this operation for all the decoder layers are added together to get the final loss value.

5. ResNet-18 has been chosen as the encoder architecture to obtain faster depth estimations than those that use the architecture ResNet-50. Even though the former has less trainable parameters than the latest, the results obtained with this architecture outperforms the others.

The network architecture of Monodepthv2 is shown in Fig. 5.1, and the appearance loss sketch is illustrated in Fig. 5.2.

5.2.2 Color-polarization monocular depth estimation architecture

We aim to improve this network by integrating the polarization data.

Since we would like to use two different modalities (color and polarization), a fusion mechanism has to be chosen. The first solution one might think is to take all the polarization data (either the polarization channels as intensities, or the DoLP and AoLP), concatenate

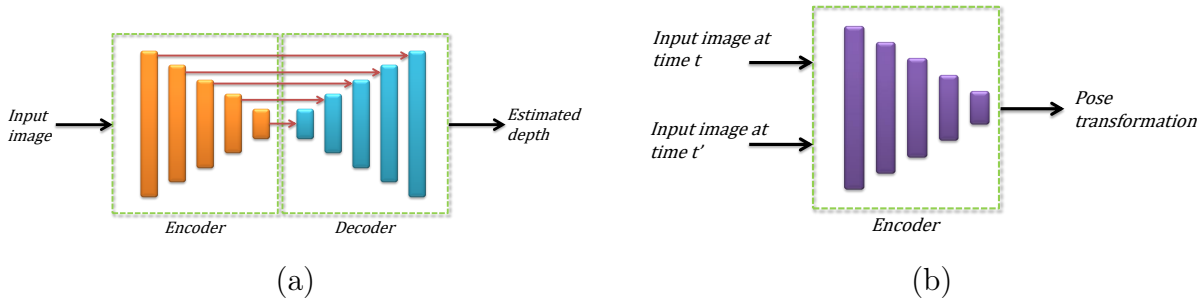


Figure 5.1: Monodepthv2 network architecture components. (a) An encoder-decoder with skip connection architecture is used to estimate the depth out from a single input color image. (b) Two images of the same scene at different time instants t and t' are used to estimate the pose transformation between them using a pose encoder. This information is used to estimate the reconstruction loss.

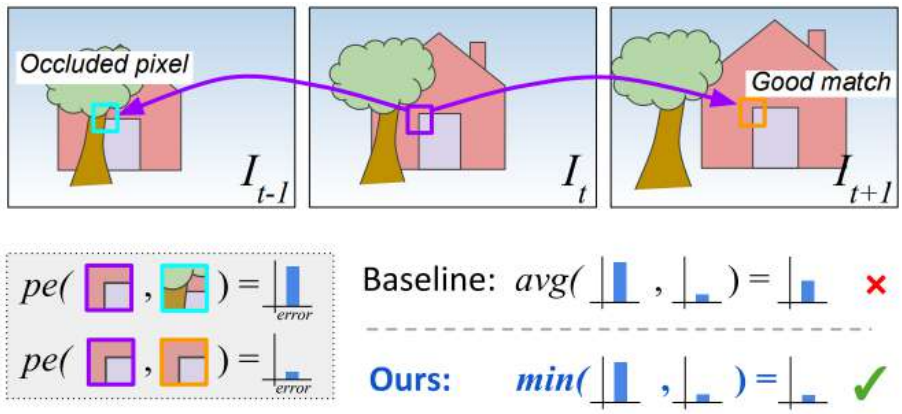


Figure 5.2: Sketch that explains how the introduced appearance loss works [36]. The reference frame is the image in the middle. Then, one image before and one after the time of the reference image are considered. In the instant before, an occluded pixel projected in the reference frame will give a large value in the SSIM loss. Nonetheless, for the instant after, the Structural Similarity Index Measure will give a low value for the same point. Between these two possibilities, the introduced loss considers only the minimum SSIM value.

them, and input this block of data to the network. Nonetheless, this type of approach generally does not maximize the performance that can be obtained if the network structure is adapted to the problem to solve [8]. Therefore, an adapted combination mechanism has to be developed. In the literature, there exists mainly three widely accepted fusion architectures: early fusion, middle fusion and late fusion [14].

Early fusion: the data is mixed in a certain way before entering the encoder network for feature extraction. This fusion mechanism is performed in general through operations that contain learnable parameters as the network, and these parameters are adjusted during the training process. All the fused data pass through a single encoder. Nonetheless, since the fusion is produced at early stages of the feature extraction process, the semantic information

that can be extracted is of low level, and in general, this type of fusion does not produce large improvement gains. Also, it is the fusion mechanism that does not modify the base network architecture, and it adds the lowest number of parameters to the training process.

Late fusion: the data from each modality passes through different encoders. The weights for each modality can be shared between them, but it is not a common practice since each modality requires to extract different type of patterns from the inputs. Once the input modalities have passed through the encoders, the high-level feature maps are fused into a single one that is then passed to the decoder module. In this architecture, the fused maps contain high-level features of the input images, but they are frequently hard to interpret, and it is not evident to find a good mathematical relationship to correctly combine them. The type of operations implemented in this fusion model are generally more complex than for the early fusion, which means a larger number of parameters are added to the network, resulting in slower responses in the forward pass.

Middle fusion: as for the late fusion model, the data from each modality is inputted to an independent encoder, but in this case, the feature maps in the encoder at the different hierarchical levels are fused. This means that at each network layer of the encoder, the corresponding feature maps are fused with a certain fusion architecture. Then, the output of the fusion modules are either combined into a single feature map and given to the decoder input, or passed to the decoder at the same feature size decoder layer to continue the estimation process. This last mechanism is similar to the skip connections process, but the transferred information has been processed by the fusion module. The middle fusion is the heaviest fusion architecture in terms of parameters and forward pass time. Nonetheless, it ensures a complete mixture of the data at all the levels in the network, avoiding information loss while the data passes from one layer to another. A representation of each fusion architecture for an encoder-decoder network is shown in Fig. 5.3.

Based on the details given above, to better exploit the multi-modality of color and polarization data, we will use an architecture with at least two encoders, with either a middle or a late fusion model. This is because the features that can be extracted from the polar-

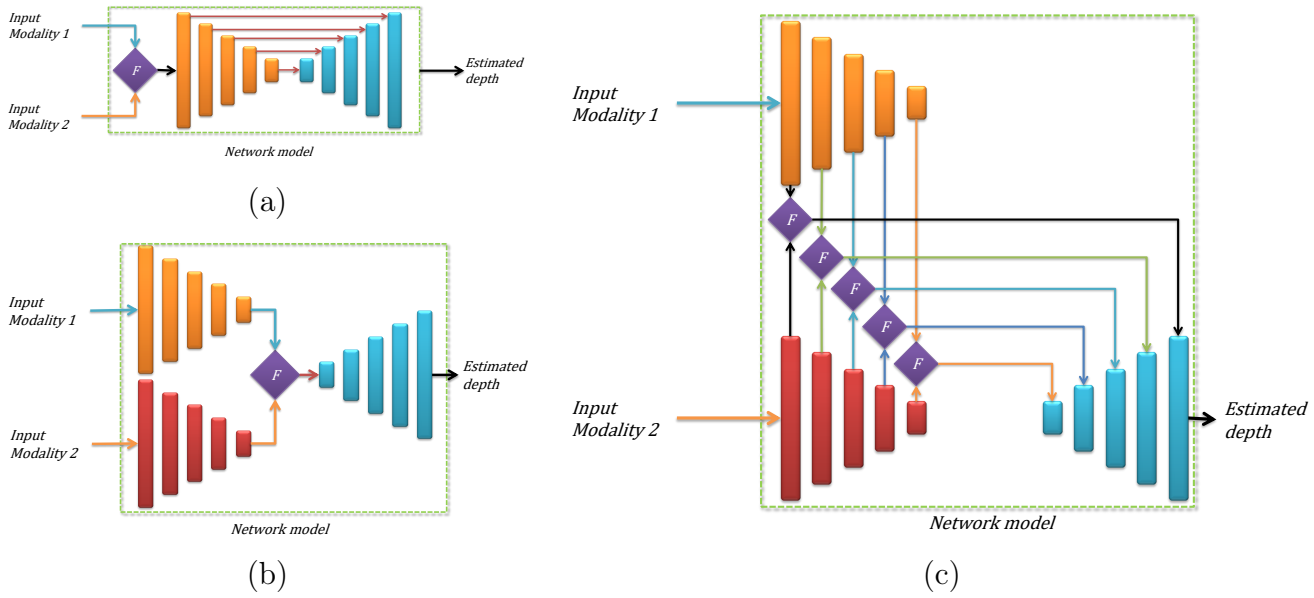


Figure 5.3: Fusion architectures. (a) Early Fusion: the combination of the different modalities is done before the feature extraction process. (b) Late fusion: the mixture of the multi-modality data is done at higher representation levels. (c) Middle fusion: the data is combined at different hierarchical levels of the network.

ization and the color data are not the same. Furthermore, we would like a network that is capable of distinguishing which modality provides the most relevant data for each object. Therefore, we need to have two encoders to extract the corresponding features, and then find a proper method to weight and fuse these features. Thereby, we will not consider an early fusion method. Physically, this decision is taken because the polarization parameters (the AoLP and the DoLP) are independent of the intensity. Therefore, it makes sense to have a branch that extracts the most from the textures, and another that takes advantage of the polarization, and then fuse the corresponding features of each modality.

Since we do not know which features are the most valuable for each modality (either low or high representation level features), we will start with a middle fusion architecture, because the migration to a late fusion architecture would be straightforward. This way, we will be also able to test both architectures.

5.2.3 Transformer-based or Convolution-based Neural Networks ?

The next question is which deep learning architecture for the encoder and the fusion module must be used to extract the most out of the input data in terms of features. Unfortunately,

there is no straightforward answer to this question. On the other hand, since the final result does not strongly depend on the decoder, we have decided to keep the original decoder of the Monodepthv2 network.

Regarding the encoder block, we can identify two categories of commonly used architectures: Convolution-based Neural Networks and Transformers-based Neural Networks. In the former case, the layer blocks are based on combinations of convolution, pooling, activation, and normalization operations. The convolution operations serve to filter the input feature map, the pooling serves to reduce the number of parameters in the network, and the normalization serves to reduce the spatial resolution of features. For a long time, this type of networks formed the state-of-the-art in terms of deep data-driven algorithms. One of the most known architecture is the Residual Neural Network, also known as ResNet [39]. Recently, a new network architecture called Transformers has been developed [107]. It is currently considered as the best performing architecture. Although the original concept was aimed at solving language interpretation tasks, it is now widely used to solve various computer vision tasks [28]. Nonetheless, this new type of network are data hungry, meaning that large datasets with large training times are required to outperform their predecessors. One of the state-of-the-art transformer for vision tasks is the Swin Transformer [69].

Regarding the fusion blocks, there exists three main mechanisms that can be used: Convolution-based, transformers-based, attention-based. The first two cases are similar networks as the ones presented before, but with less operations, or slightly modified versions of the entire networks. The attention mechanisms are sub-networks generally used to find weights to balance the contribution of different areas in the feature map. An attention mechanism can be defined as self-attention if the weights are computed based on the same input image, or cross-attention, if the weights are based on the feature map coming from another modality. There is no general rule of thumb to decide for one method or another. Nonetheless, it would be interesting to test a cross-fusion module in which the feature maps of one modality are used to find the weights for the other modality. This methodology might be used to determine the spots in which a certain modality does not contain valuable features for the depth estimation.

In sum, our network will follow a middle fusion architecture, with one type of network

for the encoder, one type of network for the fusion module, and the same decoder as for the original Monodepthv2. There are several possible combinations to chose an encoder and a fusion block, but there is not too many clues to narrow down this choice. In terms of development, we aim to build a system in which these two parts of the network can be replaced easily in order to test different combinations.

5.2.4 Training data

As for any deep learning network algorithm, a good set of data is required. In our case, since we want to do depth estimation from color-polarization images, we need to have a set of registered triplets of color, polarization, and ground-truth depth images. When we mention registered, we mean that three modalities have to be projected into a common reference frame. For the type of imaging sensor we aim to use, the color and polarization data are registered by definition. Nonetheless, the depth to color projection is required, and for that, a geometrical calibration algorithm has to be run. Additionally, the data needs to be synchronized in time, which is not an obvious task[1]. Finally, the amount of data required to train a depth estimation network needs to be large enough to allow the network to learn the most generic characteristics of the considered scenes. If there are not enough diverse information, the network will tend to memorize the answers to the shown scenes, producing good accuracy during training, but with a bad performance during testing. This phenomenon is known as *over-fitting*.

As mentioned in Sec. 2.3, there is no standard, high quality benchmark that all the polarization imaging researchers can use to evaluate their algorithms. Due to this, the authors of each paper create their own set of data, which is generally not large, not always available, and without the ground truth data for several applications (classification, object detection, navigation, scene segmentation, depth estimation, etc). This also makes it difficult to compare with other works in the same domain.

[1]Data synchronization means that the image capturing process done by a set of sensors must start at the same time. To do so, a triggering signal needs to be created and send to all the devices, the images should be captured after the signal has been received. Additionally, the cameras need to be configured accordingly, and the triggering signal generally sent through a specific connector, or through a specific Ethernet protocol, which are not included in all the available cameras.

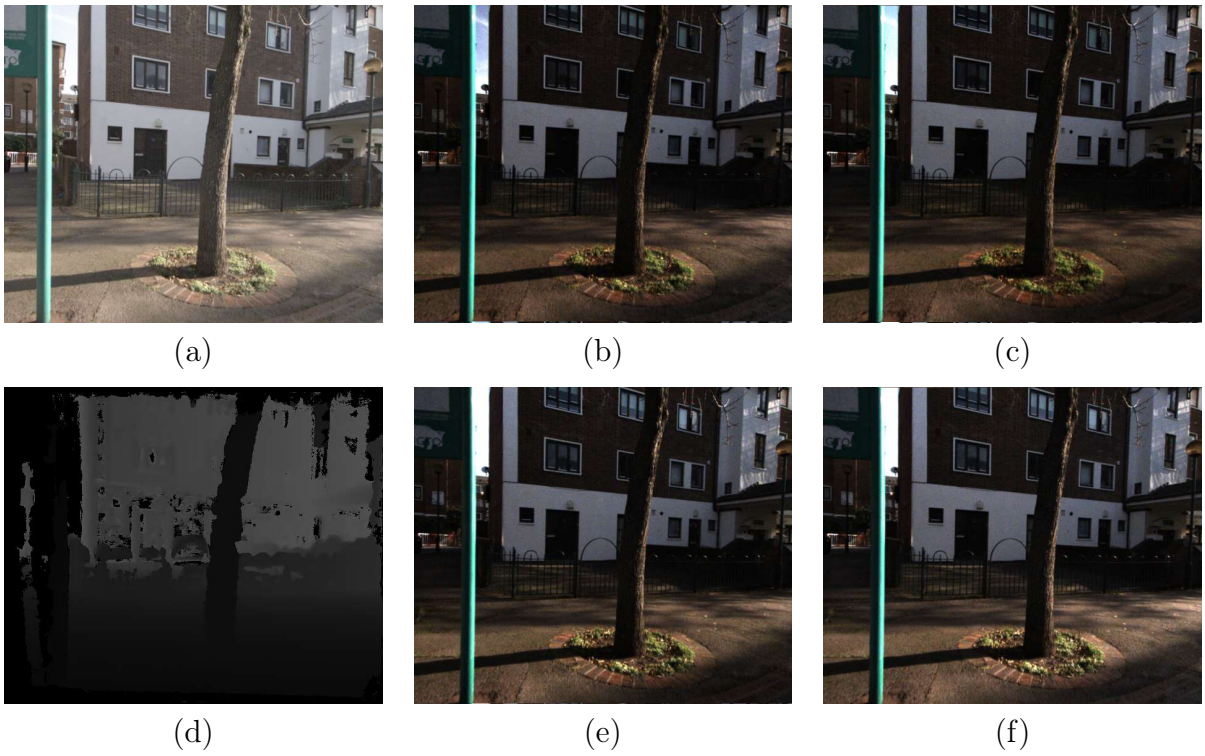


Figure 5.4: Cromo Dataset sample. (a) Color image. (b) Demosaiced polarization channel 0°. (c) Demosaiced polarization channel 45°. (d) Depth from COLMAP. (e) Demosaiced polarization channel 90°. (f) Demosaiced polarization channel 135°.

Recently, the CroMo dataset [108] has been released, which is a large-size dataset with registered depth, color and polarization images, and the parameters result of the camera geometric calibration. The acquisition rig consists of a pair of color-polarization cameras, indirect Time-of-Flight (iToF) sensor, structured light depth sensor, and an Inertial Measurement Unit (IMU) sensor. All the measurements have been projected into the left-polarization camera image plane. The captured scenes comprise a kitchen, a park, house facades, and a bus station. For each environment, acquisitions have been done in different days. The depth map has been retrieved from the structured light depth sensor, and it has been refined using the COLMAP pipeline [94]. For the work proposed in this thesis for depth estimation, we will use the color data, the polarized data, and the depth data. A sample from the dataset of these modalities is shown in Fig. 5.4.

It is important to note that the color images have been gamma corrected with a value $\gamma = 2.2$ approximately. That is why the color images look brighter than the polarization images[2].

[2]This detail has not been included either in the paper or in the dataset documentation

For the evaluation of our algorithm, we also selected the recent dataset from Baltaxe *et al.* [9]. This dataset contains color, AoLP, and DoLP measurements taken with an RGB-polarimetric camera, LiDAR measurements for depth, and GNSS sensor for global positioning. The authors provide a dataset split into train, validation, and test sets containing respectively 6116, 778, and 778 images. Even though smaller than CroMo, this dataset provides accurate depth samples of urban scenes with traffic signs, a large number of cars, people, and road pavements, captured with a car moving in the city. In contrast, CroMo depth measurements are obtained through COLMAP and the sequences were collected by a person walking in a variety of environments. These two datasets provide complementary information to evaluate the approaches in different conditions.

5.2.5 The input image encoding

The next important step is to determine the input data the network will need to perform its task. This data must be representative and informative regarding the type of problem we want to solve. If the information is insufficient or ambiguous, the results may not meet the expected level of performance. From the Monodepthv2 implementation, it is confirmed that for a variety of situations, the texture information provides important clues for determining the depth from a single color image. Therefore, this information should be kept, and it will be the input to one of our network branches. Our second modality, the polarization, has already been explored in the literature, and there is no universally adopted input format for this data. Some authors decided to use the full polarization parameters (AoLP, DoLP, and intensity) as inputs [11, 48, 72]. Others choose to include the raw intensities from the polarization channels [8, 12, 53, 59, 60], and there are authors that use the physics theory to estimate the ambiguous normal maps as input priors [8, 32, 127]. After considering the different options, we have narrowed the input formats to two. The first one is the same as adopted in [60]:

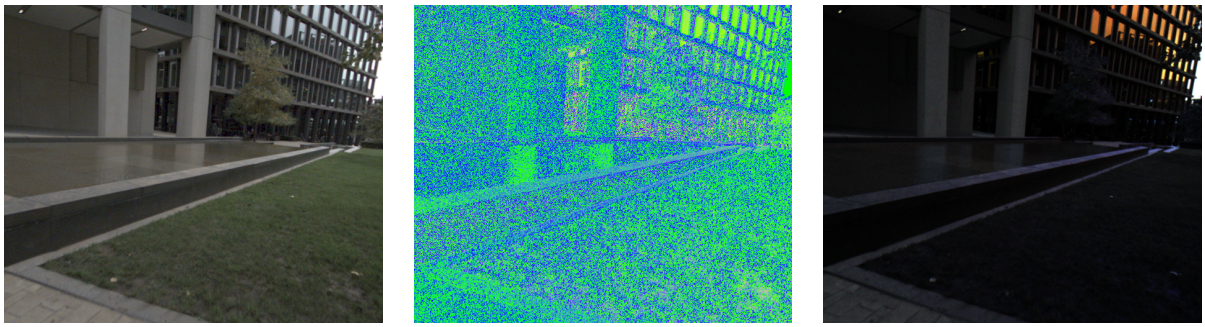
$$I_{pol} = (\rho, \cos(2\phi), \sin(2\phi)), \tag{5.1}$$

where ρ is the DoLP, and ϕ is the AoLP. Throughout this thesis, we use a color-polarization camera, and it is important to note that the DoLP and the AoLP are dependent on the

wavelength. Therefore, we will observe a different value for these parameters in each color channel. Nonetheless, for the visible spectrum, this difference is not too large, therefore, we can combine the signals to reduce the number of input channels. To keep data consistency, we will sample the DoLP and the AoLP as follows: firstly, we take the maximum DoLP from the three channels, and then we choose the AoLP that corresponds to that DoLP. Additionally, we have noted that when the intensity signal is low, some pixels exhibit a DoLP higher than 1, which has no physical meaning. These outlier values are obtained because the signal value is comparable to that of the noise signal. To avoid taking the measurements of such pixels, we set their DoLP value to zero before searching for the maximum value. By following this procedure, we can reduce the number of inputs, while not penalizing too much the performance.

The data representation of Eq. (5.1) is interesting because the obtained image contains 3 channels as for the RGB data, it represents the polarization information only, and it considers the periodicity of the AoLP. Therefore, no specific operation is required in the network model to correctly integrate this circular variable.

The second representation that we think may work is the division by polarization channels only, without computing the AoLP nor the DoLP. When using this data, the works from [12, 59] produce better results than those that make use of the polarization parameters. For the specularity removal network, Lei *et al.* [59] justify this fact by claiming that the operations in the raw intensities are linear whilst in any pre-processed image (for instance, converting the intensities into the RGB space) they are not. As a matter of completeness, we aim to test this second input encoding to evaluate the performance of the obtained network. Thereby, the network input in this case would consist in a multi-channeled image, in which each channel corresponds to the raw values of all the pixels with the same polarization filter orientation. An example of the two input encoding is shown in Fig. 5.5. As it can be noticed in Fig. 5.5, the first encoding method from Eq. (5.1) seems to have more noise than the second one. This is expected since the DoLP and the AoLP are computed based on the difference of two polarization channels, as explained in Sec. 2.2.1. Nonetheless, the former case contains purely the polarization data, and there is no overlapped information with the color channel. In the intensity concatenation case, the intensities have several values in common, and the



(a) (b) (c)

Figure 5.5: Possible network input encoding for the polarization encoder. (a) Original color image. (b) Polarization parameters image form Eq. (5.1). In this image, the absolute value of the sine and cosine functions are considered to avoid doing a color shift to represent negative values. (c) Color image obtained of concatenating the channels I_{0°, I_{45°, and I_{90°.

intensities contain texture information that is already included in the RGB input image. On the other hand, having less noise, and the hypothesis of linearity between the measurement and the incident light may be beneficial for the network to learn the polarization features. It is for this reason that both inputs encoding are potential candidates to fairly represent the polarization state of the light.

Concerning the network evaluation, the depth provided with both datasets will be used as a reference. It is important to note that, as mentioned in the documentation of the datasets, all the depths of value zero are either missing points, or invalid measurements, thus they will not be considered for evaluating the results. For the CroMo dataset, we decided to use the series of the environment ***station***, referenced as ***20201028-111403*** as the test set. This is because it comprises several objects of interest for the polarization that can be easily evaluated also by visual inspection: flat surfaces, glasses, water, and planar columns. All the metrics will be computed at every iteration of the training process, and the best model obtained after N epochs will be kept. From the dataset of Baltaxe *et al.* , we will use the provided split as test set.

In sum, we will test the two above-mentioned input encoding for the polarization data, and the results given by the network will be evaluated against the corresponding depth images. Only the valid depth pixels will be taken into account for the model evaluation, and the series ***station - 20201028-111403*** from the CroMo dataset, and the test split from Baltaxe *et al.* will be used as test set.

5.2.6 Loss

The loss function constrains the output signal to comply with certain laws during the problem-solving process. For instance, these laws can be geometric equations that bounds two points in an image, or a physical property of the variables we are manipulating. Nonetheless, this function should comply with certain mathematical properties. It should be differentiable, and it should have a minimum value (at least locally) to which the results provided by the network should converge. This minimum value will be reached if the set constraints are perfectly matched by the network outcome. In other words, this loss function should quantify the error between the estimation given by the network and the expected value of this estimation.

The shape of this loss depends on what the network is aimed to estimate. In the case considered for this thesis, we seek to estimate the distance of the objects present in a scene with respect to the camera coordinate frame by using a single color-polarization image. Concerning the color-only methods, depth estimation is carried out by using a loss with mainly two components: a smoothing term, and a Structural Similarity term. The smoothing component aims to produce smooth surfaces transitions, whilst keeping the objects edges. In general, first and second order prior smoothing functions [37, 117] are considered. The equations for first and second order smoothing terms are the following:

$$L^1_{smooth} = |\partial_x d^*_t| e^{-|\partial_x I_t|} + |\partial_y d^*_t| e^{-|\partial_y I_t|}, \tag{5.2}$$

$$L^2_{smooth} = |\partial^2_x d^*_t| e^{-|\partial^2_x I_t|} + |\partial^2_y d^*_t| e^{-|\partial^2_y I_t|}, \tag{5.3}$$

where $d^*_t = d_t/\bar{d}_t$ is the mean-normalized inverse depth, I_t is the intensity image at the current time t, and $\partial_i X$ and $\partial^2_i X$ are the first and second order partial derivatives of X with respect to the spatial axis i. Considering three neighbor pixels $\{a, b, c\}$ in the i axis direction (i.e., x-axis and y-axis directions), these functions are defined as:

$$\partial_i X = X\,(a) - X\,(b)\,, \tag{5.4}$$

$$\partial^2_i X = X\,(a) - 2X\,(b) + X\,(c)\,. \tag{5.5}$$

The Structural Similarity Index Measure (SSIM) loss aims to minimize the reprojection error. This component penalizes the artifacts at the object edges, and leads to better accuracy. This function is defined as [36]:

$$L_{repr} = \min_{t'} pe\left(I_t, I_{t'\to t}\right) \tag{5.6}$$

where $t' \to t$ indicates the pose transformation between the image taken at the time t' and the one taken at time t. The error $pe\left(I_1, I_2\right)$ is defined as:

$$pe\left(I_1, I_2\right) = \frac{\alpha}{2}\left[1 - \text{SSIM}\left(I_1, I_2\right)\right] + (1-\alpha)\left\|I_1 - I_2\right\|_1 \tag{5.7}$$

The factor α changes the relative weight between the structural dissimilarity, and the per pixel reprojection deviation. In the original Monodepthv2 paper, this value was set to $\alpha = 0.85$.

There exists some loss functions that are commonly found in the polarization imaging field when using deep learning algorithms. If the normal field is estimated, and the ground-truth normal field is known, the cosine similarity can be used [8, 60]. When estimating the depth in a self-supervised manner, the AoLP is compared with respect to the azimuth of the normal to the surface [10, 11, 124]. This last normal is computed from the gradient of the depth map estimated by the network.

In this thesis, we propose a contribution to the loss by using the polarization constraints, that does not depend on the reflection type. We propose a new loss term, that is similar to the one proposed by Smith *et al.* [97]. That paper proposes a comparison between the azimuth of the normal vector and AoLP that is reduced to test the colinearity of two 3D vectors. As explained in Sec. 4.2, the normal vector from any type of reflection contains a π-ambiguity with respect to the AoLP ϕ, since it is not possible to distinguish from the electric field orientation, if the normal azimuth is equal to either ϕ or to $\phi + \pi$. Nonetheless, if we project the electric field onto a plane perpendicular to the direction of propagation, the two possible vectors are on the same line, and the projection of the normal vector onto the same plane is colinear to both of them. Considering that the vectors are in 2D, we can

write the colinearity constraint for the diffuse reflection as follows:

$$\mathbf{n}(\mathbf{x}) \odot \begin{bmatrix} \cos(\phi(\mathbf{x})) \\ -\sin(\phi(\mathbf{x})) \\ 0 \end{bmatrix} = 0, \tag{5.8}$$

where $\mathbf{n}(\mathbf{x})$ is the normal vector to the surface at the point $\mathbf{x}$, $\phi(\mathbf{x})$ is the AoLP at the same point, and $\odot$ is the dot product operator. Similarly, we can write the colinearity constraint for the specular reflection as:

$$\mathbf{n}(\mathbf{x}) \odot \begin{bmatrix} \cos(\phi(\mathbf{x}) - \pi/2) \\ -\sin(\phi(\mathbf{x}) - \pi/2) \\ 0 \end{bmatrix} = 0, \tag{5.9}$$

Thus, to be able to use these equations, it is required to know if the incoming light is diffuse or specular, which is a challenging task for a generic scene. For the work in this thesis, we will create a mask of specular- and diffuse-dominant points following a similar approach as in [11]. Differently from that work, we will consider the two reflection types instead of only specular cases. Then, a point will be considered specular-dominant if its DoLP is higher than a threshold value β, and it will be considered diffuse-dominant in any other case. Then, our final polarization loss is:

$$L_{pola} = \gamma \left| M L_{spec} + (1 - M) L_{diff} \right| \tag{5.10}$$

where L_{diff} is the left-side of Eq. (5.8), and L_{spec} is the left-side of Eq. (5.9). The $|.|$ is the absolute value operation, and γ is a weight factor to balance this component loss contribution with respect to the other components, and M is a specular pixels mask, such that:

$$M = \begin{cases} 1 & \rho > \beta \\ 0 & a.o.c. \end{cases} \tag{5.11}$$

where β is a threshold value, and ρ is the DoLP. By using this mask, at each pixel, only one

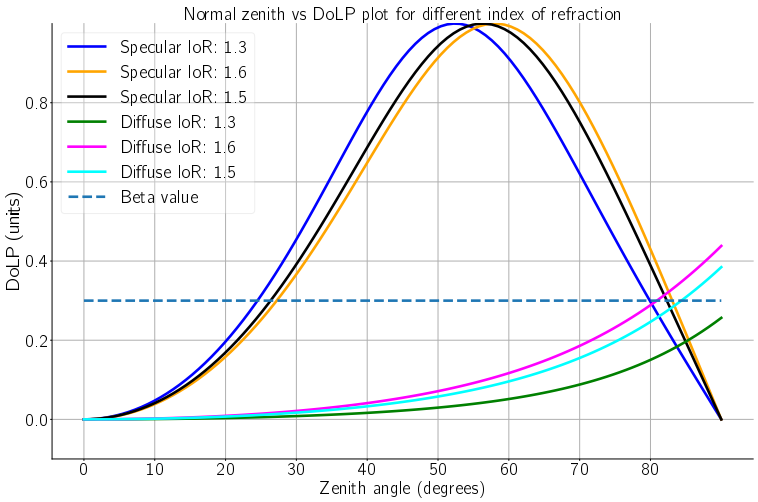


Figure 5.6: DoLP as a function of the zenith angle for different values of the Index of Refraction (IoR), and for the specular and diffuse reflections.

of the reflection models will contribute to the loss functions. The threshold value has been set to $\beta = 0.3$. This decision is not random, but taken based on the relationship between the zenith angle and the DoLP. This relationship has been shown and explained in Sec. 4.2, and in Fig. 5.6 we can see all the functions in a single plot, for different index of refraction values, and the threshold value.

If the reflection is classified as diffuse we will cover a range of zenith angles $[0, 80]$ degrees without error, and if it is classified as specular, this range will be $[25, 80]$ degrees. Therefore, the regions in which the classification may fail will be the range $[0, 25]$, and $[80, 90]$. Changing the threshold will also result in a change in the ranges where incorrect classifications may occur. In our future works, we aim to find a method to determine a better estimate of the diffuse and specular light proportions to further improve this loss function.

One may think that this classification is not necessary and that a good compromise and better solution would be to find the minimum value between the two loss components, similarly to what is done in [124]. However, performing such an operation may constraint the network to produce an incorrect result just to achieve the minimum loss value between the two loss components. Better results are obtained if classification is done before computing the loss. Additionally, this loss does not need to consider the periodicity of the angles, and with two cases, the four possibilities are covered.

In sum, the proposed system will have the following characteristics:

1. The architecture follows a middle-fusion model. One encoder will receive only the color information, and the other one will receive polarization based information.

2. The decoder architecture is the same as the one used in [36].

3. The encoder and the fusion module are not fixed, but the code will ensure a plug-and-play architecture that facilitates network architecture changes and the search of a combination of high-performance modules.

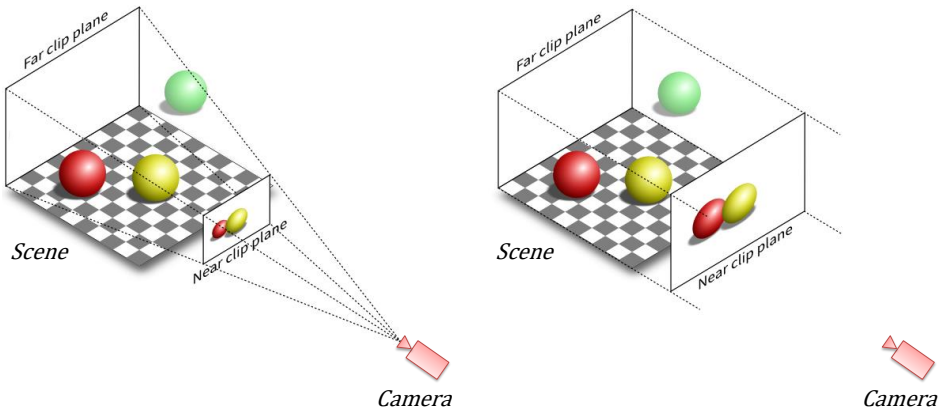
4. We will use on one hand the CroMo dataset [108], and we will use the polarization images from the dataset to train, and the depth dataset to quantify the results. On the other hand, we will test the network over the dataset from Baltaxe *et al.* with the train, validation, and test set split given by the authors.

5. The loss will have three components: first order smoothness L^1_{smooth}, reconstruction error L_{repr}, and our polarimetric loss L_{pola}. The three contributions will be added with corresponding weights. These weights will be the hyper-parameters of the network, that will be tuned with the aim of producing the best network optimization results.

5.2.7 Perspective or orthographic projection ?

In the classical SfP problem, most authors assume the orthographic projection hypothesis. This hypothesis assumes that all the light rays arrive parallel to the sensor surface. Nonetheless, most systems do not use an orthographic lens. In general, they use a perspective lens, and it is assumed that the scene projects in an orthographic way. Nonetheless, this happens only around the center of the sensor area. As we move to the borders of the sensor of a camera equipped with a perspective lens, this hypothesis is violated. A sketch of the difference between the orthographic and the perspective projection is given in Fig. 5.7.

For the compensation of this effect in data-driven algorithms, Lei *et al.* [60] propose a solution to the usage of perspective cameras, that consists in including the viewing encoding as network input. This new data is supposed to be assimilated by the network and to let it "understand" from which direction the light is coming. Even though it produces an



(a) (b)

Figure 5.7: Sketch of the orthographic and the perspective projection principle. (a) Perspective projection. (b) Orthographic projection.

improvement in the network accuracy, there is no real proof that the network is integrating the viewing encoding.

More recently, Pistellato and Bergamasco [82] introduced a geometrical model that explains how the light rays are projected over the DoFP sensitive area, and the effective filter axis seen by the polarized light. Briefly, starting from the camera coordinate frame of a perspective camera, and by using the intrinsics parameters of the camera obtained through geometric camera calibration, the authors construct a coordinate frame whose Z axis matches the direction of propagation of the light. Since this direction is different for each pixel, there exists a rotation matrix $\mathbf{R}_i$ that transforms a point or a vector from the Local Coordinate Frame (LCF) i to the global camera coordinate system.

The rotation matrix $\mathbf{R}_i$ serves to convert the filter axis of a pixel into an effective filter whose axis is perpendicular to the direction of propagation of the light. This will convert each of the polarization states into a polarization state in the LCF, allowing to work as if the light is under the orthographic projection. Once the calculations done, the normal vector obtained from the polarization should be moved to the global camera coordinate system to obtain the final solution. This is done though the matrix $\mathbf{R}_i^T$. A sketch of the coordinate system transformation is shown in Fig. 5.8.

To correctly make use of the polarization measurements, we must consider the perspective projection of the light onto the sensor. By doing so, we will ease the network assimilation

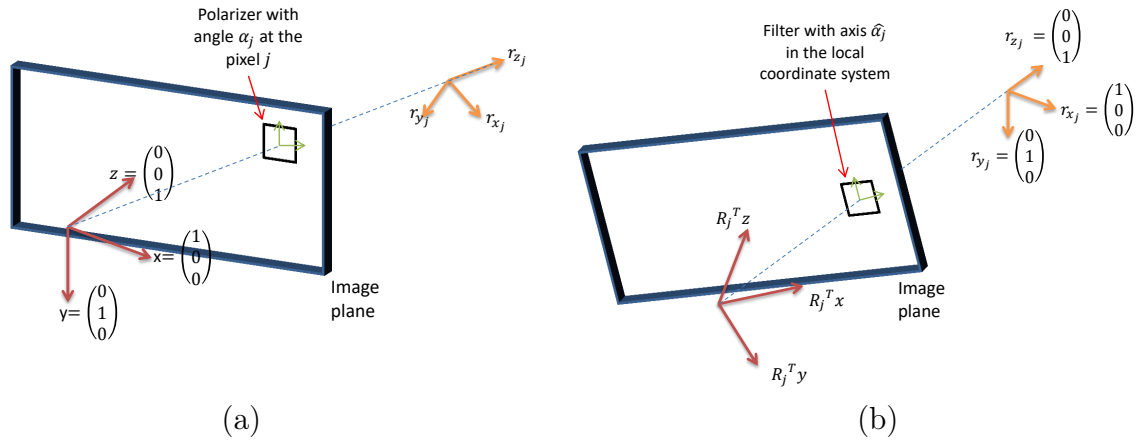


Figure 5.8: Sketch that explains how the LCF transformation works. (a) Camera Coordinate Frame for a perspective camera. In this system, the light arrives with a certain angle to the surface of the filter, thus the filter orientation with respect to the Camera Coordinate system does not match the effective angle by which the light is filtered. (b) The coordinate system is rotated by a rotation matrix R_j such that the filter axis is such that it matches the effective angle of the filter. This rotation matrix depends on the jth pixel position. In this case, the light propagation direction matches the z-axis of the local coordinate system, so we are under the orthographic projection hypothesis.

of the polarization cues and constraints. This will be done by using the model developed by [82], since it has been well proven, and it does not modify the network architecture. Indeed, the proposed model is applied before entering to the network, and the backward pass is applied after computing the final normal vector. In our case, the model application is done as follows:

1. Compute the pixel matrices $\mathbf{A}_i$, and the rotation matrix $\mathbf{R}_i$ as explained in [82] per pixel. Since the calibration parameters are unique, this is done only once.

2. For each input image, correct the intensity measurements by doing:

$$\mathbf{I}'_i = \mathbf{A}_{ideal}\mathbf{A}_i^{+}\mathbf{I}_i, \tag{5.12}$$

where $\mathbf{I}_i$ are the stacked measurements from the camera for the ith pixel, $\mathbf{A}_i^{+}$ is the pseudo-inverse of the pixel matrix estimated in the Local Coordinate Frame, $\mathbf{A}_{ideal}$ is the pixel matrix considering that the sensor is ideal, and $\mathbf{I}'_i$ are the corrected measurements in the local coordinate frame. The matrix $\mathbf{A}_i$ is estimated using Eq. (2.8) from Sec. 2.1.3, where the different filter orientations are the ones estimated as in Eq. (13) of the paper [82].

3. Estimate the Stokes vector in the local coordinate frame of each pixel by considering the camera as ideal, with the corrected measurements $\mathbf{I}'_i$.

4. Compute the input encoding to the network with the Stokes vector of the previous step.

5. In the loss function, compute the normal vector from the depth. Since this depth is given in the camera coordinate frame, the normals are in the camera coordinate frame too.

6. Use the rotation matrix $\mathbf{R}_i$ to rotate the previously obtained normal vector into the local coordinate frame.

7. Compute the loss of Eq. (5.10), between the AoLP and the normal vector previously obtained, which are both in the corresponding local coordinate frame.

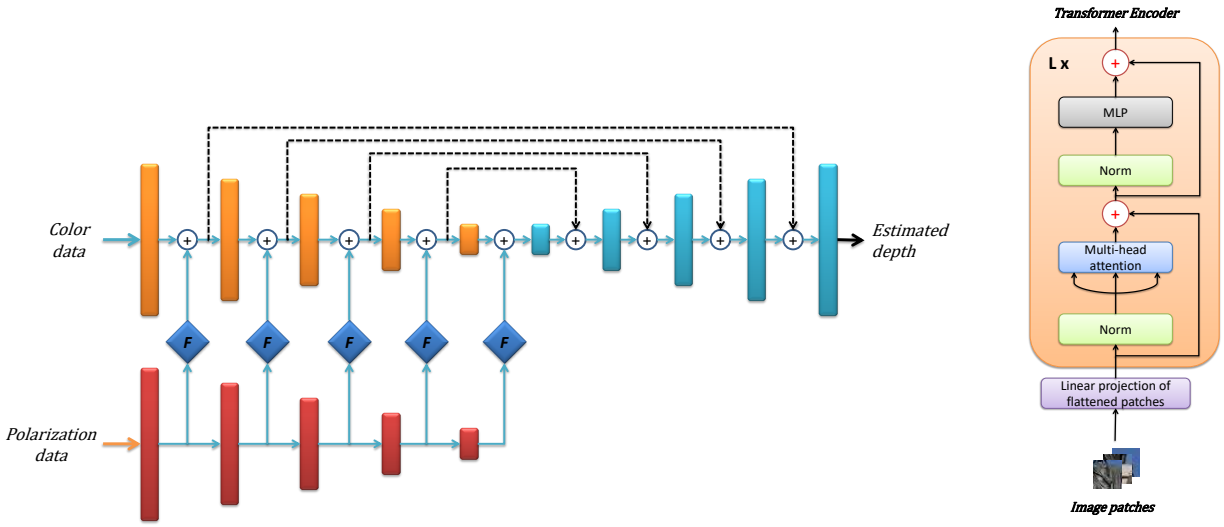
To accelerate the training time, we have preprocessed the CroMo dataset with the steps in Items 1 and 2.

5.3 Experiments

In this section, we show the results obtained with two already existing depth estimation methods. One of them is the baseline Monodepthv2 [36], and the second one is the work from Blanchon *et al.* [11], in which they have added a loss contribution to the Monodepthv2 code to consider the polarization state of the light while keeping the base network architecture. Differently from our case, they work with a monochrome camera, and they only consider specular reflections. Furthermore, we include the results of our network, developed based on the constraints given in the previous section.

5.3.1 Implementation details

In what follows, we will detail the network architecture developed to do monocular depth estimation using color-polarization images. An overview of the entire architecture is shown in Fig. 5.9. This network consists of two Convolution-based encoders to do feature extraction at



(a) (b)

Figure 5.9: (a) Proposed network architecture. This network is composed of two encoders (one per modality) based on CNN. The fusion module F takes the feature map from the polarization branch, and it modifies it such that the areas of interest will have larger values than the regions that are not of interest. Then, the output of this fusion module is added to the output of the corresponding feature map of the color branch. This addition is entered to the following convolutional layer. (b) The fusion module F is a transformer, that will apply a self-attention mechanism to the polarization data. This way, the areas of interest will have large values, and the areas with no valuable value will have low values.

different hierarchical levels. We have two ResNet-18 encoders as in the original Monodepthv2: one for the color data, and one for the polarization data. The polarization feature maps are passed through a fusion module to find the areas with the most valuable features. The output of this module is added to the corresponding feature map of the color branch, and this result is given to the next convolutional layer of the RGB encoder. This architecture preserves the performance of the color-only depth estimation network. Indeed, if there is no important data from the polarization branch, the original color feature map is left untouched. Only when the polarization data is valuable, the color branch will benefit from it. The fusion module follows the structure of a Visual transformer [28] encoder. This block will apply the self-attention mechanism to the polarization branch, weighting the different regions of the input feature map depending on the relevance of that data to the final objective task. By doing so, we expect to lower the values of the feature map in the regions where there is no valuable polarization information, and to provide important clues where the color information is not enough to solve for the distance. For each fusion module, we have chosen a patch size of 16 as in the original model, a set of $L = 12$ consecutive encoder blocks, 12 attention heads, and an embed dimension of 768.

To the original Visual Transformer, we have added a 2D convolution operation to adapt the number of channels to the required value to combine this feature map with the corresponding color map. Furthermore, to be flexible to any input image size, we have added a padding operator that is applied to the transformer block input. This padding will add enough zeros in the last columns and rows to make the image divisible by the selected patch size. Then, the added columns and rows are removed before providing the final output image.

In our architecture, there is an independent weighting module for each hierarchical level of the encoder. This is required since each feature map has a different size, thus a single transformer module cannot adapt to all of them.

As input encoding, we use the equation Eq. (5.1), since it is the one that correctly separates the intensity information from the polarization information. Finally, the hyperparameter for the loss in Eq. (5.10) has been set to $\gamma = 0.001$.

5.3.2 Evaluation

In this section we evaluate the performance of the baseline algorithms, and the proposed one. For the baseline methods, we include the results obtained with the publicly available weights, and with the weights trained with the images from the CroMo and Baltaxe *et al.* datasets. Regarding the code from [11], monochrome polarization images has to be considered as inputs. To obtain these images, we do the average of the color channels to obtain an equivalent polarization image (I_{0°, I_{45°, I_{90°, I_{135°) in gray levels. The models have been evaluated with the test set mentioned above (***station - 20201028-111403***, and the corresponding test set from Baltaxe *et al.*). The chosen evaluation metrics are the ones commonly adopted for the depth estimation algorithms [17]. Given an estimated depth image $\hat{d}$, the corresponding ground-truth image d, and considering we have N pixels from which we are going to evaluate the performance of the depth estimator, the depth metrics are defined as follows:

- Absolute relative error (ARE):

$$ARE = \frac{1}{N}\sum_{i=1}^{N}\frac{|d_i - \hat{d}_i|}{d_i} \tag{5.13}$$

- Square Relative Error (SRE):

$$SRE = \frac{1}{N}\sum_{i=1}^{N}\frac{\left(d_i - \hat{d}_i\right)^2}{d_i} \tag{5.14}$$

- Linear Root Mean Square Error (RMSE):

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(d_i - \hat{d}_i\right)^2} \tag{5.15}$$

- Log Scale Invariant Mean Square Error:

$$MSE_{log} = \frac{1}{N}\sum_{i=1}^{N}\left(\log\left(\hat{d}_i\right) - \log\left(d_i\right) + \alpha\left(\hat{d}_i, d_i\right)\right)^2 \tag{5.16}$$

  where $\alpha\left(\hat{d}_i, d_i\right)$ is in charge of the scale alignment. It is defined as $\frac{1}{N}\sum_{i=1}^{N}\log\left(\hat{d}_i\right) - \log\left(d_i\right)$.

- Accuracy under threshold:

$$\delta = \max\left(\frac{d_i}{\hat{d}_i}, \frac{\hat{d}_i}{d_i}\right) < \tau \tag{5.17}$$

  where τ is a predefined threshold. In general, we find that $\delta < 1.25$, $\delta < 1.25^2$. and $\delta < 1.25^3$ are common threshold values.

The default training configuration from Monodepthv2 is: Adam optimizer, input image size 640×192, learning rate of 10^{-4}, and disparity smoothness weight $1e-3$. We have trained the models over 50 epochs, with a batch size of 16. All the tested code has been

developed in Python programming language, with PyTorch as deep learning library. For fair comparisons, we have added additional code to fix the initial state of the training process at the same point at every run. This enables reproductibility through successive training runs. We have trained the models on a computing server running CentOS 7 with a single NVIDIA A100 40Gb GPU card. Our model inference time is 42.4 ms, whilst for Monodepthv2 it is 6.7 ms, and for the P2D model, it is 11.2 ms. These times have been computed as the median inference time over the test set. The difference in inference time between Monodepthv2 and P2D is due to the difference in the encoder architectures which are a ResNet-18, and a ResNet-50, respectively. In our case, we follow a middle fusion architecture with two ResNet-18 encoders, plus a transformer module to balance the polarization contribution. Despite the increased computational load compared to baseline methods, we maintain a frame rate of over 20 fps. We leave as future works the evaluation of a late fusion architecture. In Fig. 5.10 we show some results of the models for a sample image, and the ground-truth depth image.

In Tab. 5.1 we summarize the metrics results of the different trained models. In this table, we have included the results of the two models with the corresponding weights given by the authors, and we also did a full training with the CroMo dataset. The two evaluated models have not been chosen arbitrarily but with a clear objective. The model Monodepthv2 is a network that does depth estimation based on textures only. The code from P2D works similarly, but is based mainly on the polarization information. Even though the total light intensity data is given as part of the inputs, the color information is not present. Therefore, the two baseline networks produce alike outputs, based on only one of the modalities. Our model takes the best of two worlds, and it extracts polarization features and color features separately, and as shown in Tab. 5.1, using polarization and color information simultaneously outperforms both baseline algorithms.

In Fig. 5.10, we have marked some areas of interest with colored rectangles. These areas (insulators, monochrome, planar surfaces, highly reflective) are of interest since they possess important polarization cues to disambiguate the results when there is a lack of texture in the color images. This information can be seen in the DoLP and the AoLP shown in Fig. 5.12. In these images, the windows, the water, and the planar surfaces present a color gradient in the AoLP value, and a DoLP value that changes based on the material, which is not the case

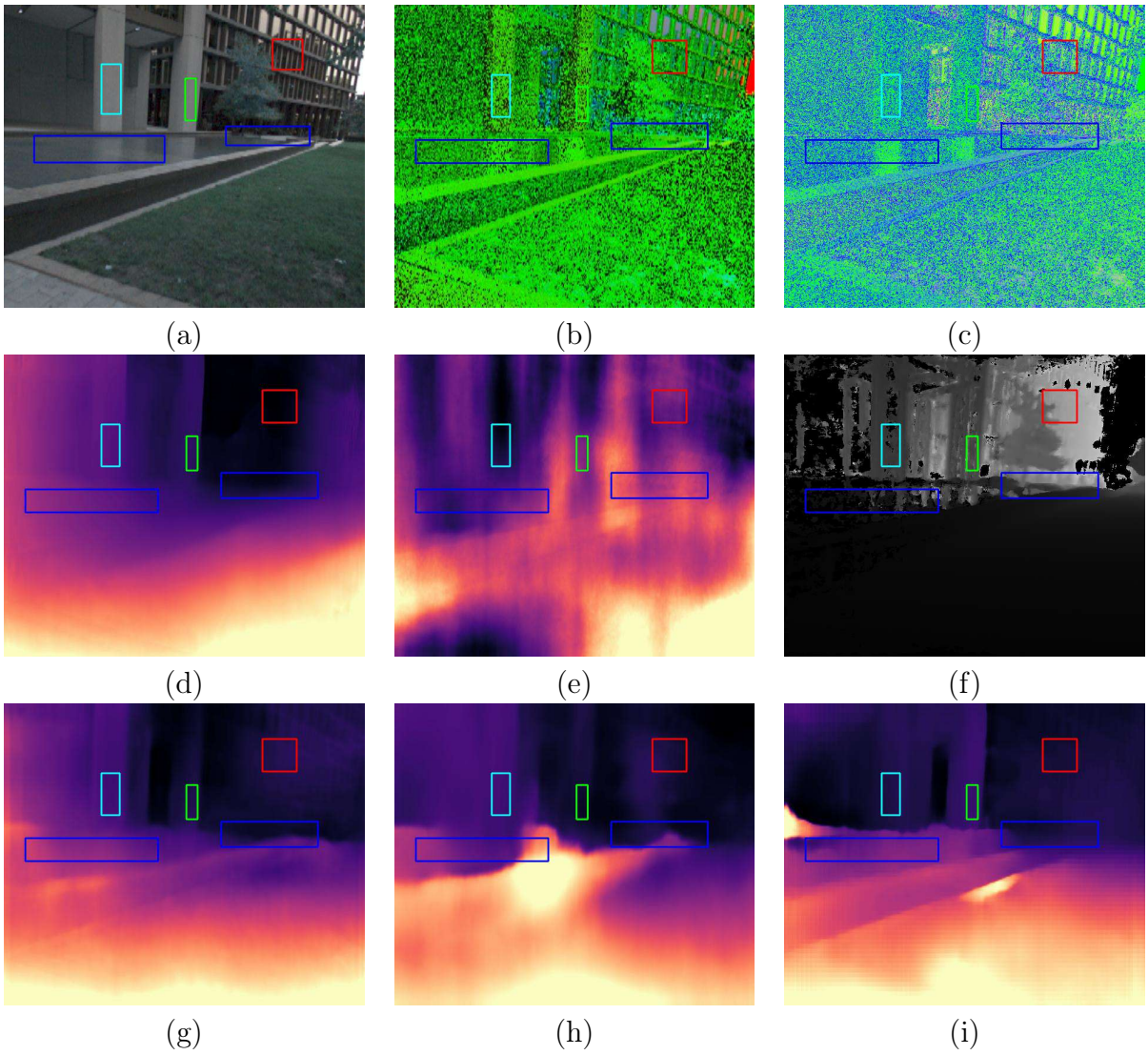
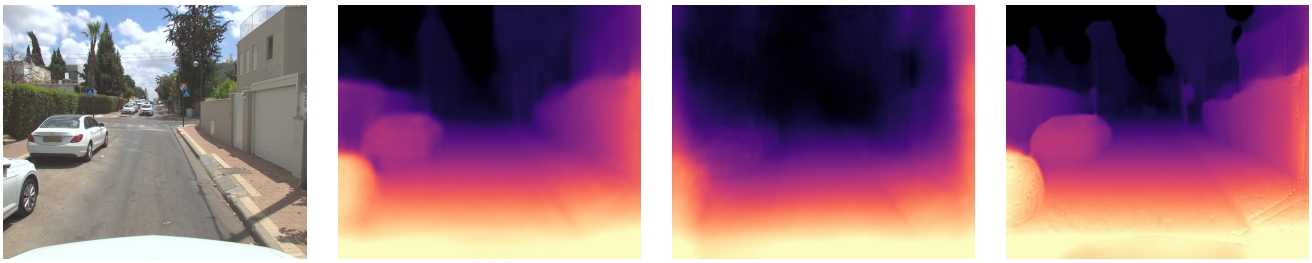


Figure 5.10: Qualitative results of the trained networks with the CroMo dataset. The input image corresponds to one sample of the test set. The drawn rectangles are just to bring the attention to the points of interest where polarization information generally provides valuable cues (water, flat single color surfaces, glasses). (a) Color image used as input for the Monodepth2 and the color branch of the proposed network. (b) Input encoding for the P2D algorithm. (c) Input encoding for the polarization branch of the proposed algorithm. To be able to correctly visualize the image, the absolute value of the cosine and sine function have been used. (d) Results obtained with the pre-trained weights of the Monodepthv2, provided by the authors [36]. (e) Results with the pre-trained weights of the P2D network [11], provided by the authors. (f) Ground-truth depth image. (g) Results obtained obtained after training the Monodepth2 network with the CroMo dataset. (h) Results obtained obtained after training the P2D network with the CroMo dataset. (i) Results obtained obtained after training the proposed method network with the CroMo dataset.

for the color images. None of the tested networks perform well in those regions, even if one of them receives as input the polarization state of the light. In Fig. 5.11 we show the results obtained with a sample image from the dataset of Baltaxe *et al.* . Particularly in this image, we can see how the cars windshields, and the far field objects are better reconstructed when the polarization data is correctly integrated with the texture information, in contrast to the results obtained with Monodepthv2.

Dataset	Model name	Trained ?	ARE	SRE	RMSE	MSE_{log}	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
CroMo	Monodepthv2	No	0.418	20.393	51.155	0.591	35.04 %	61.18 %	69.81 %
		Yes	**0.302**	**12.105**	**36.928**	**0.302**	**51.43 %**	**70.78 %**	**83.80 %**
	P2D	No	0.928	30.667	54.702	0.976	17.46 %	34.89 %	51.40 %
		Yes	0.358	13.650	38.835	0.364	37.67 %	65.71 %	79.49 %
	Ours	Yes	**0.274**	**10.845**	**34.463**	**0.254**	**57.19 %**	**73.61 %**	**86.23 %**
Baltaxe *et al.* [9]	Monodepthv2	No	0.150	3.003	16.965	0.068	77.50 %	92.42 %	97.49 %
		Yes	**0.079**	**1.479**	**11.435**	**0.022**	**91.73 %**	**97.60 %**	**99.30 %**
	P2D	No	0.449	13.099	27.727	0.293	34.39 %	62.73 %	80.72 %
		Yes	0.235	6.074	25.812	0.167	58.28 %	83.07 %	92.08 %
	Ours	Yes	**0.072**	**1.422**	**11.513**	**0.021**	**93.08 %**	**97.73 %**	**99.16 %**

Table 5.1: Quantitative results of the evaluated models (Monodepthv2 [36], P2D [11], and proposed) over two different datasets. When the model has not been trained, the weights given by the authors have been used. When trained, we have used the configuration mentioned in the original corresponding paper. Numbers in bold black indicate the best result, and numbers in bold blue means the second-best result.

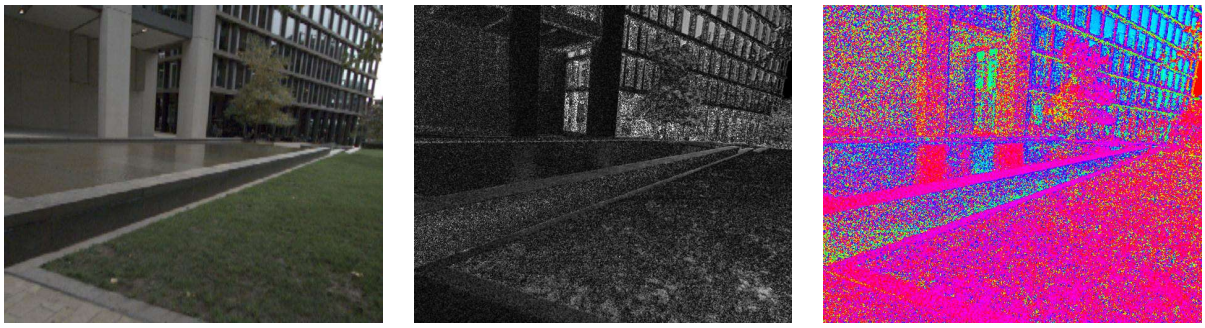


(a) (b) (c) (d)

Figure 5.11: Qualitative evaluation of the re-trained models over in [9]. (a) Color image. (b) Output from the Monodepthv2. (c) P2D algorithm; and (d) the proposed algorithm.

From both, qualitative and quantitative evaluation, we can see that the pretrained networks do not outperform the trained networks. In general, a data-driven algorithm is strongly dependent on the data with which it has been trained. In the case of images, changing the camera lens, the sensor dimensions, sensor noise, the pixel sensitivity, and the type of environment used during training will introduce visual effects that the deep learning network is not capable to understand. This is why a training process is required from either the pretrained weights, either from scratch. Since for our model we do not have pretrained weights for the polarization encoder, we also trained all the models from scratch with the CroMo dataset to make a fair comparison of their performance.

From the trained versions, we can see that the Monodepth2 model provides a good performance considering that it is only using the color information of the light. In Fig. 5.10 we can distinguish several objects as the columns behind the pool, part of the building, and slightly the tree. Nonetheless, there are several inconsistencies, such as the borders of the swimming pool are not sharp, the roof below the first stage disappears behind the right column, and the end of the pool close to the tree vanishes. The P2D approach does not produce satisfactory results either, which can be caused by several aspects. Firstly, the color



(a) (b) (c)

Figure 5.12: Polarization parameters of the sample image used to test the networks. (a) Color image. (b) DoLP as a gray-scale image. A white color means a Degree of Linear Polarization of 100 %, and a black color means a DoLP of 0%. (c) AoLP colored with the HSV palette.

information is not provided as input, but the gray-scale intensity. This reduces the amount of texture information that can be extracted from the image. The polarization information is given at the input, without any processing. Particularly, the periodicity of the AoLP is not considered either by the network operations or by the input encoding. As a consequence, there will be several strong variations in the color when the measured angle is close to zero. Our proposed method improves several of these aspects. The columns and the roof of the first stage are present in the depth image, and the estimation of the depth for the pool is geometrically consistent. Additionally, a larger area of the building is reconstructed, and the transitions between objects are sharp.

From the quantitative evaluation in Tab. 5.1, we also observe that the proposed method outperforms the other models for all the considered metrics.

5.3.3 Ablation study

In order to validate our methodology, we conducted several ablation studies. Our method contains the following improvements:

- A loss that encapsulates the polarization, and both reflection types (diffuse and specular),
- Two independent encoders, bounded by a transformer-based weighting module,
- A perspective to orthographic reprojection stage.

Model name	ARE	SRE	RMSE	MSE_{log}	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
M.1	0.292	11.339	36.613	0.249	53.92 %	73.54 %	85.39 %
M.2	**0.290**	11.876	37.882	0.275	**55.27 %**	**72.76 %**	84.96 %
M.3	1.064	36.026	59.790	1.266	14.06 %	26.09 %	37.50 %
M.4	0.440	18.883	45.793	0.464	40.78 %	61.18 %	73.57 %
M.5	0.354	16.525	43.394	0.469	46.09 %	65.27 %	75.66 %
M.6	0.300	**10.746**	**36.319**	**0.270**	51.07 %	71.66 %	**85.72 %**
M.7	0.287	9.922	35.761	0.266	53.01 %	71.58 %	85.11 %
M.8	0.307	11.550	36.748	0.319	49.99 %	70.03 %	82.67 %
Monodepthv2	0.302	12.105	36.928	0.302	51.43 %	70.78 %	83.80 %
Proposed	**0.274**	**10.845**	**34.463**	**0.254**	**57.19 %**	**73.61 %**	**86.23 %**

Table 5.2: Ablation study results. Numbers in bold black indicate the best result, and numbers in bold blue means the second-best result.

Thus, we verify the influence of each part independently. The results of these tests are included in Tab. 5.2. Visualizations for different scenes are shown in Figs. 5.13 and 5.14.

Architecture modifications: The first test corresponds to the model M.1, in which the transformer block has been replaced by a direct connection. In other words, the output of the transformer block has been set equal to the input to it. A consequence of removing this block is that the self-attention mechanism is removed, which is in charge of finding the regions of interest where the polarization data can contribute to the final results. Even though the quantitative results are good, the visual results present several inconsistencies when using this model.

Loss contribution: For these tests, we keep the proposed network architecture, and then we remove our loss contribution to see how the optimization process changes during the model training. The results corresponding to this test are the ones of model M.2. This test confirms that our model complies with our main premise: adding the polarization information should preserve the performance of the baseline, color-based network. Moreover, we see that all the metrics provide better results when the polarization data is added than when only the color data is used. In our case, the polarization data provides geometrical constraints that are learned by the model. This data is processed by the Convolutional encoder to extract features, but those features are filtered by the transformer encoder. This last block will weight the features, and if they do not provide valuable information, they will be assigned

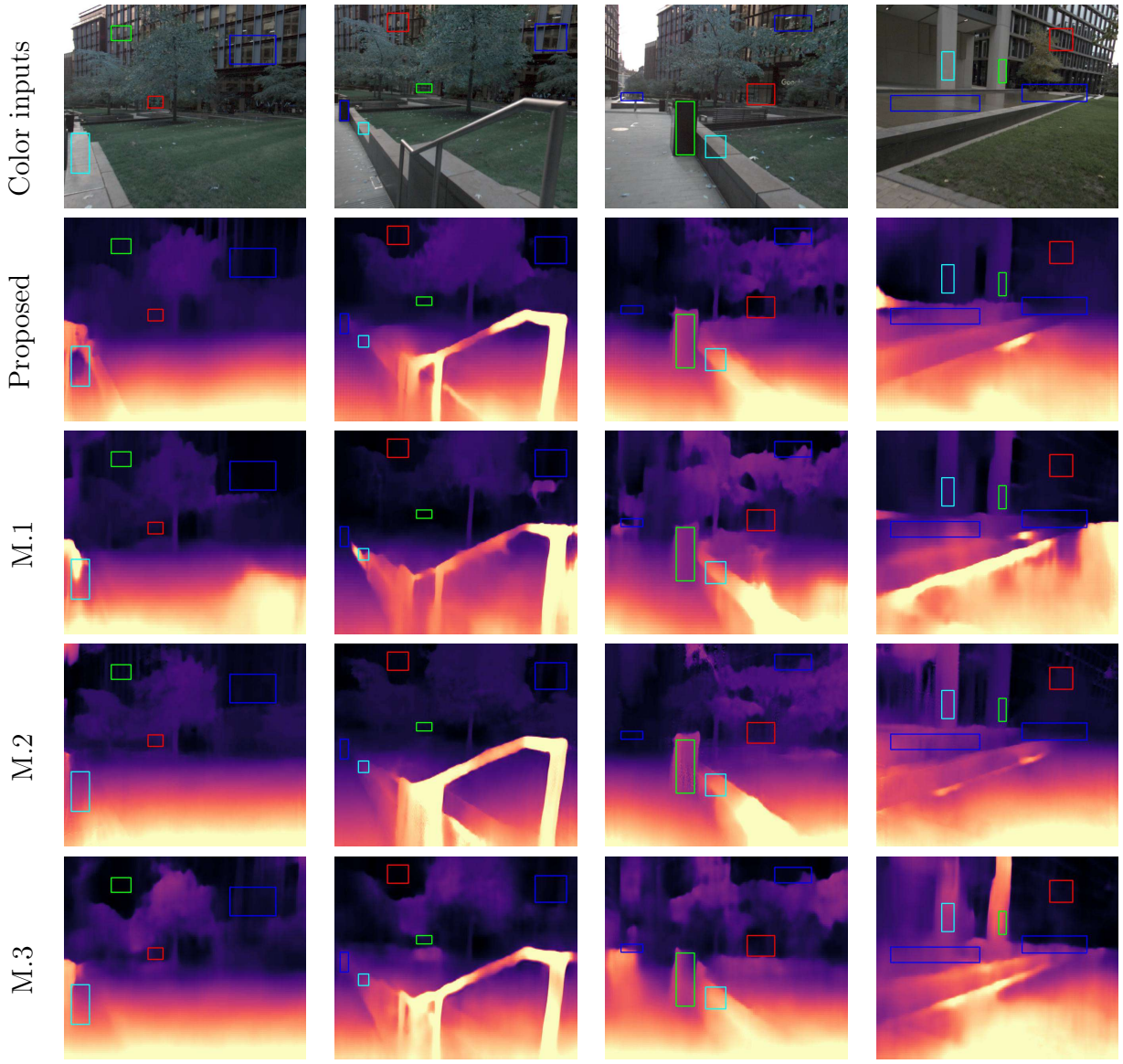


Figure 5.13: Results obtained with the different models of the ablation study. Part 1

a value close to zero. Next, the filtered features are mixed to the color data. Therefore, the only effect of including the polarization data is to improve the original results.

Perspective consideration: Our model also considers the perspective projection produced by the camera lens. Thus, for the model M.3 we have removed the preprocessing step that converts the filter orientations to the local coordinate frame, and from the loss, we have removed the step in which the normal vectors are moved to this coordinate system to make the comparison between the azimuth angle, and the AoLP. This model perfectly shows the effect of not considering the perspective projection effect in the polarization data. Even

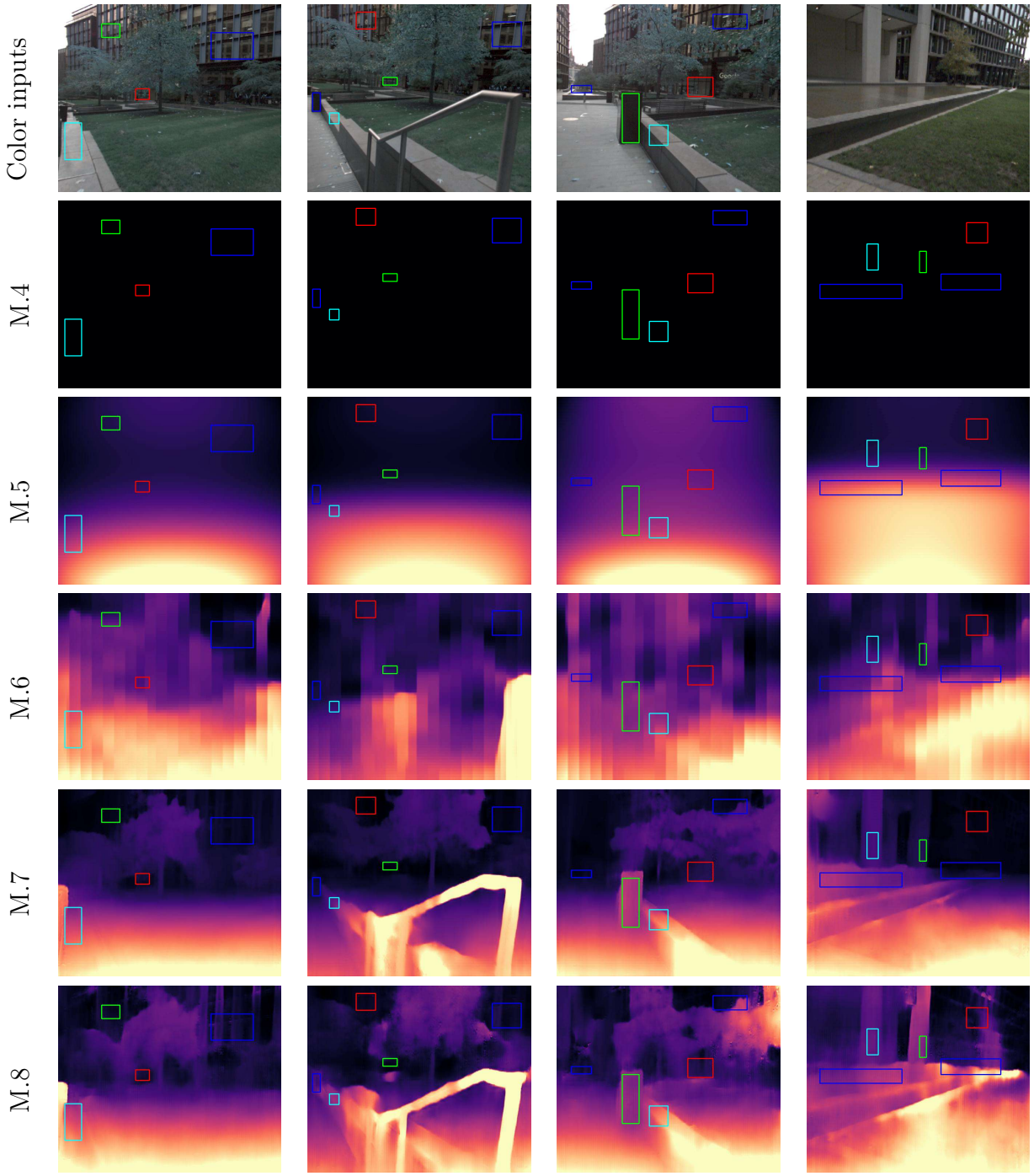


Figure 5.14: Results obtained with the different models of the ablation study. Part 2

though the model architecture is exactly the same as the proposed one, the outcomes are completely different. The metrics are degraded with respect to the full model, and we can see inconsistencies in the output images. Notably, the relative value between far and close objects is deteriorated when the perspective projection is not considered.

Loss contribution weight: We have observed that the loss contribution can have a value up to four times the Structural similarity loss value. This might bias the network results to tend to ignore the texture-based information all the time. Thereby, we have made tests for different γ values, in which the model M.4 corresponds to a value of $\gamma = 1$, M.5 corresponds to a value of $\gamma = 0.1$, M.6 corresponds to a value of $\gamma = 0.01$, and the proposed method has a value of $\gamma = 0.001$. The models obtained with these different values of the hyper-parameter γ confirm our hypothesis. When $\gamma = 1$, the polarization contribution mainly guides the training process, and the result is a network that outcomes dark images. This is an indication that the model has found a local minima far away from the convergence point. An explanation to this effect might be that the images of the polarization parameters are noisy as shown in Fig. 5.12. These images include more noise than the intensity images since the AoLP and DoLP images are obtained after doing the difference of two polarization channels. This operation is equivalent to computing the gradient of the intensities, an operation that tends to increase the level of noise with respect to the original images. Thus, having a loss function based on these observations only will not guide the model toward a general solution. Therefore, the polarization measurements should be incorporated in the loss to improve the results, but other terms have to be included to compensate the fact that the inputs have a low Signal-to-Noise ratio. It is for this reason that as we reduce the strength of the polarization contribution, the performance metrics of the network start increasing. From the results shown in Fig. 5.14, we see that we start with dark images when $\gamma = 1$, and we start having some little gradients when $\gamma = 0.1$. Next, when $\gamma = 0.01$, the images start estimating a depth similarly to the final model, and when the $\gamma = 0.001$ the proposed model is obtained.

Reflection type: We have tested the effects over our model if only one reflection type is considered. To do so, we use the function M as the diffuse- and specular-dominant separation mask, and remove one of the two loss contributions. In other words, in Eq. (5.10) we considered two cases: one in which we removed the term $(1 - M)\, L_{diff}$, and the other one where it is the term $M L_{spec}$ which is removed. These models are named M.7 and M.8 respectively. From the quantitative point of view, neither of the models has been able to

perform as well as the proposed method. This result is expected since the added information is less when considering one reflection than when considering both of them. Considering the qualitative results, we see that the results obtained with the specular part only are worse than for the diffuse light, and that the diffuse results are similar to those obtained with the proposed method. This is because the diffuse component considers the regions with a DoLP between 0 and 0.3, which correspond to most of the regions of the scene. We can also see that far away regions are less described by this model than by the proposed one. On the contrary, when keeping only the specular reflection data, the number of pixels that reflect specular light is less than the number that reflect diffuse light. This is due to only highly reflective regions, such as the water and the glasses, will reflect highly polarized light. Furthermore, this polarization level will happen only when the viewing angle is close to the Brewster angle, which is not always the case. Therefore, the polarization contribution is less frequent in these cases, thus producing results closer to the ones obtained with color-only images.

5.4 Conclusions

In this chapter, we have presented a methodological path to incorporate the polarization state of the light in a deep learning network. Particularly, we aim to improve an already existing algorithm that estimates the scene depth based only on textures, by integrating the polarization information in the pipeline. Some works have tried to address this problematic, but their results are not optimal, and they do not show the estimation improvements in the areas in which color-based methods generally fail. Throughout this chapter we have analyzed the different network options, and tools already available to create our own method. We have detailed all the required functions, and we have shown the results obtained with two data-driven algorithms to establish our baseline, and to stress the weak points that we want to address with our model. Then, we have chosen a configuration of a deep learning network to do depth estimation by jointly using color and polarization information. We have evaluated all the models quantitatively and qualitatively, and we have shown that our method surpasses the baselines. Our model does not degrade the performance of the baseline

model which makes use of the color information only. Our loss considers both, diffuse and specular reflections, and the perspective reflection over conventional cameras. Taking into account these considerations is essential to produce outperforming polarization algorithms. As stated before, the baseline algorithms are not able to correctly reconstruct flat surfaces such as water, glass, or highly reflective surfaces. Our algorithm improves these results by achieving a perfect balance between the contributions of the two modalities: color and polarization. Indeed, through the transformer encoder, our network can dynamically decide where is the most useful polarization information to correctly reconstruct the scene depth, and since this information is simply added to the color encoder, we do not degrade the performance of the original color-based network.

Chapter 6

Conclusion

Robots are more and more present in our daily life. This is mainly because the target markets have grown in the last years, and because the price of the components with which they are made have become more affordable. Vacuum cleaners, lawn mowers, humanoids, mobile telepresence, autonomous driving cars, aerial and ground robots for video recording or scene analysis are the ones that almost everybody has seen or heard. Thanks to the increased computing resources available today compared to the past, these machines now incorporate more complex algorithms and often operate with greater independence from humans. To accomplish their task, the robots need to sense their surroundings in different manners and fuse the data coming from sensors with the aim of accomplishing their mission in a safe and accurate way. Of the different available sensors, vision is the one that provides the richest information about the environment, but many of those algorithms limit their usage to the visible spectrum since it is the same range of frequencies that humans can see. Thereby, the obtained images are easier to interpret for us than non-visible modalities such as infrared or polarization. Nonetheless, using other modalities complements the information that cannot be disambiguated by using only visible light. Particularly, with the polarization state, the light is completely described since we consider its intensity, its color, and the way it moves as it travels.

In this thesis, our main objective is to show how the polarization data can be used to do 3D scene reconstruction, and how we can modify already existing algorithms that make use of visible colors only to outperform their results by using the polarization data. Our

first contribution in this sense is the development of a calibration algorithm able to fit a complete pixel model to the measured data. By doing so, we correct the distortions in the measurements due to parameter dispersion, manufacturing errors, and lens deformations. Differently from other algorithms, our aim is to facilitate the usage of this modality with the DoFP sensor. It is for this reason that our method simplifies the acquisition setup by estimating the light parameters during the calibration procedure, and by being more flexible with the acquisition conditions while keeping the system accuracy. With our proposed method, the required time to do the polarimetric camera calibration is reduced.

To show the effects of our calibration algorithm over a concrete application, we have developed a Shape from Polarization pipeline based on the polarization state physics. To the best of our knowledge, for the very first time, we have done a complete description of the SfP algorithm, by considering the effect of the different types of reflections and materials. We have provided all the required formulas based on Fresnel equations, and we have included their inversed versions. Furthermore, we have shown that jointly the polarization state and the calibration enable more accurate normal estimation at different orientations of a plane surface than an uncalibrated setup. With this application, we have demonstrated the power as well as the limitations of the geometrical constraints set by the electromagnetic waves theory. When an application makes use of specific materials from which we know an approximate shape, and index of refraction, the normal vector can be estimated with a single picture. When not too much information about the object is known, assumptions need to be made (for instance, reflection type, incoming light polarization state, or concavity). In the simple but effective method developed, we have shown that, even if the sensor is of good quality, the calibration algorithm is able to introduce an improvement of more than 12% when running the SfP application, which is not negligible. Thus, the calibration pipeline is a requirement for this modality, and sensing methodology.

Finally, we have a possible Deep Learning network that can be used to estimate the distance to the objects with respect to the camera coordinate frame. This application is of importance since it allows us to estimate a possible danger close to our robot. Therefore, accuracy is vital. Due to the type of geometrical constraints and its independence on the light intensity, polarization is a perfect candidate to improve the quality of texture-based

networks that perform this task. In the Chapter 5 of this thesis, we have analyzed the pros and cons of a well-adopted deep learning network to do self-supervised, monocular depth estimation: Monodepthv2. We have analyzed its components, its loss function, and we have adapted a polarization contribution that can improve the performance of the baseline network. The proposed loss term is independent of the material, and the reflection type of the objects present in the scene. We have carefully built an architecture that does not degrade the original network performance, and it improves the reconstruction quality where the color information is not enough. Differently from other approaches, we have introduced a geometrical model to consider the effects of the perspective projection over the polarization measurements. The contribution of this consideration has been shown in our ablation studies in which we confirm that this correction is mandatory to obtain outstanding results when using the polarization information. We have shown results of two baseline networks, which consist of the original Monodepthv2 network, and a version of it that considers only specular, polarized reflections. Our proposed method provides better quantitative and qualitative results than both baselines. Our algorithm performs better mainly in the regions where polarization provides valuable clues that are generally hard to analyze by texture-based algorithms. To the best of our knowledge, there is not any work that makes use of the polarization state, and at the same time, stress the fact that they improve the results in these type of materials. Generally, these works show the global accuracy gain over the test set, without specifying the accuracy at some challenging elements.

One important point to note is that the results shown with our deep learning network are based on an uncalibrated setup. Indeed, we are using a third party dataset that did not calibrate the camera in the sense of the polarimetric measurements, and since our camera does not have the same configuration as theirs, it is impossible to do the training with calibration. Therefore, we are unable to estimate the error gap of our data-driven algorithm as we did in Chapter 4.

Despite our contributions, there are still several improvements to carry out in our future work. The proposed loss function in Chapter 5 is computed based on a factor that depends on the detection of the type of reflection at each pixel. In this work, we have considered a mask computed by doing a thresholding operation on the DoLP. The required operation

to do this separation is more complex, and most works that do specular and diffuse light separation produce the intensity separation, and not the polarization parameters separation. Our immediate line of research is aimed at obtaining a diffuse and specular separation for the polarization parameters.

As we mentioned above, the depth estimation neural network is done with a dataset that does not consider sensor calibration. Based on the results presented in Chapter 4, the results obtained with the neural network can be further improved if we use a calibrated camera. Indeed, the calibration will correct the AoLP and DoLP polarization measurements, which will modify the values computed in the loss function. Therefore, the training process will be guided differently when using uncalibrated and calibrated setups. By using the developed toolkit, we aim to do the polarimetric camera calibration, and create a dataset with our camera to analyze the effect of the calibration over the training process.

Finally, we aim to extend the applications included in our software to reach a wider public. Namely, we intend to add a deep learning library to the compilation process, and with it, open the door to the development of data-driven applications for the robotics field.

Bibliography

[1] Basler AG. Basler aca2440-75umpol. https://www.sodavision.com/product/basler-aca2440-75umpol/, 2023.

[2] Khadidja Ould Amer, Marwa Elbouz, Ayman Alfalou, Christian Brosseau, and Jaouad Hajjami. Enhancing underwater optical imaging by using a low-pass polarization filter. *Opt. Express*, 27(2):621–643, Jan 2019.

[3] John B Anderson and Arne Svensson. Coding and information theory. *Coded Modulation Systems*, pages 75–131, 2002.

[4] Oleg Angelsky, Aleksandr Bekshaev, Claudia Zenkova, Dmytro Ivanskyi, and Jun Zheng. Correlation optics, coherence and optical singularities: Basic concepts and practical applications. *Frontiers in Physics*, 10:924508, 06 2022.

[5] European Machine Vision Association. Genicam: The generic interface for cameras standard. https://www.emva.org/standards-technology/genicam/introduction-new/, 2003.

[6] Gary A Atkinson and Edwin R Hancock. Recovery of surface orientation from diffuse polarization. *IEEE transactions on image processing*, 15(6):1653–1664, 2006.

[7] Gary A. Atkinson and Edwin R. Hancock. Surface reconstruction using polarization and photometric stereo. In Walter G. Kropatsch, Martin Kampel, and Allan Hanbury, editors, *Computer Analysis of Images and Patterns*, pages 466–473, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[8] Yunhao Ba, Alex Gilbert, Franklin Wang, Jinfa Yang, Rui Chen, Yiqin Wang, Lei Yan, Boxin Shi, and Achuta Kadambi. Deep shape from polarization. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 554–571, Cham, 2020. Springer International Publishing.

[9] Michael Baltaxe, Tomer Pe'er, and Dan Levi. Polarimetric imaging for perception, 2023.

[10] Kai Berger, Randolph Voorhies, and Larry H. Matthies. Depth from stereo polarization in specular scenes for urban robotics. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1966–1973, 2017.

[11] Marc Blanchon, Désiré Sidibé, Olivier Morel, Ralph Seulin, Daniel Braun, and Fabrice Meriaudeau. P2d: a self-supervised method for depth estimation from polarimetry. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 7357–7364, 2021.

[12] Rachel Blin, Samia Ainouz, Stéphane Canu, and Fabrice Meriaudeau. Road scenes analysis in adverse weather conditions by polarization-encoded images and adapted deep learning. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 27–32, 2019.

[13] Max Born and Emil Wolf. *Principles of optics: electromagnetic theory of propagation, interference and diffraction of light*. Elsevier, 2013.

[14] Said Yacine Boulahia, Abdenour Amamra, Mohamed Ridha Madi, and Said Daikh. Early, intermediate and late fusion strategies for robust deep learning-based multimodal action recognition. *Mach. Vision Appl.*, 32(6), nov 2021.

[15] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, jan 2011.

[16] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[17] Cesar Cadena, Yasir Latif, and Ian D. Reid. Measuring the performance of single image depth estimation methods. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4150–4157, 2016.

[18] Guangcheng Chen, Li He, Yisheng Guan, and Hong Zhang. Perspective phase angle model for polarimetric 3d reconstruction. In *European Conference on Computer Vision*, pages 398–414. Springer, 2022.

[19] Hua Chen and L.B. Wolff. Polarization phase-based method for material classification and object recognition in computer vision. In *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 128–135, 1996.

[20] Zhenyue Chen. Calibration method of microgrid polarimeters with image interpolation. *Applied Optics*, 54:995–1001, 02 2015.

[21] The Qt Company. Qt documentation. https://doc.qt.io/qt-5.15/classes.html, 2022.

[22] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223, 2016.

[23] cppreference. The c++ reference. https://en.cppreference.com/w/, 2020.

[24] Thomas Cronin and Justin Marshall. Patterns and properties of polarized light in air and water. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 366:619–26, 03 2011.

[25] Zhaopeng Cui, Viktor Larsson, and Marc Pollefeys. Polarimetric relative pose estimation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2671–2680, 2019.

[26] Valentin Deschaintre, Yiming Lin, and Abhijeet Ghosh. Deep polarization imaging for 3d shape and svbrdf acquisition. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15562–15571, 2021.

[27] Zhichao Ding, Chunsheng Sun, Hongwei Han, Liheng Ma, and Yonggang Zhao. Calibration method for division-of-focal-plane polarimeters using nonuniform light. *IEEE Photonics Journal*, 13(1):1–9, 2021.

[28] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[29] Tao Du, Changzheng Tian, Jian Yang, Shanpeng Wang, Xin Liu, and Lei Guo. An autonomous initial alignment and observability analysis for sins with bio-inspired polarized skylight sensors. *IEEE Sensors Journal*, 20(14):7941–7956, 2020.

[30] Teledyne Flir. Blackfly s gige. https://www.flir.fr/products/blackfly-s-gige/?model=BFS-PGE-51S5PC-C, 2023.

[31] Yoshiki Fukao, Ryo Kawahara, Shohei Nobuhara, and Ko Nishino. Polarimetric normal stereo. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 682–690, 2021.

[32] Daoyi Gao, Yitong Li, Patrick Ruhkamp, Iuliia Skobleva, Magdalena Wysocki, HyunJun Jung, Pengyuan Wang, Arturo Guridi, and Benjamin Busam. Polarimetric pose prediction.

In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 735–752, Cham, 2022. Springer Nature Switzerland.

[33] Missael Garcia, Christopher Edmiston, Radoslav Marinov, Alexander Vail, and Viktor Gruev. Bio-inspired color-polarization imager for real-time in situ imaging. *Optica*, 4(10):1263–1271, Oct 2017.

[34] S. Garrido-Jurado, R. Muñoz-Salinas, F.J. Madrid-Cuevas, and M.J. Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292, 2014.

[35] Yilbert Gimenez, Pierre-Jean Lapray, Alban Foulonneau, and Laurent Bigué. Calibration algorithms for polarization filter array camera: survey and evaluation. *Journal of Electronic Imaging*, 29:1, 03 2020.

[36] Clement Godard, Oisin Mac Aodha, Michael Firman, and Gabriel Brostow. Digging into self-supervised monocular depth estimation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3827–3837, 2019.

[37] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017.

[38] Nathan Hagen, Shuhei Shibata, and Yukitoshi Otani. Calibration and performance assessment of microgrid polarization cameras. *Optical Engineering*, 58:1, 02 2019.

[39] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[40] E. Hecht. *Optics*. Pearson education. Addison-Wesley, 2002.

[41] Haofeng Hu, Yang Lin, Xiaobo Li, Pengfei Qi, and Tiegen Liu. Iplnet: a neural network for intensity-polarization imaging in low light. *Opt. Lett.*, 45(22):6162–6165, Nov 2020.

[42] Haofeng Hu, Yanbin Zhang, Xiaobo Li, Yang Lin, Zhenzhou Cheng, and Tiegen Liu. Polarimetric underwater image recovery via deep learning. *Optics and Lasers in Engineering*, 133:106152, 2020.

[43] Pengwei Hu, Jian Yang, Lei Guo, Xiang Yu, and Wenshuo Li. Solar-tracking methodology based on refraction-polarization in snell's window for underwater navigation. *Chinese Journal of Aeronautics*, 35(3):380–389, 2022.

[44] Leanne E. Iannucci, Matthew B. Riak, Ethan Meitz, Matthew R. Bersi, Viktor Gruev, and Spencer P. Lake. Effect of matrix properties on transmission and reflectance mode division-of-focal-plane Stokes polarimetry. *Journal of Biomedical Optics*, 28(10):102902, 2023.

[45] Tomoki Ichikawa, Matthew Purri, Ryo Kawahara, Shohei Nobuhara, Kristin Dana, and Ko Nishino. Shape from sky: Polarimetric normal recovery under the sky. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14827–14836, 2021.

[46] jAi. Go series go-5100mp-usb polarized (imx250mzr) machine vision area scan cameras. https://www.jai.com/products/go-5100mp-usb, 2023.

[47] Soma Kajiyama, Taihe Piao, Ryo Kawahara, and Takahiro Okabe. Separating partially-polarized diffuse and specular reflection components under unpolarized light sources. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2548–2557, 2023.

[48] Agastya Kalra, Vage Taamazyan, Supreeth Krishna Rao, Kartik Venkataraman, Ramesh Raskar, and Achuta Kadambi. Deep polarization cues for transparent object segmentation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8599–8608, 2020.

[49] Christos Kampouris, Stefanos Zafeiriou, and Abhijeet Ghosh. Diffuse-specular separation using binary spherical gradient illumination. In *EGSR (EI&I)*, pages 1–10, 2018.

[50] May Phyo Khaing and Mukunoki Masayuki. Transparent object detection using convolutional neural network. In Thi Thi Zin and Jerry Chun-Wei Lin, editors, *Big Data Analysis and Deep Learning Applications*, pages 86–93, Singapore, 2019. Springer Singapore.

[51] Daisuke Kiku, Yusuke Monno, Masayuki Tanaka, and Masatoshi Okutomi. Beyond color difference: Residual interpolation for color image demosaicking. *IEEE Transactions on Image Processing*, 25(3):1288–1300, 2016.

[52] Kichang Kim, Akihiko Torii, and Masatoshi Okutomi. Multi-view inverse rendering under arbitrary illumination and albedo. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 750–767, Cham, 2016. Springer International Publishing.

[53] Yuhi Kondo, Taishi Ono, Legong Sun, Yasutaka Hirasawa, and Jun Murayama. Accurate polarimetric brdf for real polarization scene rendering. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 220–236, Cham, 2020. Springer International Publishing.

[54] Jan Korger, Tobias Kolb, Peter Banzer, Andrea Aiello, Christoffer Wittmann, Christoph Marquardt, and Gerd Leuchs. The polarization properties of a tilted polarizer. *Opt. Express*, 21(22):27032–27042, Nov 2013.

[55] Thomas Kronland-Martinet, Léo Poughon, Marcel Pasquinelli, David Duché, Julien R. Serres, and Stéphane Viollet. Skypole—a method for locating the north celestial pole from skylight polarization patterns. *Proceedings of the National Academy of Sciences*, 120(30):e2304847120, 2023.

[56] Bruce Lamond, Pieter Peers, Abhijeet Ghosh, and Paul Debevec. Image-based separation of diffuse and specular reflections using environmental structured illumination. In *2009 IEEE International Conference on Computational Photography (ICCP)*, pages 1–8, 2009.

[57] Connor Lane, David Rode, and Thomas Roesgen. Calibration of a polarization image sensor andinvestigation of influencing factors. *Applied Optics*, 61, 10 2021.

[58] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation, 2021.

[59] Chenyang Lei, Xuhua Huang, Mengdi Zhang, Qiong Yan, Wenxiu Sun, and Qifeng Chen. Polarized reflection removal with perfect alignment in the wild. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1747–1755, 2020.

[60] Chenyang Lei, Chenyang Qi, Jiaxin Xie, Na Fan, Vladlen Koltun, and Qifeng Chen. Shape from polarization for complex scenes in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12632–12641, June 2022.

[61] Jinshan Li, Jinkui Chu, Ran Zhang, Jianhua Chen, and Yinlong Wang. Bio-inspired attitude measurement method using a polarization skylight and a gravitational field. *Appl. Opt.*, 59(9):2955–2962, Mar 2020.

[62] Ning Li, Yongqiang Zhao, Quan Pan, Seong G. Kong, and Jonathan Cheung-Wai Chan. Illumination-invariant road detection and tracking using lwir polarization characteristics. *ISPRS Journal of Photogrammetry and Remote Sensing*, 180:357–369, 2021.

[63] Ning Li, Yongqiang Zhao, Rongyuan Wu, and Quan Pan. Polarization-guided road detection network for lwir division-of-focal-plane camera. *Opt. Lett.*, 46(22):5679–5682, Nov 2021.

[64] Xiaobo Li, Jianuo Xu, Liping Zhang, Haofeng Hu, and Shih-Chi Chen. Underwater image restoration via stokes decomposition. *Opt. Lett.*, 47(11):2854–2857, Jun 2022.

[65] Xiaobo Li, Lei Yan, Pengfei Qi, Liping Zhang, François Goudail, Tiegen Liu, Jingsheng Zhai, and Haofeng Hu. Polarimetric imaging via deep learning: A review. *Remote Sensing*, 15(6), 2023.

[66] Yupeng Liang, Ryosuke Wakaki, Shohei Nobuhara, and Ko Nishino. Multimodal material segmentation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19768–19776, 2022.

[67] S. Lin and Heung-Yeung Shum. Separation of diffuse and specular reflection in color images. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–I, 2001.

[68] Ju Liu, Jin Duan, Youfei Hao, Guangqiu Chen, and Hao Zhang. Semantic-guided polarization image fusion method based on a dual-discriminator gan. *Opt. Express*, 30(24):43601–43621, Nov 2022.

[69] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9992–10002, Los Alamitos, CA, USA, oct 2021. IEEE Computer Society.

[70] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Smpl: A skinned multi-person linear model. *ACM Trans. Graph.*, 34(6), nov 2015.

[71] P. Marconnet, Luc Gendre, A. Foulonneau, and Laurent Bigué. Cancellation of motion artifacts caused by a division-of-time polarimeter. *Proc SPIE*, 8160, 09 2011.

[72] Haiyang Mei, Bo Dong, Wen Dong, Jiaxi Yang, Seung-Hwan Baek, Felix Heide, Pieter Peers, Xiaopeng Wei, and Xin Yang. Glass segmentation using intensity and spectral polarization cues. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12612–12621, 2022.

[73] Miyazaki, Tan, Hara, and Ikeuchi. Polarization-based inverse rendering from a single view. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 982–987 vol.2, 2003.

[74] Olivier Morel, Fabrice Meriaudeau, Christophe Stolz, and Patrick Gorria. Polarization imaging applied to 3D reconstruction of specular metallic surfaces. In Jeffery R. Price and Fabrice Meriaudeau, editors, *Machine Vision Applications in Industrial Inspection XIII*, volume 5679, pages 178 – 186. International Society for Optics and Photonics, SPIE, 2005.

[75] Olivier Morel, Ralph Seulin, and David Fofi. Handy method to calibrate division-of-amplitude polarimeters for the first three stokes parameters. *Optics Express*, 24:13634, 06 2016.

[76] Miki Morimatsu, Yusuke Monno, Masayuki Tanaka, and Masatoshi Okutomi. Monochrome and color polarization demosaicking using edge-aware residual interpolation. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 2571–2575, 2020.

[77] Izaak Neutelings. Electromagnetic wave. https://tikz.net/electromagnetic_wave/, 2018.

[78] Tam Nguyen, Quang Nhat Vo, Hyung-Jeong Yang, Soo-Hyung Kim, and Guee-Sang Lee. Separation of specular and diffuse components using tensor voting in color images. *Appl. Opt.*, 53(33):7924–7936, Nov 2014.

[79] Taishi Ono, Yuhi Kondo, Legong Sun, Teppei Kurita, and Yusuke Moriuchi. Degree-of-linear-polarization-based color constancy. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19708–19717, 2022.

[80] Zhiliang Peng, Wei Huang, Shanzhi Gu, Lingxi Xie, Yaowei Wang, Jianbin Jiao, and Qixiang Ye. Conformer: Local features coupling global representations for visual recognition. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 357–366, 2021.

[81] R. Perez, R. Seals, and J. Michalsky. All-weather model for sky luminance distribution—preliminary configuration and validation. *Solar Energy*, 50(3):235–245, 1993.

[82] Mara Pistellato and Filippo Bergamasco. A geometric model for polarization imaging on projective cameras. *arXiv preprint arXiv:2211.16986*, 2022.

[83] Bryan Poling. A tutorial on camera models. *University of Minnesota*, pages 1–10, 2015.

[84] S Powell and Viktor Gruev. Calibration methods for division-of-focal-plane polarimeters. *Optics express*, 21:21039–21055, 09 2013.

[85] S. Bear Powell and Viktor Gruev. Calibration methods for division-of-focal-plane polarimeters. *Opt. Express*, 21(18):21039–21055, Sep 2013.

[86] Wei Qingchao, Zhou GuoQing, Zhu Qing, and Zhang Qinghang. On dlt method for ccd camera calibration. In *Proceedings of Third International Conference on Signal Processing (ICSP'96)*, volume 2, pages 883–885 vol.2, 1996.

[87] Morgan Quigley, Brian Gerkey, Ken Conley, Josh Faust, Tully Foote, Jeremy Leibs, Eric Berger, Rob Wheeler, and Andrew Ng. Ros: an open-source robot operating system. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA) Workshop on Open Source Robotics*, Kobe, Japan, May 2009.

[88] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3):1623–1637, 2022.

[89] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3):1623–1637, 2022.

[90] Joaquin Rodriguez, Lew Lew-Yan-Voon, Renato Martins, and Olivier Morel. Pola4all: A survey of polarimetric applications and an open-source software to analyze polarimetric images - repository. https://github.com/vibot-lab/Pola4all_2023.

[91] Joaquin Rodriguez, Lew Lew-Yan-Voon, Renato Martins, and Olivier Morel. A practical calibration method for rgb micro-grid polarimetric cameras. *IEEE Robotics and Automation Letters*, 7(4):9921–9928, 2022.

[92] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

[93] Yoav Y. Schechner. Self-calibrating imaging polarimetry. In *IEEE International Conference on Computational Photography*, 2015.

[94] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, 2016.

[95] Moein Shakeri, Shing Yang Loo, Hong Zhang, and Kangkang Hu. Polarimetric monocular dense mapping using relative deep depth prior. *IEEE Robotics and Automation Letters*, 6(3):4512–4519, 2021.

[96] L. Shen and Y. Zhao. Underwater image enhancement based on polarization imaging. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLIII-B1-2020:579–585, 2020.

[97] William A. P. Smith, Ravi Ramamoorthi, and Silvia Tozza. Linear depth estimation from an uncalibrated, monocular polarisation image. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 109–125, Cham, 2016. Springer International Publishing.

[98] Hon. J.W. Strutt. Lviii. on the scattering of light by small particles. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 41(275):447–454, 1871.

[99] Kenichiro Tanaka, Yasuhiro Mukaigawa, and Achuta Kadambi. Polarized non-line-of-sight imaging. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2133–2142, 2020.

[100] Bo Tao, Yunfei Shen, Xiliang Tong, Du Jiang, and Baojia Chen. Depth estimation using feature pyramid u-net and polarized self-attention for road scenes. *Photonics*, 9(7), 2022.

[101] Vimal Thilak, David G. Voelz, and Charles D. Creusere. Polarization-based index of refraction and reflection angle estimation for remote sensing applications. *Appl. Opt.*, 46(30):7527–7536, Oct 2007.

[102] Nygel Thomas, Mayaluri Zefree Lazarus, and Supratim Gupta. Separation of diffuse and specular reflection components from real-world color images captured under flash imaging conditions. In Afzal Sikander, Dulal Acharjee, Chandan Kumar Chanda, Pranab Kumar Mondal, and Piyush Verma, editors, *Energy Systems, Drives and Automations*, pages 265–275, Singapore, 2020. Springer Singapore.

[103] Shoji Tominaga and Akira Kimachi. Polarization imaging for material classification. *Optical Engineering*, 47(12):123201, 2008.

[104] J. Scott Tyo. Design of optimal polarimeters: maximization of signal-to-noise ratio and minimization of systematic error. *Appl. Opt.*, 41(4):619–630, Feb 2002.

[105] Masada Tzabari and Yoav Y. Schechner. Polarized optical-flow gyroscope. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI*, page 363–381, Berlin, Heidelberg, 2020. Springer-Verlag.

[106] S. Umeyama and G. Godin. Separation of diffuse and specular components of surface reflection by use of polarization and statistical analysis of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):639–647, 2004.

[107] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, Inc., 2017.

[108] Yannick Verdié, Jifei Song, Barnabé Mas, Benjamin Busam, Aleš Leonardis, and Steven McDonagh. Cromo: Cross-modal learning for monocular depth estimation, 2022.

[109] Lucid Vision. Phoenix 5.0 mp polarization model (imx250mzr/myr). https://thinklucid.com/product/phoenix-5-0-mp-polarized-model/, 2023.

[110] Xin Wang, HuaZhi Pan, Kai Guo, Xinli Yang, and Sheng Luo. The evolution of lidar and its application in high precision measurement. *IOP Conference Series: Earth and Environmental Science*, 502(1):012008, may 2020.

[111] Zhixiang Wang, Yinqiang Zheng, and Yung-Yu Chuang. Polarimetric camera calibration using an lcd monitor. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[112] Jamie Watson, Oisin Mac Aodha, Victor Prisacariu, Gabriel Brostow, and Michael Firman. The temporal opportunist: Self-supervised multi-frame monocular depth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1164–1174, June 2021.

[113] Sijia Wen, Yinqiang Zheng, and Feng Lu. Polarization guided specular reflection separation. *IEEE Transactions on Image Processing*, 30:7280–7291, 2021.

[114] Sijia Wen, Yinqiang Zheng, Feng Lu, and Qinping Zhao. Joint chromatic and polarimetric demosaicing via sparse coding. *CoRR*, abs/1912.07308, 2019.

[115] L.B. Wolff. Polarization-based material classification from specular reflection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(11):1059–1071, 1990.

[116] Lawrence B. Wolff. Polarization vision: a new sensory approach to image understanding. *Image and Vision Computing*, 15(2):81–93, 1997.

[117] Oliver Woodford, Philip Torr, Ian Reid, and Andrew Fitzgibbon. Global stereo reconstruction under second-order smoothness priors. *IEEE transactions on pattern analysis and machine intelligence*, 31(12):2115–2128, 2009.

[118] Kaite Xiang, Kailun Yang, and Kaiwei Wang. Polarization-driven semantic segmentation via efficient attention-bridged fusion. *Opt. Express*, 29(4):4802–4820, Feb 2021.

[119] Jianwei Yang, Lixing Liu, and Stan Li. Separating specular and diffuse reflection components in the hsi color space. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 891–898, 2013.

[120] Nan Yang, Lukas von Stumberg, Rui Wang, and Daniel Cremers. D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[121] Lichi Zhang and Edwin R. Hancock. A comprehensive polarisation model for surface orientation recovery. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pages 3791–3794, 2012.

[122] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22(11):1330–1334, 2000.

[123] Zhengyou Zhang. Microsoft kinect sensor and its effect. *IEEE MultiMedia*, 19(2):4–10, 2012.

[124] Jinyu Zhao, Yusuke Monno, and Masatoshi Okutomi. Polarimetric multi-view inverse rendering. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 85–102, Cham, 2020. Springer International Publishing.

[125] Chu Zhou, Minggui Teng, Yufei Han, Chao Xu, and Boxin Shi. Learning to dehaze with polarization. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 11487–11500. Curran Associates, Inc., 2021.

[126] Dizhong Zhu and William A. P. Smith. Depth from a polarisation + rgb stereo pair. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7578–7587, 2019.

[127] Shihao Zou, Xinxin Zuo, Sen Wang, Yiming Qian, Chuan Guo, and Li Cheng. Human pose and shape estimation from single polarization images. *IEEE Transactions on Multimedia*, pages 1–1, 2022.

Appendix A

Explanation of angle of linear polarization estimator error

A.1 Base theory

The polarization state of a light source is defined by its 4D Stokes vector $\mathbf{S}$. If the polarization analyzer has only linear filters, then only the first three components of this vector can be measured. This is the case of our camera, thus, S_3 will not be considered. In other words, we are interested in the linear Stokes vector $\mathbf{S} = [S_0,\ S_1,\ S_2]^T$. From this vector, the Angle of Linear Polarization α and the Degree of Linear Polarization ρ can be computed as:

$$\alpha = \frac{1}{2} atan\left(\frac{S_2}{S_1}\right) \quad \rho = \frac{\sqrt{S_1^2 + S_2^2}}{S_0}. \tag{A.1}$$

In order to measure the Stokes vector, and therefore, its linear components, at least three measurements with a linear polarizer at three different orientations are required. In the case of a RGB DoFP sensor, a set of 2×2 pixels with the same color filter can be used for this purpose. Each pixel of this set has a linear polarization filter oriented at 0°, 45°, 90°, and 135°. If these filters are perfectly placed at these orientations, and the pixel qualities are

ideal, then the Stokes vector can be computed from these measurements as:

$$\begin{bmatrix} S_0 \\ S_1 \\ S_2 \end{bmatrix} = \begin{bmatrix} \frac{I_0 + I_{45} + I_{90} + I_{135}}{2} \\ I_0 - I_{90} \\ I_{45} - I_{135} \end{bmatrix}, \tag{A.2}$$

where I_0, I_{45}, I_{90}, and I_{135} are respectively, the intensity measurements of the pixels whose filter has an orientation of 0°, 45°, 90°, and 135°. This equation can be demonstrated by using Eq. (3.5) in Sec. 3.2.1. Indeed, when the camera is ideal, $T_i = 0.5$, $P_i = 1.0$, and $\theta_i = i$, for $i = \{0°, 45°, 90°, 135°\}$. As a consequence, the identity shown in Eq. (A.2) allows to measure the AoLP α directly from the super-pixel intensities.

Nevertheless, in the general case, the pixels and the filters are not ideal, and the filter orientations are not perfect. Therefore, using Eqs. (A.1) and (A.2) to estimate α brings an error that must be quantified. In order to demonstrate the shape of this error, we need to remember two set of mathematical properties:

- Sine and cosine properties:

$$\begin{aligned} sin\left(\tfrac{\pi}{2} + \theta\right) &= cos\left(\theta\right) \quad & sin\left(\tfrac{3\pi}{2} + \theta\right) &= -cos\left(\theta\right) \\ cos\left(\tfrac{\pi}{2} + \theta\right) &= -sin\left(\theta\right) \quad & cos\left(\tfrac{3\pi}{2} + \theta\right) &= sin\left(\theta\right) \\ sin\left(\pi + \theta\right) &= -sin\left(\theta\right) \quad & cos\left(\pi + \theta\right) &= -cos\left(\theta\right) \end{aligned} \tag{A.3}$$

- Taylor expansion of sine and cosine functions up to order 2, around $\theta = 0$:

$$sin\left(\theta\right) \simeq \theta \qquad\qquad cos\left(\theta\right) \simeq 1 - \frac{\theta^2}{2} \tag{A.4}$$

  where θ is measured in radians.

Additionally, when a pixel and its polarization filter are not ideal, the relationship between the Stokes vector and the measured intensity is given by Eq. (3.2). This equation has been

copied here for convenience:

$$I_i = T_i \begin{bmatrix} \frac{1}{P_i} & cos\,(2\theta_i) & sin\,(2\theta_i) \end{bmatrix} \begin{bmatrix} S_0 \\ S_1 \\ S_2 \end{bmatrix}, \tag{A.5}$$

where I_i is the measured pixel intensity, T_i is the pixel gain, P_i is a factor that models the non-ideality of the pixel micro-filter, and θ_i is the micro-filter orientation, for $i = \{0°, 45°, 90°, 135°\}$.

A.2 Demonstration

To demonstrate the error equation in the estimated AoLP, we compute the error in the measured Stokes components when the pixels are considered ideal.

$$\begin{aligned} \hat{S}_1 &= I_0 - I_{90} \\ \hat{S}_2 &= I_{45} - I_{135} \end{aligned} \tag{A.6}$$

Considering the pixel model of Eq. (A.5), and assuming that the micro-filter with orien-

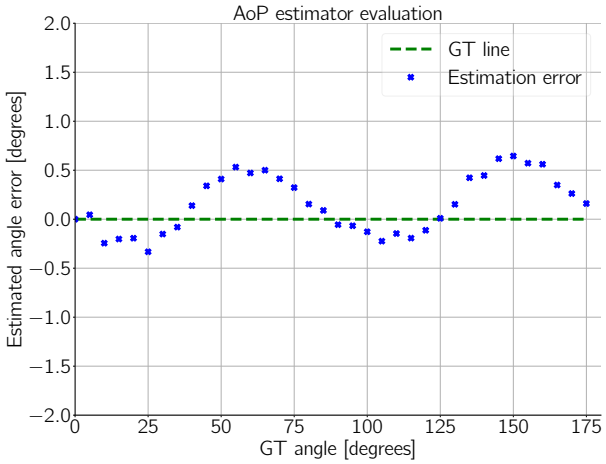


Figure A.1: AoLP estimator error plot.

tation i has an error $\Delta\theta_i$ with respect to its ideal orientation, we obtain:

$$\hat{S_1} = \left[T_0 \left[\begin{array}{ccc} \frac{1}{P_0} & cos\left(2\Delta\theta_0\right) & sin\left(2\Delta\theta_0\right) \end{array} \right] - \right.$$
$$\left. T_{90} \left[\begin{array}{ccc} \frac{1}{P_{90}} & cos\left(\pi + 2\Delta\theta_{90}\right) & sin\left(\pi + 2\Delta\theta_{90}\right) \end{array} \right] \right] \left[\begin{array}{c} S_0 \\ S_1 \\ S_2 \end{array} \right] \tag{A.7}$$

Using the sine and cosine properties, and grouping terms gives:

$$\hat{S_1} = \left[\begin{array}{ccc} A & B & C \end{array} \right] \left[\begin{array}{c} S_0 \\ S_1 \\ S_2 \end{array} \right], \tag{A.8}$$

with

$$A = \frac{T_0}{P_0} - \frac{T_{90}}{P_{90}}$$

$$B = T_0 cos\left(2\Delta\theta_0\right) + T_{90} cos\left(2\Delta\theta_{90}\right)$$

$$C = T_0 sin\left(2\Delta\theta_0\right) + T_{90} sin\left(2\Delta\theta_{90}\right)$$

Since the angle error can be considered close to zero, then the corresponding Taylor expansions in Eq. (A.4) can be used to replace the sine and cosine functions. Moreover, by doing the matricial multiplication we obtain:

$$\hat{S_1} = AS_0 + \left[T_0 + T_{90} - 2T_0\Delta\theta_0^2 - 2T_{90}\Delta\theta_{90}^2\right] S_1 +$$
$$\left[2T_0\Delta\theta_0 + 2T_{90}\Delta\theta_{90}\right] S_2. \tag{A.9}$$

If the angle errors are between $[-10°, 10°]$, the corresponding range in radians is $[-0.1745, 0.1745]$. Thus, if we square this range, we obtain a range of values $[0, 0.03]$. The orientation errors due to manufacturing problems have values less than the given example, therefore, the second order variables can be neglected.

$$\hat{S_1} = AS_0 + G^{'} S_1 + K_1 S_2. \tag{A.10}$$

with:

$$G^{'} = T_0 + T_{90}$$

$$K_1 = 2T_0\Delta\theta_0 + 2T_{90}\Delta\theta_{90}.$$

Similarly, $\hat{S}_2$ can be obtained as a function of the Stokes components.

$$\hat{S}_2 = DS_0 - K_2S_1 + G^{''}S_2. \tag{A.11}$$

where:

$$D = \frac{T_{45}}{P_{45}} - \frac{T_{135}}{P_{135}}$$

$$G^{''} = T_{45} + T_{135}$$

$$K_2 = 2T_{45}\Delta\theta_{45} + 2T_{135}\Delta\theta_{135}.$$

It follows that the estimated AoLP $\hat{\alpha}$ is equal to:

$$\hat{\alpha} = \frac{1}{2}atan\left(\frac{\hat{S}_2}{\hat{S}_1}\right) = \frac{1}{2}atan\left(\frac{DS_0 - K_2S_1 + G^{''}S_2}{AS_0 + G^{'}S_1 + K_1S_2}\right) \tag{A.12}$$

Remembering that $S_1 = S_0\rho\cos(2\alpha)$, and $S_2 = S_0\rho\sin(2\alpha)$, where ρ is the degree of linear polarization, and α is the real angle of linear polarization of the incoming light, Eq. (A.12) becomes Eq. (A.13).

$$\hat{\alpha} = \frac{1}{2}atan\left(\frac{D - K_2\rho cos(2\alpha) + G^{''}\rho sin(2\alpha)}{A + G^{'}\rho cos(2\alpha) + K_1\rho sin(2\alpha)}\right) \tag{A.13}$$

This equation converges to the true AoLP α if the pixels and the filters are ideal, i.e., $P_i = 1$, $T_i = 0.5$, and $\Delta\theta_i = 0$, for $i = \{0°, 45°, 90°, 135°\}$. In a real case, slight deviations from these values will appear. The sources of these deviations are the manufacturing process of the sensor, and the lens added to the camera. As mentioned in Sec. 3.2.2, considering a small region around the center of the sensor reduces the deviations caused by the lens.

Analyzing this equation, it is possible to conclude that:

- The deviations in the pixel parameters will make the other Stokes parameters to influence the AoLP measurement.

- The deviations in the orientations of the micro-polarizers, denoted by $\Delta\theta_i$, for $i = \{0°, 45°, 90°, 135°\}$ will introduce an error based on the value of the complementary Stokes parameter (for the measurement of S_1, a deviation in the orientation of the micro-polarizers will introduce an error based on the value of S_2, and an error based on S_1 in the measurement of S_2).

- For measurements of the same light at different AoLP, the deviations in the non-ideality factor P_i and the gain T_i will produce a constant shift in both, numerator and denominator, of Eq. (A.13).

- The values of K_1 and K_2 should be close to zero, and this can happen in two situations: either the pixel parameters are almost ideal and therefore the orientation errors are almost zero, or the pixels orientation error are almost the same, but in opposite directions. The second case can be understood by looking at the definitions of these variables. For instance, $K_1 = 2T_0\Delta\theta_0 + 2T_{90}\Delta\theta_{90}$, where T_0 and T_{90} are positive numbers, and in general, they are close to 0.5. Therefore, if $\Delta\theta_0 \simeq -\Delta\theta_{90}$, then K_1 will have a very tiny value. Similarly, K_2 will be almost zero when $\Delta\theta_{45} \simeq -\Delta\theta_{135}$. Finally, it can be seen that the errors in the orientations can be compensated if they are in opposite directions.

In all the cases, the errors will produce sine-like functions, since they will change the ratio of the sine to the cosine functions. Nevertheless, the effect of each parameter to the final shape of the error is different. The error in the orientations can change only the minimum and maximum values in the estimation error function, and the factors T_i and P_i can create sine shaped error functions and additionally change the position of its extreme values.

A.3 Experiments

In this section, the error function has been computed for several set of samples. The samples to which the functions are fitted have been captured using the RGB-polarization sensor with

the following lenses:

- Lens 1: Fuji-film HF8XA-5M - F1.6/8mm
- Lens 2: Fuji-film HF12XA-5M - F1.6/12mm
- Lens 3: Fuji-film HF16XA-5M - F1.6/16mm
- Lens 4: Fuji-film HF25XA-5M - F1.6/25mm

Additionally, all the lenses have been correctly focused on the light source used, and their F-number have been set to 3, which is higher than 2.8. This configuration have been chosen to comply with the recommendations given by [57].

To run this experiment, the AoLP estimator as described in the Sec. 3.2.2 has been used. Then, with a uniform unpolarized light source and a rotative linear polarization filter, a linearly polarized light is generated. The position of the filter is changed progressively in the range $[0°, 180°]$, with a step of $5°$. The reference angle of linear polarization of each sample have been measured from the rotative mount of the linear filter. Additionally, the AoLP is estimated with the implemented algorithm for each of these samples. Finally, the error between the reference value and the estimation is computed and plotted in Fig. A.2. To avoid a constant shift in the measurements due to misalignment, since the first reference angle is zero, the first estimated angle have been subtracted from all the estimations. By using a least-squares optimizer, the pixel parameters have been found for each set of samples taken with the different lenses. These parameters are shown in Tab. A.1. For creating this data, the degree of linear polarization used is $\rho = 0.97$.

As shown in Fig. A.2, estimating the AoLP by doing the circular average of the measurements given by the central pixels produces a maximum error of $0.65°$. This upper limit is valid for all the tested lenses.

Tab. A.1 shows all the pixel parameters obtained by least-squares optimization of Eq. (A.13) with the real data. From this table it is possible to confirm the effective pixel values are not far away from the ideal ones. Particularly, the maximum orientation error is $\Delta\theta_0 = 1.47°$. Nonetheless, as explained in the previous section, this error is compensated by the complementary pixel orientation which is in this case $\Delta\theta_{90} = -1.32°$. Additionally, the values

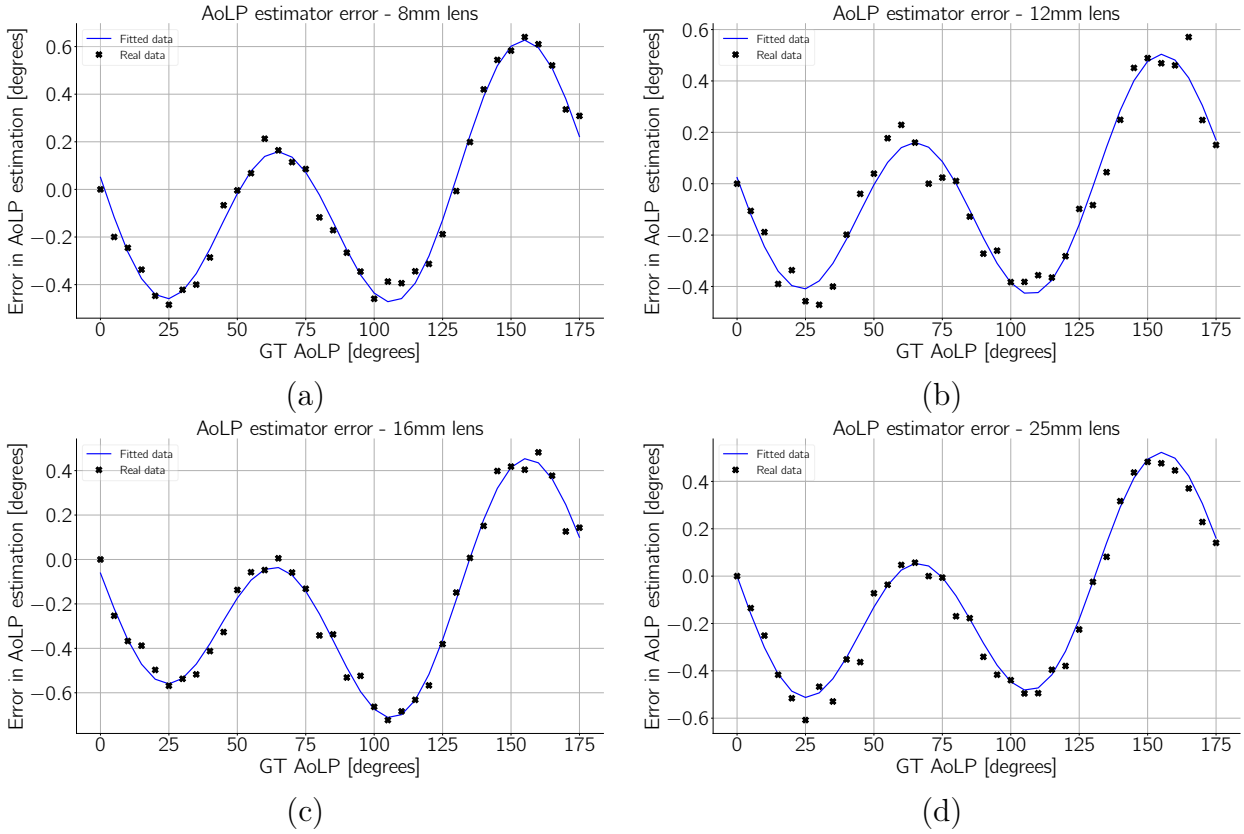


Figure A.2: Error in the estimated AoLP, and curve fitted of Eq. (A.13) for different cases. (a) Lens 1. (b) Lens 2. (c) Lens 3. (d) Lens 4

Lens model	Parameter	$i = \{0, 45, 90, 135\}$
T_i	Lens 1	$[0.525, 0.542, 0.548, 0.499]$
	Lens 2	$[0.53, 0.58, 0.57, 0.49]$
	Lens 3	$[0.53, 0.48, 0.49, 0.51]$
	Lens 4	$[0.52, 0.49, 0.44, 0.45]$
P_i	Lens 1	$[1.009, 0.992, 1.067, 0.922]$
	Lens 2	$[0.96, 1.06, 1.03, 0.91]$
	Lens 3	$[1.04, 0.88, 0.97, 0.95]$
	Lens 4	$[1.09, 1.05, 0.95, 0.98]$
$\Delta\theta_i$	Lens 1	$[0.109, 0.114, -0.192, 0.092]$
	Lens 2	$[-0.42, -0.94, 0.37, 1.31]$
	Lens 3	$[1.47, -1.07, -1.32, 1.54]$
	Lens 4	$[0.07, 0.17, 0.07, 0.16]$

Table A.1: Parameters obtained by non-linear optimization for Eq. (A.13).

exposed in this table show that the different lenses influence the pixel parameters. Indeed, the figures have similar shapes, but the corresponding maximum values are not the same, and they are located at different positions. This is because the corresponding pixel parameters have changed for each case.

As a conclusion, it is possible to confirm that using the ideal values of the pixels parameters introduces an error in the estimation of the AoLP, and this error depends on the measured angle, and the actual pixel parameters. Even though the pixels can be ideal, placing a lens between the light to measure and the sensor will introduce an error in the measured Stokes vector that is reflected as a change in the effective pixel parameter values.

To minimize these deviations, a better estimation of the Stokes vector must be used. This better estimation can be obtained by doing calibration to find the real values of the pixel parameters.

Appendix B

Pola4All: Acquisition, development, and integration platform

When working in any project, it is important to develop software tools that can help the every day work. The time invested when doing this, can reduce and ease the work to do later on. To comply with this objective, it is necessary to be able to identify and to see in advance, the type of tasks which we will have to deal with, and which of them are going to be done repeatedly. Additionally, the source code should be organized in such a manner that it allows easy addition or suppression of components without compromising the other functionalities. Finally, to optimize our time, if there are several ways to perform a task, we should choose the set of instructions that runs it in the shortest time.

In this appendix, we describe a complete software toolkit with a graphical user interface that we have developed during this thesis. This software allows us to work with the color-polarization camera in a very simple manner. It can also serve to develop computer vision tasks oriented towards robotics applications. We present its code base, its source code organization, its components, and the algorithms included in it.

B.1 Motivation

Nowadays in the market, there exist several camera models that make use of a polarization sensor of the type DoFP. Lucid Vision Phoenix [109], Flir Blackfly [30], jAi Go [46], and

Basler Ace [1] are some of the examples that are powered by the Sony Polarsens polarization sensor (either RGB, either monochrome). Each camera can be divided into three main components: the sensor itself, the electronic board that makes the interface between the sensor and the acquisition machine, and the related software that is able to communicate with the camera, retrieve the images from the sensor, and process them.

We have identified several weak points in all these systems. Firstly, the electronic board is not common to all the available cameras. Each of them includes a set of functionalities that are not the same among manufacturers. Furthermore, even though the common API GeniCam [5] is generally used for industrial cameras, the way the electronic board is initialized, the specific connection configuration, and the set of particular instructions included with each camera remain specific to each manufacturer. Regarding the software provided by the manufacturer, it only allows to work with the camera produced by the manufacturer. Additionally, even though the sensor is the same, each manufacturer decides whether to include or not polarimetric processing algorithms. For instance, the Basler AG cameras do not provide any consideration for polarization in their software (not even the interpolation nor the split by polarization channel), and Lucid Vision includes up to the computation of the AoLP and the DoLP in their software. Although some manufacturers may include some polarization algorithms in it, the latest research outcomes are not included in their software, and we cannot modify their software suites since they are closed source.

For all these reasons, we created a new software toolkit that aims to overcome all the previously mentioned difficulties. This software is thought to work with any camera available in the market that makes use of a DoFP technology. The architecture of the software is split in two main blocks: one that makes the interaction with the physical camera and it creates a common interface to communicate with it, and another block that contains all the polarization related processing (camera configuration, polarization algorithms, results display, calibration, and data analysis). The chosen algorithms have been developed to take the minimum possible time (for instance, the calibration data loading and the calibration step take 8 seconds each).

The switch from one camera to another (even if they are from different manufacturers) is done in the first block by changing a single variable on start-up. If the available camera

is not included in our software, the only required step is to write a software driver with the most basic instructions (initialization, change acquisition parameters, and retrieve images). Once done, the rest of the software does not change.

Finally, the communication between the camera and the graphical user interface is done using the middleware Robot Operating System (ROS), which enables the usage of the camera with a more complex robotics system. The entire code has been made publicly available in GitHub [90]. In the rest of this chapter, we will detail the developed software properties, and the implemented algorithms, and we will show the different outcomes we can obtain with it.

B.2 The software

In what follows, we briefly describe the software, including its basic components and the implemented image-processing algorithms. The objective is to contribute to the community by providing several tools integrated with a single, common Graphical User Interface (GUI) software that can interact with any color-polarization camera available on the market. Additionally, this software is meant to provide access to all the developed algorithms that showcase the power of polarization information, making it easier for anyone interested in working with polarization modality to get started.

This software has been developed in C++ [23] to achieve faster execution speeds than in other languages, such as Python, and using Object Oriented Programming (OOP). All the code has been well documented, thus people interested in a deeper understanding of the algorithms can easily understand how they work. Three main libraries have been used for this project:

- OpenCV [16]: An open-source library that includes several algorithms for Computer Vision tasks, and it eases large matrices manipulations.

- Robot Operating System (ROS) [87]: A framework for Robotics applications that enables the creation of distributed systems, and establishes an abstraction layer between the sensors and the top-level applications.

- Qt5 [21]: A framework that serves to create graphical user interfaces. It is for free

when it is not for commercial use, which is our case.

The library is composed of several modules designed to be independent so as to enable easy maintenance and debugging. All the modules are integrated and connected in the main program file, giving a structure like a star: the main program is in the middle, and the other modules are around it. With such a structure the addition or removal of a functionality is straightforward. Regarding the architecture of the code, it is composed of two general components: the camera server, and the Graphical User Interface (GUI) client. They are detailed in the documentation of the library repository, and in what follows we will summarize its components and functionalities.

B.2.1 Core components and basic processing

B.2.2 Camera server

The first module is a ROS package that works as a camera server. This server will interact directly with the physical camera, getting images from it, and changing (or querying) its parameters as the pixel gain, exposure time, and frame rate. Then, the captured images are sent through a communication channel called *topic* in the ROS nomenclature. When information is sent through a topic it is said that the information has been *published* in the topic. Once a topic is made available, a client can connect to it to receive the data. In that case, it is said that the client has *subscribed* to the topic. Only the topic name and the type of data it transports need to be known to establish the connection between the server and the client. Furthermore, several applications can subscribe to a single topic, and receive the same data each time it is published.

The ROS server will publish only raw images from the camera, without any processing made on it. This way, the user gets the most basic information the camera can provide, and it is not affected by any manufacturer-dependent algorithm, as in general, these algorithms are not open source. Another functionality implemented in the server is that the client can request the camera parameters as well as change them. This is done through *services* in the ROS nomenclature, which means that the user in the client side can request information or set information synchronously with the server, and the execution will be blocked until the

operation is finished. This functionality is not the same as when the user receive images. The images are sent to the client as soon as they are captured by the camera, and the first one is informed when this happens. The last image received is stored locally in the client side, and when the user request the image, only a data copy needs to be executed. This avoids communication hangs and late graphical responses in the client side, since the image transfer takes time.

Regarding the code organization of the server, an abstraction layer has been included in the code to ease the addition of new camera models to the server side. If a new camera model wants to be added to the server, three steps must be followed:

1. A C++ class has to be created that implements some basic functions detailed in the interface file *IPolarizationCamera*. Mainly, they are camera initialization, functions to change the acquisition parameters, and captured image retrieval.

2. A string that identifies the new class model has to be included in a variable that contains the list of driver options.

3. The particular instance of the camera model has to be included in the *CameraHandler* file, in the constructor function. Then, the new camera driver will be used when the corresponding camera model is selected.

This is required to be able to dynamically select the correct driver to use. When starting the server, the provided identifier as argument will inform the software the connection type to be used. The advantage of using ROS is that, once the raw image is sent through the topic, the software needs no information about the camera model. Furthermore, ROS allows a remote connection between the server and the client. As a consequence, a small embedded computer with a WiFi connection can be connected to the camera, and the images can be retrieved in a local computer connected to the same network, where the GUI application is running. The architecture of the interaction between the client and the server is illustrated in Fig. B.1.

At the moment of writing this thesis, the software counts with two drivers implemented. One for the Basler acA2440-75ucPOL camera, which is a USB3 color polarization camera

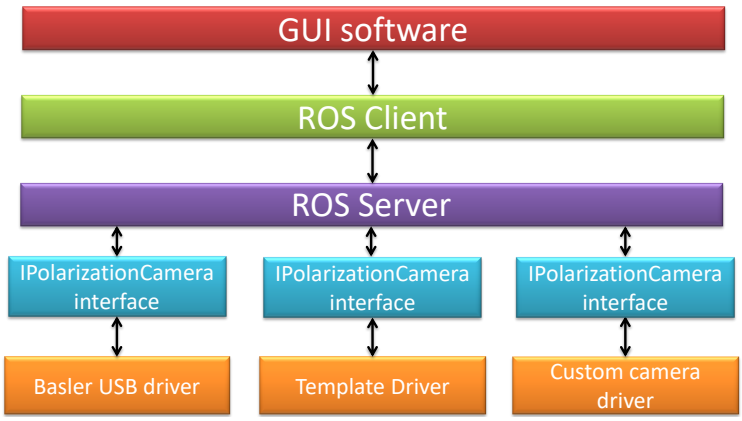


Figure B.1: Software communication architecture. Each camera model has its own driver code. Those drivers are implementations of a common class called *IPolarizationCamera*, which defines a set of functions, transversal to any camera model. Then, the ROS server interacts with each camera by using only the functions defined in the common interface. The GUI talks to this server through the ROS client module. Thereby, it does not need to know which camera we are using, and how it works. The way to exchange information is always the same for it.

powered by the Sony Polarsens IMX250MYR sensor. This driver can be used with the ID *BaslerUSB*. The second driver is called Template, which can be accessed with the ID *TemplateDriver*. This last case is used for the development of new algorithms when a camera is not available physically, or the algorithms want to be tested with reference images. It can also be used as an example of how to implement a new driver. This particular piece of software is an implementation of the interface IPolarizationCamera, that only reads a fixed image stored on disk. Each time the server requests a new image, a copy of it is returned.

The instructions about how to install, compile and run the server code are included in the README file of the GitHub repository, publicly available in [90].

B.2.3 Camera client and base polarization processing

The second component of the software is the Graphical User Interface (GUI), which works as a client of the ROS server. A complete view of it, and the options included on it is shown in Fig. B.2 Each functionality in the software has been developed as a Widget of the Qt framework [21], which is a graphical element that can contain other basic graphical elements. This interface allows the user to perform all the required tasks involving the camera such as changing the acquisition parameters, and the super-pixels filter configuration. It also enables image processing, raw image display, sensor calibration, and plotting functions to analyze the calibration performance. The core functionalities and processing techniques included

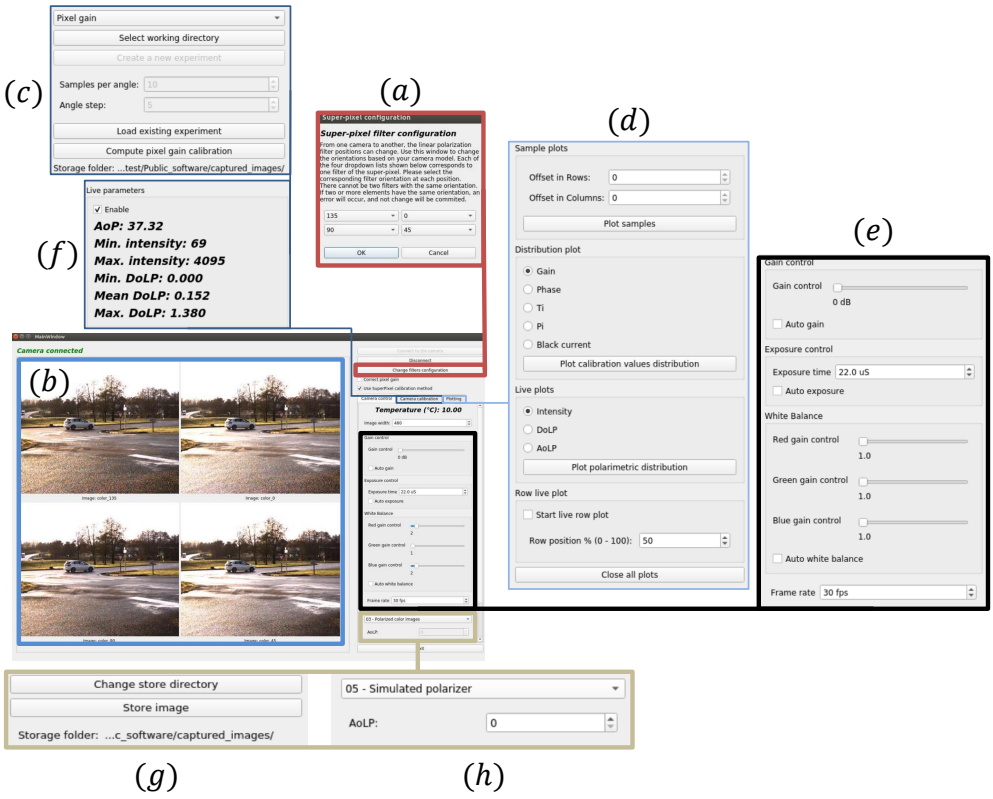


Figure B.2: Developed graphical user interface. (a) Widget that will appear each time the button *Change filters configuration* in the MainWindow is pressed. This widget allows us to change the position of the linear polarization filters of the camera we have. (b) Widget that shows the images obtained after applying a set of operations, depending on the selected visualization mode. The images are updated each time a new image from the camera is received. (c) Load / save widget interface. (d) Matplotlib widget interface. (e) Camera parameters widget interface. (f) Real-time parameters computation widget. (g) Save image widget interface. (h) Visualization mode widget interface.

are:

Raw split images: It produces as output four images. Each of them corresponds to the set of pixels that have the same polarization filter orientation, independently of the color filters. This transformation is the separation of the image by polarization channels.

Polarized color images: It produces as output four images. They are the corresponding demosaiced version of the output from the Raw split images mode. The algorithm used over each image is the conventional Bayer interpolation algorithm.

Original color: Only one image is returned. This outcome is equivalent to the image captured with a conventional color camera. It corresponds to the first component of the Stokes vector S_0, which comes from doing the average of the polarization channels (i.e., from

the average of the output images of the Raw split images mode). The obtained image is then demosaiced.

Stokes images: After separating the input image by polarization channel, the Stokes components images can be obtained by applying Eq. (2.9). Since the polarization state of the light depends on the frequency of the light, the Stokes vectors are split by color channel. As a consequence, this function will provide $3 \times 4 = 12$ images. Four color channels are considered since the Bayer patterns consist of 2×2 arrangements of red, green, green, and blue color filters.

Raw I - Rho - Phi: As also explained in Sec. 2.1, the Stokes vector can be represented as a function of three physical parameters: the total received intensity I, the degree of linear polarization ρ, and the angle of linear polarization ϕ. The equations to compute these variables as a function of the Stokes vector are given in Eq. (2.1). Again, for each color channel, these three physical parameters can be computed, and that is the reason why 12 images are also returned in this mode. Since each parameter has a different interval of values, all of them have been normalized to be in the range $[0, 255]$.

I - Rho - Phi: In the Raw I - Rho - Phi mode, all the images are single channeled, thus they are displayed in gray-scale. Particularly for the Angle of Linear Polarization (AoLP), this is not an adequate representation, since it is a circular variable. A proper representation would be one that assigns the same color to the maximum and minimum values in the range $[0, 255]$. In this mode, this is done by creating a color palette based on the Hue Saturation Value (HSV) color space. Let X be a gray-scale value in the range $[0, 255]$. The HSV palette is defined as a function that assigns a 3D vector $\mathbf{C_{hsv}}$ to each value X, such that:

$$\mathbf{C_{hsv}} = \left[\frac{179X}{255}, 255, 255\right]^T \tag{B.1}$$

Then, the obtained three-channeled image is converted from the HSV to the RGB color space. On the other hand, the Degree of Linear Polarization (DoLP) is colored using the Jet palette, which maps blue colors to low values, green colors to middle values, and red colors

to high values. As in the Raw I - Rho - Phi mode, 12 images are returned in this mode.

Fake colors: When dealing with polarization imaging, a correspondence is established between the Hue Saturation Value (HSV) color space and the intensity, Degree of Linear Polarization, and Angle of Linear Polarization [116]. Since the AoLP is a circular variable, it is considered to be the hue of the color. The saturation is the purity of that color, meaning that a saturation of 100% is a pure color, and a saturation of 0% means that it is a gray-level value. Similarly, the DoLP indicates the "purity" of the polarized light: if it is totally unpolarized, ρ is equal to zero, and if it is totally linearly polarized, ρ is equal to 1. Finally, if a conventional color image is considered, and the value channel is extracted from it in the HSV space, a gray-scale version of the original image is obtained. This information is similar to the total intensity measured by the I parameter. Thus, a color image can be obtained if the I, ϕ, and ρ images are stacked together, considered to be in the HSV space, and then converted back to the RGB space. The result obtained in this way is called a fake color image. The colors obtained with this processing algorithm have the following properties:

- Unpolarized light is represented by gray-scale colors.
- Highly polarized light will be colored.
- The colors in the image depend mainly on the AoLP, thus, to the surface orientation.

This processing is interesting since it helps to quickly identify the objects that reflect polarized light. As explained before, the polarization parameters depend on the frequency, thus the fake color images are separated by color channel. Therefore, this operation returns four color images.

B.2.3.1 White-balance module

The observed light by a camera depends on two parameters: the observed object reflectance, and the illumination of the color [79]. Thus, a white-balance algorithm needs to be used in order to restore the true colors in the scene. In some polarization cameras, and particularly in the Basler RGB-polarization camera, this type of algorithm is not correctly implemented.

Thus, our software includes an implementation of a white-balance algorithm. It consists on a gain applied to each color channel to correct the pixel intensities proportion, and restore the true color in the environment the image is taken. In our implementation, the automatic white search is done globally, and not in a user-defined region of interest (ROI). The algorithm computes the average of all the color channels of a single orientation, and it searches for the pixels whose average is the highest. If there is a white piece in the scene, even with the color gains unbalanced, the average of the white will be the highest. Thus, the pixel whose average is the highest is considered as white. Then, the highest channel value is left untouched (gain equal to 1), and the other channel gains are computed such that their values equal the highest channel value. This algorithm of automatic white balance is constrained to work with no high-level saturated images. If it does not produce satisfactory results, it can be deactivated and the different gains can be set manually.

B.2.3.2 Polarimetric camera calibration module

This module is an implementation of our polarimetric camera calibration algorithm described in Chapter 3. The calibration algorithm is used to compute a series of matrices that are applied to the raw images by a correction function. This functionality is aimged to rectify the measurement errors due to manufacturing imperfections, and polarization distortions due to the lens. In this way, two super-pixels of the same color channel that receive the same light source will provide the same output measurements. The obtain the calibrated image, if the calibration matrices have been computed beforehand, given a raw image from the camera, the correction function will return another image, with the same structure as the input, but with all the pixel measurements adjusted by the corresponding pixel matrices.

B.2.4 Polarization processing algorithms

Differently from Sec. B.2.3 where basic polarimetric operations can be done, in this module two applications of the polarization concepts are implemented.

The first one is the simulated polarization filter. As explained in Sec. 2.1, each super-pixel allows for computing the Stokes vector $\mathbf{S} = [S_0, S_1, S_2]^T$ of the incident light. Now, let

us consider a light, described by the Stokes vector $\mathbf{S}_{in}$ that passes through a linear polarizer. This filter is oriented at an angle θ and modeled by a Mueller matrix $\mathbf{M}$, as explained in Sec. 2.1. Therefore, the effects of a linear polarizer in front of a normal camera can be simulated by computing Eq. (2.7). Thus in this functionality, the inputs are a raw image from the camera and the orientation of the filter θ that one would like to simulate. Then, this algorithm returns two images: the input image, and the filtered image after applying Eq. (2.7) to all the super-pixels. This technique is commonly used in photography to remove annoying polarized reflections from the environment. In a real system with a RGB camera, the filter is physically placed on top of the lens, and turned until the reflection is removed. Once captured, no further modifications of this type can be done over the image. With this software, the filter and its effects are simulated after the image has been captured, and the exact angle for reflection removal can be found. The theory tell us that this orientation corresponds to the AoLP of the incident light, shifted by 90 degrees.

The second functionality of this module is the polarized specularity removal. It is an extension of the simulated polarization filter, explained above. In the previous case, all the pixels are affected by a single polarization filter. But, in the FoV of the camera, there might be several objects that produce this type of reflection, with different AoLP. To erase them all at once, let us consider the Stokes vector of the observed light. This vector can be split into two other Stokes vectors: one that represents totally unpolarized light and another that represents a totally linearly polarized light $\mathbf{S} = \mathbf{S_{unpol}} + \mathbf{S_{pol}}$ such as:

$$\begin{bmatrix} S_0 \\ S_0\rho\cos(2\phi) \\ S_0\rho\sin(2\phi) \end{bmatrix} = (1-\rho)\begin{bmatrix} S_0 \\ 0 \\ 0 \end{bmatrix} + \rho\begin{bmatrix} S_0 \\ S_0\cos(2\phi) \\ S_0\sin(2\phi) \end{bmatrix}. \tag{B.2}$$

Removing the polarized reflection means erasing the component corresponding to $\mathbf{S_{pol}}$. This is equivalent to computing the $\mathbf{S_{unpol}}$ vector. This functionality returns two images: the input and the filtered images, both demosaiced. In contrast to the previous case, this functionality does not require the user to enter an angle to each filter. The filtering is done based on the measured DoLP at each super-pixel.

Figure B.3: Raw image of an urban scene used as a reference for all the algorithms implemented in the toolkit Pola4All.

B.3 Experiments

In this section, we show and discuss the results of the different processing algorithms. To be able to compare the polarization properties through the developed software library toolkit, we adopt the same test image of an urban scene in all the experiments. This image, corresponding to the raw image obtained from the camera, is shown in Fig. B.3.

B.3.1 Basic polarimetric representation

The raw image is a single-channel matrix, formed by all the pixel measurements. The light that arrives at each pixel is filtered by color, and by polarization. Then, each channel must be extracted and interpolated to obtain the 12 corresponding channels ($[R, G, B] \times [I_0, I_{45}, I_{90}, I_{135}]$).

As the first step, this raw image can be split by polarization filter orientation. Thus, four images are obtained, and this functionality corresponds to the *Raw split images* mode. Each of these images contains all the pixels that are filtered by linear polarization filters with the same orientation, independently of the color filter. Therefore, each image contains the required information to apply an interpolation algorithm and with it, obtain the corresponding color image. This last behavior is obtained by using the mode *Polarized color images* of the software. Since the raw measurements of the camera are used, they are not white-balanced. Thus, the resulting images exhibit a greenish aspect. An example of one of the

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

Figure B.4: (a) Raw, mosaiced version of the reference image, for the polarization channel 0°. (b) Demosaiced version of the reference image, without white-balance, for the 0° polarization channel. (c) White-balanced, demosaiced images for the polarization channel 0°. (d) White-balanced, demosaiced images for the polarization channel 135°.

raw images, its corresponding demosaiced version, and the same image after white-balance is shown in Fig. B.4. The correction is performed using the white balance feature presented in Sec. B.2.3.1. For this testing image, the gains have been chosen manually, with a value of 2 for the red and blue channels, and with a value of 1 for the green channel. These are the first cases in which the polarization can be seen clearly. If the light received by the camera is polarized, then the response of each filter will be different. As explained in Sec. 2.1, the intensity of a linearly polarized light will describe a sine wave shape when it passes through a linear polarizer that rotates (due to the gain produced by this type of optical device). This sine wave reaches its maximum value when the orientation of the filter is equal to the angle of the linearly polarized light and it decreases as the filter rotates. The intensity reaches a minimum when the orientation of the filter is shifted by $\pi/2$ radians with respect to the Angle of Linear Polarization of the light.

In the case of the color-polarization camera, four filter orientations are considered. If any of these orientations matches with the AoLP of the incoming polarized light, then the corresponding image will have a bright spot, and the one that is shifted by $\pi/2$ radians will produce a dark spot at the same position. This can be seen over the windshields of the cars in the Fig. B.4 (d). Since these surfaces made of an insulator material produce specular reflections, the measured light is partially linearly polarized. The orientations at which the dark spots appear is 135°. This means that the AoLP of the incoming light is close to 45°, and it is highly linearly polarized. If a conventional camera is used, and a linear polarizer is placed in front of it with an orientation of 135°, the polarized specular reflection is removed, and it will be possible to see through the glass.

Figure B.5: Polarization images for the red channel. (a) (b) (c) S_0, S_1, S_2 components of the Stokes vector. (d) (e) (f) Total intensity, Raw AoLP, and Raw DoLP.

From the raw split images, it is also possible to compute the images of the linear components of the Stokes vector. As mentioned before while explaining the software functionalities, since this vector depends on the frequency, each Stokes parameter is separated by color channel. Due to its similarity, only the images for the red channel are shown in Fig. B.5 (a) to (c). In the S_1 and S_2 images, the brighter the pixel, the higher the Stokes vector value. Since these two parameters can have negative values, the absolute value of S_1 and S_2 are shown. As drawn from the previous case, the windshields present highly polarized reflections in the $\pm 45°$ directions. That is why they appear whiter in the S_2 image. Other regions in the images emit a very low degree of polarization, and as a consequence, they are represented by dark colors in the S_1 and S_2 images. The S_0 image corresponds to the red channel of the original color image. Once the Stokes vector for each color channel have been computed, the physical variables AoLP, denoted by ϕ, the DoLP, denoted by ρ, and the total intensity, denoted by I, can be calculated per color channel. The resulting images for the red channel are shown in Fig. B.5 (d) to (f).

These images are a good representation of all the objects that reflect linearly polarized

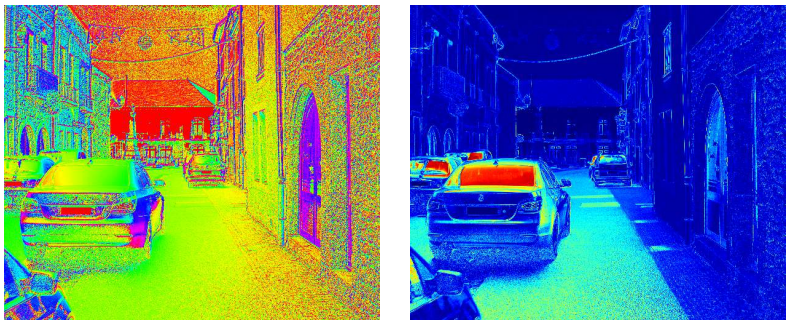


Figure B.6: Left image: Angle of Linear Polarization of the red channel, colored with the HSV palette. Right image: Degree of Linear Polarization of the red channel, colored with the Jet palette.

light. Note that a pixel value in the AoLP image has a meaning only if its corresponding pixel in the DoLP image has a non-zero value. In this set of images, it is possible to confirm that the road, the windshields, part of the body of the car, and some door glasses have a large DoLP. Thus, these features are extremely valuable and can be used, for example, in Deep Learning models to improve the accuracy of the network results over these objects. However, this representation has a problem: it does not consider the AoLP as circular, with a period of 180°. As a consequence, the maximum and minimum values are represented by different gray-scale values, although they should be equal. Particularly, when the measured angle is close to 0°, or to 180°, oscillations between them may happen due to noise. Therefore, the robustness of an implemented algorithm may be undermined if circularity is not considered.

To better represent the value changes, a circular color palette can be used for the AoLP, and a linear color palette for the DoLP. This colorization method is obtained by using the *I - Rho - Phi* mode. The results are shown in Fig. B.6 for the red channel. The AoLP and the DoLP have been colored as explained in Sec. B.2.1. The images presented so far constitute the richest polarization information the camera can provide: red, green, blue, and the polarization parameters ρ and ϕ. Nine single-channeled images are generated. Thus, a more compact version of this information may be required. For doing so, the *fake colors* representation can be used. The resulting image for the red channel is shown in Fig. B.7. As previously stated, in this representation, colored regions indicate partial or total polarization, while gray regions represent unpolarized areas.

(a) (b)

Figure B.7: (a) White-balanced, color image. (b) Fake colors image representation for the red channel.

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

Figure B.8: Specularity removal application results. (a) Unfiltered, total intensity. (b) (c) Filtered image by simulating a linear polarization filter oriented at 60°, and 120°, respectively. (d) Polarized specularity removal by filtering with the Degree of Linear Polarization.

B.3.2 Polarimetric applications

Finally, two useful algorithms have been included in the software. They make direct application of the polarization concepts: the linear polarization filter simulation, and the specularity removal. These functionalities aim to reduce undesired reflections in the color image. In the case of the linear polarization filtering, since each super-pixel allows for computing the Stokes vector, then this vector can be converted to any other Stokes vector by using Mueller matrices, as explained in Sec. 2.1. Particularly, an ideal linear polarizer oriented at an angle θ can be software simulated, and the results are shown in Fig. B.8 (b) and (c) for the angles $\theta = 60°$ and $\theta = 120°$, respectively.

As it can be noted, the choice of one angle or another will either reinforce or erase the windshields reflections. In conventional photography, this technique is applied using a physical filter that is turned until the bright spots are removed from the scene. The disadvantage of that method is that once captured, the filter effect cannot be modified anymore. With a color-polarization camera, this filtering can be done offline, by choosing the exact angle required to erase the undesired reflections.

The previously explained technique is not ideal when several objects are reflecting light at different Angle of Linear Polarization. As mentioned in Sec. B.2.4, a second specularity removal algorithm is implemented by combining the Degree of Linear Polarization and the total intensity of the incident light. This is equivalent to filtering the light at each super-pixel with a linear polarization filter oriented at an angle $\theta = \phi + \pi/2$, where ϕ is the Angle of Linear Polarization of the received light at that pixel. The filtered image is shown in Fig. B.8 (d). One can note that most reflections from the shiny surfaces, such as the windshield, the road and the door glasses, have been removed.

B.3.3 Calibration

All the results obtained so far have been generated using an uncalibrated camera. If more accurate measurements are required, then the camera needs to be calibrated. The calibration problem can be solved by taking several images of a uniform and linearly polarized light. If the light source is uniform but unpolarized, it can be polarized using a linear polarization filter. By turning the filter, the light received by the camera at each filter position will have a different AoLP. A sample of a polarized light source with an AoLP of 40° is shown in Fig. B.9 (a) to (c), for the 0°, 45° and 90° polarization channels, respectively.

The calibration procedure will compute the pixel parameters given the model defined in [91], i.e., (T_i, P_i, θ_i). In this model: T_i is the pixel gain; P_i is a parameter that accounts for the non-ideality of the micro-polarization filter implemented on the pixel; θ_i is the effective orientation of the micro-polarization filter of the pixel; i is the position of the pixel considered. From Mueller calculus, these parameters have an ideal value of $(T_i, P_i) = (0.5, 1.0)$ for all i, and $\theta_i \in \{0°, 45°, 90°, 135°\}$. However, in general, each pixel will have a set of parameters that will be different from these ideal values. Thus, the calibration algorithm allows for compensating these differences.

To assess the acquisition quality, the different plot functions of the software can be used. With these plots, it is possible to determine if the acquisition is correct, and if the camera measurements are valid. It can also assess the quality of the sensor and confirm the effectiveness of the correction on the camera measurements.

Among the available plots, the histograms of the intensity, the DoLP and the AoLP of the

incident light can be computed from the image currently displayed by the software. These plots can be visualized only if the software is connected to the camera. This information allows to evaluate the quality of the calibration results. If not errors occured during the calibration procedure, after correction, the three parameters (intensity, DoLP and AoLP) will have a narrower distribution than the uncalibrated case. The difference between the two cases depends on the sensor quality, and the level of distorsion introduced by the lens. The histograms of these variables, before and after calibration are depicted in Fig. B.9 (d) to (i). They have been computed with the same sample image, mentioned before, of a Uniform Linearly Polarized with an AoLP of 40°.

A real-time plot of these three parameters (intensity, DoLP, and AoLP) can also be done for a given row of pixels of the sensor. These graphs are displayed in the last two columns of Fig. B.9, and they correspond to the measurements before and after applying the calibration. These plots illustrate the vignetting effect and show how calibration can reduce its impact over the three polarization parameters. It is important to note that this correction is obtained since the pixel model used considers the polarization parameters of the pixel, and not only the unbalanced sensing gain. A simple gain correction will only affect the intensity image, but not the AoLP nor the DoLP images.

Finally, the consequence of applying the calibration can be observed in the reference urban scene image. In our developed software, the correction by calibration can be enabled or disabled using the checkbox *Correct pixel gain*, located in the top right region of the GUI. The effects of the calibration over the intensity, AoLP, and DoLP images are shown in Fig. B.10. Particularly from these images, the contribution of the calibration can be observed mostly in the AoLP and the intensity images. In the scene, there are several walls that act as planes. Thus, they should reflect the same AoLP, which translated to the fact that they should have the same color. This happens only after calibration, mainly in the building situated in the far region of the image, and in the walls on both sides of the road. In the intensity image, since the pixel model includes a gain factor, the vignetting effect is also corrected with this system, making the darker areas in the borders to be as bright as the center of the image.

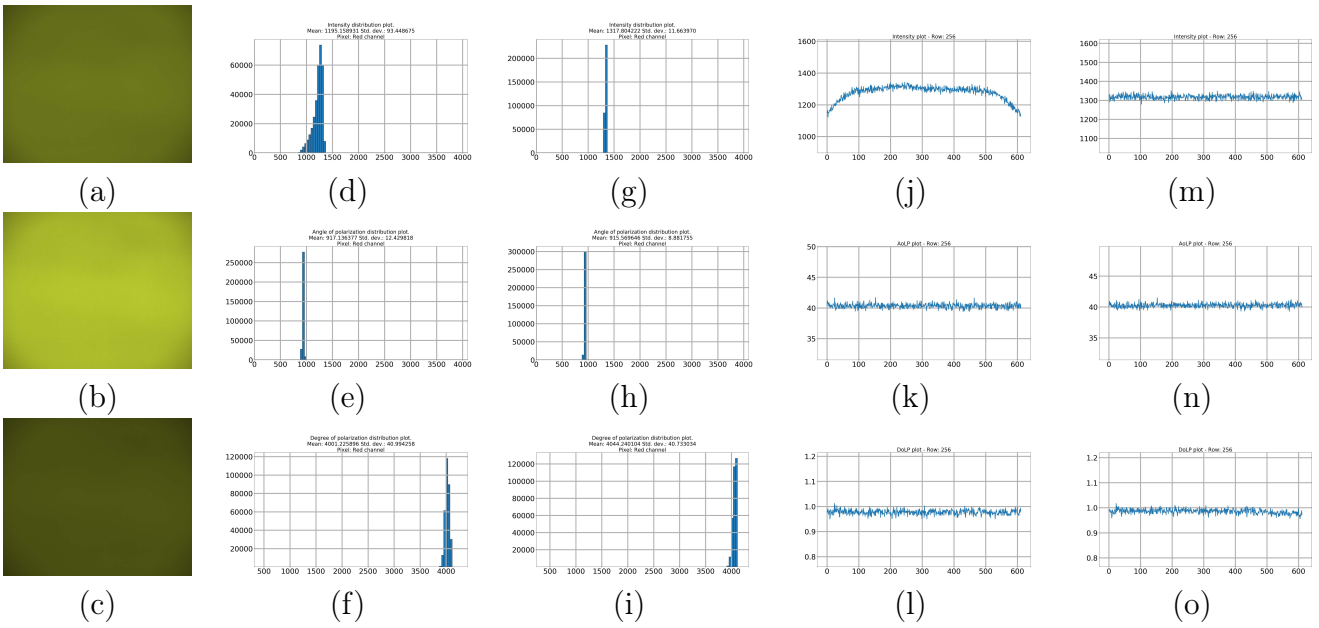


Figure B.9: Calibrated camera quality evaluation. All the plots are computed before (second and forth columns) and after (third and fifth columns) calibration. (a), (b), and (c) are the demosaiced, RGB polarization channels 0°, 45°, and 90°, respectively. The images correspond to a uniform, linearly polarized light source used for the calibration, with an AoLP of 40°. (d) and (g) Intensity histograms. (e) and (h) AoLP histograms. (f) and (i) DoLP histograms. (j) and (m) Intensity measurements plots over a single row of pixels. (k) and (n) AoLP measurements plots over a single row of pixels. (l) and (o) DoLP measurements plots over a single row of pixels.

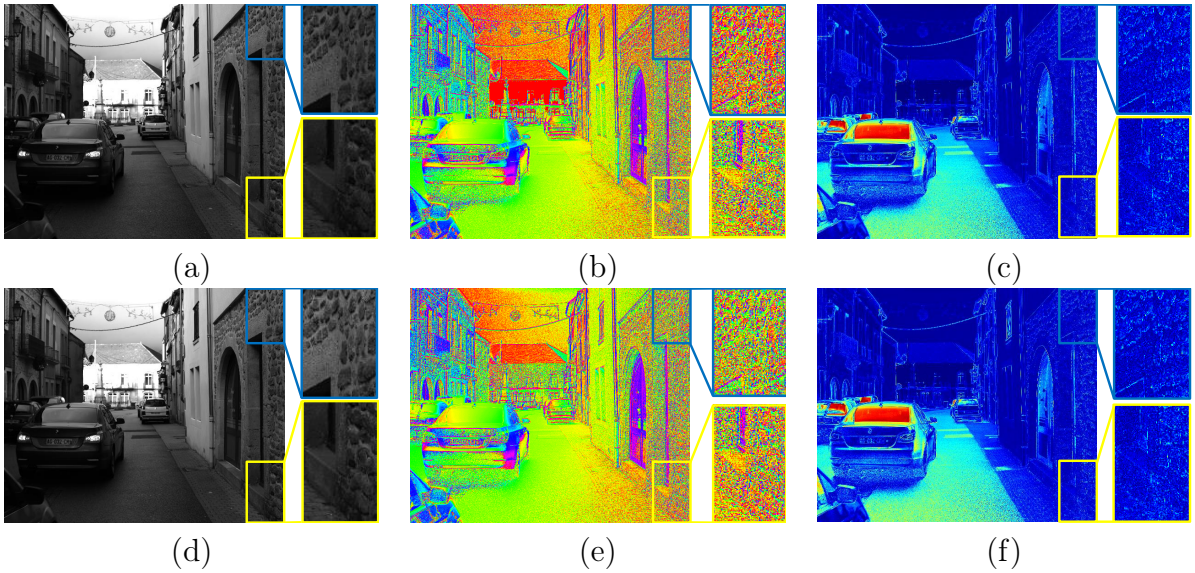


Figure B.10: Calibration effects over the polarization images, for the red channel. Top row: uncalibrated images. Bottom row: calibrated images. (a) (d) Total intensity. (b) (e) Angle of Linear Polarization, colored with the HSV palette. (c) (f) Degree of Linear Polarization, colored with the Jet palette.